50 1190 0101

Утвержден

РУСБ.10015-01-УД

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

ОПЕРАЦИОННАЯ СИСТЕМА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ «ASTRA LINUX SPECIAL EDITION»

Руководство администратора. Часть 1

РУСБ.10015-01 95 01-1

Листов 389

2024

Литера $О_1$

АННОТАЦИЯ

Настоящий документ является первой частью руководства администратора программного изделия РУСБ.10015-01 «Операционная система специального назначения «Astra Linux Special Edition» (далее по тексту — ОС).

Документ предназначен для администраторов системы и сети. Администраторы безопасности должны руководствоваться документом РУСБ.10015-01 97 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1».

Руководство администратора состоит из двух частей:

- РУСБ.10015-01 95 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 1»;
- РУСБ.10015-01 95 01-2 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 2. Установка и миграция».

Перед установкой и настройкой ОС необходимо провести ее контроль, предусмотренный формуляром при первичном закреплении экземпляра ОС за ответственным лицом.

В первой части руководства приведено назначение и настройка ОС. Рассмотрены системные компоненты, службы и команды, базовые сетевые службы, средства организации ЕПП, защищенная графическая подсистема, управление программными пакетами, резервное копирование и восстановление данных, система печати, защищенные комплексы программ гипертекстовой обработки данных и электронной почты, средства контроля целостности, централизованного протоколирования и разграничения доступа к подключаемым устройствам. Приведен список сообщений для администратора.

Во второй части руководства приведено описание установки ОС, а также порядок миграции на текущее очередное обновление ОС.

Требования к обеспечению безопасности среды функционирования, а также настройка параметров, необходимых для безопасной эксплуатации ОС, приведены в документе РУСБ.10015-01 97 01-1 и выполняются администратором безопасности.

Порядок работы с защищенной СУБД приведен в документе РУСБ.10015-01 97 01-3 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 3. Защищенная СУБД».

Дополнительная информация о настройке компонентов и управлении программными пакетами, а также варианты реализации отдельных решений с использованием ОС приведены на официальном сайте `wiki.astralinux.ru`.

СОДЕРЖАНИЕ

1. АДМИНИСТРИРОВАНИЕ ОС

Административное управление в ОС отделено от общего доступа пользователей.

Большинство операций по настройке и администрированию ОС требуют прав суперпользователя (`root`), например:

- монтирование и размонтирование ФС;
- изменение корневого каталога процесса командой `chroot`;
- создание файлов устройств;
- установка системных часов;
- изменение принадлежности файлов;
- задание host-имени системы;
- конфигурирование сетевых интерфейсов.

ВНИМАНИЕ! После установки ОС интерактивный вход в систему суперпользователя по умолчанию заблокирован. Для администрирования системы при установке ОС создается пользователь, входящий в группу `astra-admin`. Пользователю, входящему в группу `astra-admin`, через механизм `sudo` предоставляются права для выполнения действий по настройке ОС, требующих привилегий `root`. Далее по тексту такой пользователь именуется администратором. Описание механизма `sudo` приведено в 1.1.2.

ВНИМАНИЕ! Действия по администрированию ОС при включенном мандатном контроле целостности (МКЦ) необходимо выполнять в привилегированном режиме с высокой меткой целостности. Пользователю, создаваемому при установке ОС, назначается максимальная метка целостности. Описание МКЦ приведено в документе РУСБ.10015-01 97 01-1.

1.1. Получение прав суперпользователя

Существует несколько способов получения прав суперпользователя:

- вход в систему от имени учетной записи `root` (по умолчанию заблокирован);
- использование команды `su` (по умолчанию заблокирован);
- использование команды `sudo` (рекомендуется).

1.1.1. su

Команда `su` используется пользователем для запуска команд от имени другого пользователя. В том числе могут быть запущены команды от имени учетной записи `root`.

При запуске команды `su` без параметров подразумевается, что пользователь хочет запустить командный интерпретатор `shell` от имени учетной записи `root`. При этом система просит

ввести пароль от учетной записи root. При вводе правильного пароля запускаемый интерпретатор команд получает права и привилегии суперпользователя, которые сохраняются до завершения его работы. Пользователю для получения прав суперпользователя не требуется завершать свою сессию и вновь входить в систему.

С помощью команды su, вводимой с параметром -c, пользователь может выполнять отдельные команды от имени учетной записи root без запуска командного интерпретатора shell. При этом пользователь получает права и привилегии суперпользователя на ограниченное время, а именно, на время исполнения заданной команды. Например, при необходимости поменять атрибуты файла ввести команду от имени учетной записи root:

```
su -c 'chmod 0777 /tmp/test.txt'
```

После ввода пароля учетной записи root команда chmod получит права и привилегии суперпользователя на выполнение заданного запроса, но при этом права и привилегии пользователя на выполнение других команд не изменятся.

Кроме выполнения команд от имени учетной записи root, команда su позволяет выполнять команды от имени любого другого пользователя, при этом для выполнения команды необходимо знать пароль этого пользователя. Если вход в систему выполнен от имени root, то при использовании su для выполнения команды от имени другого пользователя знание пароля данного пользователя не требуется — все команды от имени любого пользователя исполняются без запроса пароля.

При предоставлении прав на использование команды su следует учитывать, что для нее отсутствует механизм ограничения списка команд, разрешенных конкретному пользователю выполнять от имени учетной записи root. Таким образом, если у пользователя есть права на выполнение команды su, то он может выполнить от имени учетной записи root любые команды. Поэтому использование команды su должно быть разрешено только доверенным пользователям. Также рекомендуется при вводе команды использовать полное путевое имя /bin/su (вместо su).

Описание команды приведено в man su.

1.1.2. sudo

Команда sudo используется пользователем для запуска команд от имени учетной записи root.

В качестве параметров команда sudo принимает командную строку, которую следует выполнить с правами суперпользователя. При выполнении команды sudo просматривается конфигурационный файл /etc/sudoers, в котором приведен список пользователей, имеющих полномочия на запуск команды sudo, а также перечень команд, которые каждый из пользователей имеет право выполнять от имени учетной записи root. Если данному

пользователю разрешено выполнять указанную им команду, то при выполнении команды `sudo` у пользователя запрашивается его пароль. Таким образом, для каждого пользователя установлен набор команд, которые он может выполнять от имени учетной записи `root` без необходимости вводить пароль учетной записи `root`.

При использовании `sudo` подсистемой регистрации событий регистрируется следующая информация: выполненные команды, вызвавшие их пользователи, из какого каталога вызывались команды, время вызова команд.

Для изменения файла `/etc/sudoers` используется команда `visudo`, запущенная от имени администратора.

Описание команды приведено в `man sudo`.

1.2. Механизмы разделения полномочий

К механизмам разделения полномочий между системными администраторами ОС могут быть отнесены:

- механизм привилегий;
- механизм повышения полномочий на время выполнения команды (программы);
- механизм установки ACL на файлы.

Описание механизмов разделения полномочий приведено в документе РУСБ.10015-01 97 01-1.

1.2.1. Механизм привилегий

Механизм привилегий ОС предназначен для передачи отдельным пользователям прав выполнения определенных административных действий. Обычный пользователь системы не имеет дополнительных привилегий.

Привилегии наследуются процессами от своих «родителей» и не могут быть переданы сторонним процессам. Процессы, запущенные от имени суперпользователя, независимо от наличия у них привилегий, имеют возможность осуществлять все привилегированные действия.

Распределение (первоначальная настройка) привилегий выполняется администратором с максимальной меткой целостности, установленной в ОС.

1.2.2. Механизм повышения полномочий

Механизм повышения полномочий позволяет повысить полномочия пользователя на время выполнения определенной программы. Настройка механизма может быть выполнена администратором с максимальной меткой целостности, установленной в ОС.

1.2.3. Механизм установки ACL на файлы

Механизм установки ACL на файлы облегчает задачу распределения полномочий, позволяя предоставлять доступ только к тем файловым объектам, к которым он необходим в соответствии с ролью пользователя. Настройку механизмов ACL выполняет администратор с максимальной меткой целостности, установленной в ОС.

2. УСТАНОВКА, НАСТРОЙКА И ОБНОВЛЕНИЕ ОС

2.1. Установка ОС

Подробное описание последовательности действий при установке ОС или миграции с предыдущих очередных обновлений приведены в РУСБ.10015-01 95 01-2.

В программе установки необходимо выбрать уровень защищенности ОС в соответствии с лицензионным соглашением:

1) базовый («Орел»);
2) усиленный («Воронеж»);
3) максимальный («Смоленск»).

2.2. Первичная настройка ОС

В пунктах 2.2.1–2.2.3 приведены примменяемые настроки ОС для соответствующего уровня защищенности в случае, если при установке ОС были выбраны предложенные по умолчанию значения.

2.2.1. Базовый уровень защищенности («Орел»)

После установки ОС готова к использованию без дополнительных настроек.

На данном уровне защищенности для разграничения доступа применяется механизм дискреционного управления доступом (в т.ч. в защищенной СУБД). По умолчанию выключены режим отладки `ptrace` и возможность использовать механизм `sudo` без ввода пароля.

Дополнительно для защиты информации могут использоваться доступные системные ограничения, а также функции безопасности, ограничивающие действия пользователей.

Настройка функций безопасности выполняется в соответствии с документом РУСБ.10015-01 97 01-1 и для защищенной СУБД в соответствии с документом РУСБ.10015-01 97 01-3.

2.2.2. Усиленный уровень защищенности («Воронеж»)

После установки ОС мандатный контроль целостности (МКЦ) ОС и файловой системы включаются автоматически. При включенном режиме МКЦ администрирование и настройка ОС должны выполняться только администратором с высокой меткой целостности.

На данном уровне защищенности для разграничения доступа применяется механизм дискреционного управления доступом (в т.ч. в защищенной СУБД). По умолчанию выключены режим отладки `ptrace` и возможность использовать механизм `sudo` без ввода пароля.

Дополнительно для защиты информации могут использоваться доступные системные ограничения, а также функции безопасности, ограничивающие действия пользователей.

Также на данном уровне защищенности для защиты информации доступны механизмы очистки памяти (в т.ч. для защищенной СУБД) и организация замкнутой программной среды.

Настройка средств защиты информации и функций безопасности выполняется в соответствии с документом РУСБ.10015-01 97 01-1 и для защищенной СУБД в соответствии с документом РУСБ.10015-01 97 01-3.

2.2.3. Максимальный уровень защищенности («Смоленск»)

После установки ОС режим МКЦ ОС и файловой системы включаются автоматически. При включенном режиме МКЦ администрирование и настройка ОС должны выполняться только администратором с высокой меткой целостности.

На данном уровне защищенности для разграничения доступа по умолчанию применяются механизмы мандатного управления доступом и дискреционного управления доступом (в т.ч. в защищенной СУБД). После установки ОС требуется определить режим работы КСЗ и выполнить генерацию КСЗ в соответствии с РУСБ.10015-01 97 01-1.

По умолчанию выключены режим отладки `ptrace` и возможность использовать механизм `sudo` без ввода пароля.

Дополнительно для защиты информации могут использоваться доступные системные ограничения, а также функции безопасности, ограничивающие действия пользователей.

Также на данном уровне защищенности для защиты информации доступны механизмы очистки памяти (в т.ч. для защищенной СУБД) и организация замкнутой программной среды.

Настройка средств защиты информации и функций безопасности выполняется в соответствии с документом РУСБ.10015-01 97 01-1 и для защищенной СУБД в соответствии с документом РУСБ.10015-01 97 01-3.

2.3. Смена локали в ОС

По умолчанию в ОС устанавливается и используется кодировка UTF-8 (локали `ru_RU.UTF-8` и `en_US.UTF-8`).

Смена кодировки по умолчанию не рекомендуется, так как это может вызвать ошибку загрузки графической сессии и некорректное отображение шрифтов.

Если необходимо использовать другую локаль для корректного отображения символов в определенной программе, то следует запускать эту программу с указанием локали в переменной окружения LC_ALL:

```
env LC_ALL=ru_RU.CP1251 <путь_к_исполняемому_файлу>
```

2.4. Создание LiveCD

2.4.1. Общие сведения

LiveCD — это ОС, загружаемая со съемного носителя (DVD-диска, USB-носителя), не требующая для своего функционирования установки на жесткий диск, при этом пользователю доступен весь функционал ОС.

В состав ОС входит инструмент командной строки live-build-astra для создания образа LiveCD (далее Live-образа).

Для установки live-build-astra выполнить команду:

```
sudo apt install live-build-astra
```

2.4.2. Сборка Live-образа

Для сборки Live-образа может быть использована одна из следующих команд:

```
live-build-astra [параметры]
live-build-astra -d <кодовое_имя_ОС> [параметры]
```

Команда может быть выполнена только от имени администратора.

Инструмент live-build-astra создает каталог сборки с конфигурационным файлом, скопированным из системного каталога /etc/astra-live/, другими необходимыми файлами для сборки Live-образа и запускает сборку.

При выполнении команды сборки без параметров будет открыто окно с запросом использовать для собираемого образа кодовое имя дистрибутива, на котором запущена команда сборки. При нажатии кнопки **[Нет]** выполнение команды сборки будет завершено с ошибкой отсутствия задания дистрибутива (значения параметра -d). Для согласия использовать для сборки кодовое имя дистрибутива, на котором запущена команда сборки, нажать кнопку **[Да]**. Окно с запросом будет закрыто.

В каталоге выполнения команды будет создан каталог сборки по умолчанию `./build/`, в котором:

1) в корень каталога `./build/` будет скопирован системный конфигурационный файл `/etc/astra-live/config`;

2) в `./build/sources` будет скопирован системный файл с источниками по умолчанию `/usr/share/astra-live/sources/<кодовое_имя_ОС>.list`;

3) в `./build/packages` будет скопирован системный файл с пакетами по умолчанию `/usr/share/astra-live/packages/<кодовое_имя_ОС>.list.chroot`;

4) в корень каталога `./build/` будут скопированы другие необходимые для сборки файлы и каталоги.

В `./build/` будет начата сборка Live-образа:

1) с кодовым именем дистрибутива, на котором запущена команда сборки;

2) из источника установки по умолчанию (из файла `<каталог_сборки>/sources/<кодовое_имя_ОС>.list`);

3) с установкой пакетов по умолчанию (из файла `<каталог_сборки>/packages/<кодовое_имя_ОС>.list.chroot`).

Live-образ будет создан в корне каталога `./build/`.

При выполнении команды `live-build-astra` с параметрами соответствующие значения параметров будут переопределены в конфигурационном файле `<каталог_сборки>/config` и будут перезаписаны файлы `<каталог_сборки>/sources/<кодовое_имя_ОС>.list` и `<каталог_сборки>/packages/<кодовое_имя_ОС>.list.chroot`.

ВНИМАНИЕ! В качестве репозиториев допустимо указывать только репозитории ОС. При использовании сторонних репозиториев полученный в результате сборки набор пакетов может оказаться неработоспособным.

Live-образ будет создан в каталоге сборки, заданном с помощью параметра `-b`.

Параметры инструмента `live-build-astra` приведены в таблице 1.

Таблица 1

Параметр	Описание
`-nal`, `--no-autologin`	Отключить автоматический вход в систему на собираемом образе. В LiveCD для входа в систему будет применен пустой пароль
`-hn <имя_хоста>`, `--host-name <имя_хоста>`	Указать имя хоста системы на собираемом Live-образе
`-un <имя_пользователя>`, `--user-name <имя_пользователя>`	Указать имя пользователя системы на собираемом Live-образе

Продолжение таблицы 1

Параметр	Описание
`-b <каталог>`, `--build-dir <каталог>`	Указать путь к каталогу сборки. Если путь к каталогу сборки не был указан в команде, то в качестве каталога сборки будет использован `./build/`
`-c <тип>`, `--compression <тип>`	Указать тип сжатия SquashFS. Если тип сжатия не был указан в команде, то будет применен тип сжатия `zstd`. Доступные типы сжатия `gzip`, `lzo`, `lz4`, `xz` и `zstd` (см. `man mksquashfs`)
`-d <кодовое_имя>`, `--distribution <кодовое_имя>`	Указать кодовое имя дистрибутива ОС. Если параметр не задан, то будет предложено использовать кодовое имя дистрибутива, на котором запущена команда сборки
`-rs`, `--remove-sources`	Удалить все содержимое каталога сборки перед началом сборки
`-kc`, `--keep-cache`	Не удалять кэш `apt` из каталога сборки. Используется совместно с `--remove-sources`
`-ki`, `--keep-images`	Не удалять ранее собранные ISO-образы из каталога сборки. Используется совместно с `--remove-sources`
`-kl`, `--keep-log`	Не удалять журналы сборки из каталога сборки. Используется совместно с `--remove-sources`
`-r <источник>`, `--repository <источник>`	Указать путь к одному или нескольким источникам. В качестве источника может быть указан путь к файлу образа, путь к каталогу монтирования, путь к каталогу локального репозитория, URL-адрес сетевого репозитория. По умолчанию используются источники из системного файла `/usr/share/astra-live/sources/<кодовое_имя_ОС>.list`. **ВНИМАНИЕ!** Если должно быть указано несколько источников, то первым должен быть указан установочный источник. Значения должны быть разделены пробелами или запятыми. Также данный параметр можно указывать в одной команде несколько раз
`-u`	То же, что и `-r`, используется для совместимости. Не рекомендуется к использованию. Данный параметр устарел и будет удален в будущих обновлениях
`-ap <пакеты>`, `--add-packages <пакеты>`	Добавить указанные пакеты. Если должно быть указано несколько пакетов, то имена пакетов должны быть разделены пробелами или запятыми
`-apl <файл>`, `--add-packages-list <файл>`	Добавить все пакеты из указанного файла. В файле каждое имя пакета должно располагаться на отдельной строке
`-dp <пакеты>`, `--delete-packages <пакеты>`	Исключить из файла с пакетами по умолчанию `<каталог_сборки>/packages/` все указанные пакеты. Если должно быть указано несколько пакетов, то имена пакетов должны быть разделены пробелами или запятыми

Продолжение таблицы 1

Параметр	Описание
-dpl <файл>, --delete-packages-list <файл>	Исключить из файла с пакетами по умолчанию <каталог_сборки>/packages/ все пакеты, содержащиеся в указанном файле. В файле каждое имя пакета должно располагаться на отдельной строке
-rp <пакеты>, --replace-packages <пакеты>	Заменить все пакеты, содержащиеся в файле пакетами по умолчанию <каталог_сборки>/packages/ <кодовое_имя_ОС>.list.chroot, на указанные пакеты. **ВНИМАНИЕ!** Данный параметр не может быть использован в одной команде совместно с параметрами --replace-packages-list и --tasks. Если должно быть указано несколько аргументов данного параметра, то имена пакетов должны быть разделены пробелами или запятыми
-rpl <файл>, --replace-packages-list <файл>	Заменить все пакеты, содержащиеся в файле со списком пакетов по умолчанию <каталог_сборки>/packages/ <кодовое_имя_ОС>.list.chroot, на все пакеты из указанного файла. В указанном файле каждое имя пакета должно располагаться на отдельной строке **ВНИМАНИЕ!** Данный параметр не может быть использован в одной команде совместно с параметрами --replace-packages и --tasks
-t <задачи>, --tasks <задачи>	Заменить все пакеты, содержащиеся в файле со списком пакетов по умолчанию <каталог_сборки>/packages/ <кодовое_имя_ОС>.list.chroot, на пакеты из указанных задач tasksel. **ВНИМАНИЕ!** Данный параметр не может быть использован в одной команде совместно с параметрами --replace-packages и --replace-packages-list. Если должно быть указано несколько аргументов данного параметра, то имена задач должны быть разделены пробелами или запятыми
-i <имя>, --iso <имя>	Создать ISO-образ с указанным именем. Если имя не задано, то у созданного ISO-образа будет имя типа livecd-DD.MM.YY_HH.MM.iso. Если в команде не указан ни один параметр, задающий расширение и имя выходного файла (-i, -D, -q, -R, -T, -V или -w), то параметр -i выполняется по умолчанию и генерируется ISO-образ
-D <имя>, --docker <имя>	Создать TAR-архив с DOCKER-контейнером с указанным именем. Если имя не указано, то у созданного архива будет имя типа docker-DD.MM.YY_HH.MM.tar
-q <имя>, --qcow2 <имя>	Создать QCOW2-образ с указанным именем. Если имя не указано, то у созданного QCOW2-образа будет имя типа qcow2-DD.MM.YY_HH.MM.qcow2
-R <имя>, --raw <имя>	Создать RAW-образ с указанным именем. Если имя не указано, то у созданного RAW-образа будет имя типа raw-DD.MM.YY_HH.MM.img

Окончание таблицы 1

Параметр	Описание
`-T <имя>`, `--tar <имя>`	Создать TAR-архив с указанным именем. Если имя не указано, то у созданного TAR-архива будет имя типа `tarball-DD.MM.YY_HH.MM.tar`
`-w <имя>`, `--wsl <имя>`	Создать TAR-архив дистрибутива WSL с указанным именем. Если имя не указано, то у созданного архива будет имя типа `wsl-DD.MM.YY_HH.MM.tar`
`-h`, `--help`	Вывод справки
`-v`, `--version`	Вывод версии

Пример

Создание Live-образа `astra-live.iso` с кодовым именем `1.8_x86-64` в каталоге сборки по умолчанию `./build`. В качестве источника установки использовать установочный ISO-образ и применить сжатие `zstd` для файловой системы LiveCD. Все пакеты по умолчанию заменить пакетами из задач `Base` и `Fly` и удалить пакет `mc`, входящий в задачу `Base`:

```
live-build-astra -d 1.8_x86-64 -r ~/iso-images/installation-1.8.iso \
     -i astra-live.iso -c zstd -t Base Fly -dp mc
```

Более подробное описание инструмента приведено в `man live-build-astra`.

Описание изменения набора пакетов в создаваемой LiveCD приведено в 2.4.3.

Описание записи на носитель информации собранного Live-образа приведено в 2.4.4.

2.4.3. Изменение набора пакетов в создаваемой LiveCD

Для изменения перечня пакетов, устанавливаемых в создаваемой LiveCD, используются следующие параметры инструмента `live-build-astra`: `--replace-packages`, `--replace-packages-list`, `--tasks`, `--add-packages`, `--delete-packages`, `--add-packages-list`, `--delete-packages-list`. Описание параметров приведено в таблице 1.

Параметры позволяют изменять список устанавливаемых пакетов, а также перечень пакетов из задач `tasksel`. Инструмент `tasksel` — это встроенная в ОС система управления пакетами, оперирующая предопределенными наборами пакетов, которые называются задачами `tasksel`.

Для просмотра перечня доступных задач в ОС выполнить команду:

```
tasksel --list-tasks
```

Для задания значений параметра `--tasks` необходимо использовать имена из второго столбца вывода команды.

Подробное описание инструмента `tasksel` см. в `man tasksel`.

Параметры, изменяющие список устанавливаемых в LiveCD пакетов и перечень пакетов в задачах `tasksel`, имеют разный приоритет применения.

Приоритет применения параметров для изменения списка устанавливаемых в LiveCD пакетов (в порядке уменьшения приоритета):

1) `--replace-packages` либо `--replace-packages-list`;

2) `--tasks`;

3) `--add-packages`, `--delete-packages`, `--add-packages-list`, `--delete-packages-list`.

Пакеты, от которых зависит работоспособность ОС, не указываются в файле `/usr/share/astra-live/packages/<кодовое_имя_ОС>.list.chroot` и не могут быть удалены.

Примечание. При задании в соответствующих параметрах путей с помощью переменных окружения или регулярных выражений необходимо учитывать, что все действия выполняются от пользователя `root`.

Описание записи на носитель информации собранного Live-образа приведено в 2.4.4.

2.4.4. Запись Live-образа

Созданный Live-образ можно использовать для загрузки ОС как с DVD-диска, так и с USB-носителя. Для этого необходимо записать ISO-образ на DVD-диск или USB-носитель.

Запись ISO-образа на USB-накопитель необходимо выполнять с использованием команды `dd` либо с помощью графической утилиты `fly-admin-iso` (описание утилиты приведено в электронной справке).

Пример

Запись ISO-образа `livecd.iso` на подключенный USB-носитель, представленный в системе файлом устройства `/dev/sdb`:

```
dd if=livecd.iso of=/dev/sdb bs=1M
```

ВНИМАНИЕ! Команда `dd` записывает новое содержимое, удаляя имеющиеся записи. Указание некорректных параметров может привести к потере данных или невозможности загрузки ОС.

2.5. Обновление ОС

2.5.1. Ручное обновление

Для ручной установки обновлений ОС используется инструмент командной строки `astra-update`.

Общий синтаксис команды:

```
astra-update [действие] [параметр] [источник][[источник]..]
```

В качестве источника может быть указан ISO-файл образа или сетевой репозиторий. Может быть указано несколько источников, разделенных пробелом.

При запуске команды может быть выбрано только одно действие. Список основных действий `astra-update` приведен в таблице 2.

Таблица 2

Действие	Описание
`-c`	Проверить, можно ли устанавливать обновление. Изменения в систему не вносятся. Является действием по умолчанию — выполняется, если в команде действие не указано
`-a`	Установить обновление автоматически в интерактивном режиме (с выводом запросов пользователю), выполняя автоматическое выключение и включение функций безопасности. Представляет собой последовательное выполнение действий `-d`, `-i` и `-e`
`-A`	Установить обновление полностью автоматически (без вывода сообщений и запросов пользователю), выполняя автоматическое выключение и включение функций безопасности. Представляет собой последовательное выполнение действий `-d`, `-I` и `-e`. Данный режим предназначен для массовой автоматической установки обновлений на удаленных компьютерах, в том числе для использования в сценариях ansible. Устройства чтения оптических дисков, добавленные с помощью команды `sudo apt-cdrom add`, не будут использованы в процессе неинтерактивной установки, т. к. могут потребовать действий пользователя
`-d`	Выключить функции безопасности, мешающие обновлению. Состояние функций безопасности при этом будет сохранено в файле `/etc/parsec/update-saveconf`
`-I`	Установить обновление в неинтерактивном режиме (без вывода сообщений и запросов пользователю) и не выполняя выключение и включение функций безопасности
`-i`	Установить обновление в интерактивном режиме (с выводом запросов пользователю) и не выполняя выключение и включение функций безопасности

Окончание таблицы 2

Действие	Описание
`-e`	Включить функции безопасности, которые были выключены перед обновлением действием `-d`. Состояние функций безопасности будет восстановлено из файла `/etc/parsec/update-saveconf`. Если файл не существует, то никакие изменения в систему внесены не будут
`-p <пакеты>`	Обновить инструменты обновления и пакеты, указанные в `<пакеты>`

Список параметров `astra-update` приведен в таблице 3.

Таблица 3

Параметр	Описание
`-k`	Сохранить источники для последующего использования (ISO-файлы образов будут скопированы на диск и указаны в `/etc/fstab`, сетевые репозитории будут добавлены в файл `/etc/apt/sources.list`)
`-K`	Установить последнее доступное ядро. Может использоваться только с действиями `-a`, `-A`, `-i` и `-I`
`-g`	Проверить контрольную сумму ISO-файла образа по алгоритму ГОСТ (файл с контрольной суммой должен располагаться в одном каталоге с образом)
`-m`	Проверить контрольную сумму ISO-файла образа по алгоритму MD5 (файл с контрольной суммой должен располагаться в одном каталоге с образом)
`-r`	Установка обновления из репозиториев, перечисленных в файле `/etc/apt/sources.list` (без внесения изменений в сам файл)
`-n`	Только имитировать установку обновления, без внесения изменений в систему
`-T`	Не искать репозиторий установочного диска, репозиторием обновления является технологический диск

ВНИМАНИЕ! Установочный образ системы всегда должен присутствовать в `/etc/apt/sources.list` или быть указан в качестве источника при выполнении команды `astra-update`.

Информацию по использованию инструмента `astra-update` можно просмотреть в терминале, выполнив команду:

```
man astra-update
```

Для установки обновлений также может использоваться графическая утилита `fly-astra-update`. Описание утилиты приведено в электронной справке.

2.5.2. Автоматическое обновление

Автоматическая проверка наличия обновлений ОС по подключенным репозиториям, их скачивание и установка осуществляются службой `astra-update-service`. Управление службой производится с помощью инструмента `astra-update-ctl`.

Синтаксис команды:

```
astra-update-ctl [параметр]
```

Список параметров приведен в таблице 4.

Таблица 4

Параметр	Описание
`status`	Отобразить статус службы
`enable`	Включить службу и добавить ее в автозапуск
`disable`	Отключить службу и убрать ее из автозапуска
`edit`	Открыть в консоли конфигурационный файл для редактирования
`parameters`	Вывести список доступных настроек конфигурационного файла
`set <статус>`	Принудительно установить один из возможных статусов: 1) `no-updates` — обновления отсутствуют; 2) `ready` — служба готова; 3) `activated` — служба запущена; 4) `stopped` — служба остановлена; 5) `force` — принудительная установка обновлений

Настройки службы хранятся в конфигурационном файле `/etc/astra-update-service/astra-update-daemon.conf`. Список возможных настроек приведен в таблице 5.

Таблица 5

Настройка	Описание
`T_check`	Интервал между проверками обновлений (в минутах), значение по умолчанию `60`
`T_download_min`	Минимальная задержка перед скачиванием обновлений (в минутах), значение по умолчанию `0`
`T_download_max`	Максимальная задержка перед скачиванием обновлений (в минутах), значение по умолчанию `240`
`T_delay`	Задержка перед обновлением системы (в днях), значение по умолчанию `7`

Окончание таблицы 5

Настройка	Описание
`T_retry`	Задержка перед следующей попыткой обновления системы (в часах), значение по умолчанию `4`
`Action_on_error`	Действие после ошибки обновления. Возможные значения: 1) `Reset` — сбросить изменения; 2) `Stop` — остановить процесс обновления; 3) `Retry` — повторить попытку обновления
`Always_new_update`	Удалять скачанные обновления перед скачиванием новых (`false`, `true`), значение по умолчанию `false`
`Host_to_ping`	Адрес, опрашиваемый для проверки наличия интернет-соединения, значение по умолчанию `8.8.8.8`

Журнал службы расположен в `/var/log/astra-update-service/update.log`.

Отображение уведомлений и подтверждение запуска процесса обновления осуществляется с помощью графической утилиты `fly-update-notifier`.

2.6. Установка и обновление ОС в режиме «Мобильный»

2.6.1. Подготовка к установке

Установка ОС в режиме «Мобильный» может быть выполнена на совместимое устройство (см. РУСБ.10015-01 31 01 «Операционная система специального назначения «Astra Linux Special Edition». Описание применения») со следующими характеристиками:

1) процессорная архитектура — x86-64 (AMD, Intel) с BIOS/UEFI;
2) оперативная память — не менее 1 ГБ;
3) внутренняя память — не менее 32 ГБ;
4) USB-порт.

Установка ОС на устройство выполняется с помощью установочного USB-носителя.

Для подготовки установочного USB-носителя требуется:

1) USB-носитель емкостью не менее 16 ГБ;
2) образ ОС для режима «Мобильный»;
3) инструментальный компьютер с установленной ОС или другой операционной системой семейства Linux.

Подготовка установочного USB-носителя:

1) загрузить инструментальный компьютер и войти в систему под учетной записью администратора;

2) в терминале выполнить команду:

```
sudo -s
```

3) записать на USB-носитель образ ОС, выполнив команду:

```
dd if=<путь_к_образу_ОС> of=/dev/<имя_устройства> bs=1M status=progress
```

где <имя_устройства> — имя USB-носителя в системе (данное имя будет отображаться в выводе команды dmesg после подключения USB-носителя, например sda).

После завершения процесса копирования установочный USB-носитель готов к использованию.

2.6.2. Установка

2.6.2.1. Настройка BIOS/UEFI

Перед установкой ОС в режиме «Мобильный» на устройство требуется настроить BIOS/UEFI устройства:

1) подключить установочный USB-носитель к устройству;

2) подключить внешнюю клавиатуру к устройству;

3) перезагрузить устройство и в процессе загрузки войти в меню настройки BIOS/UEFI, нажав соответствующую клавишу (на некоторых устройствах комбинация клавиш **<Fn+F2>**);

4) в меню настройки BIOS/UEFI установить корректную дату и время;

5) установить пароль на вход в BIOS/UEFI;

6) проверить настройку параметра Secure Boot — должен быть отключен («Disabled»);

7) сохранить изменения и перейти в следующее меню (на некоторых устройствах нажатием комбинации клавиш **<Fn+F10>**);

8) в меню настройки BIOS/UEFI выбрать вариант загрузки с USB-носителя и затем, в зависимости от модели устройства, нажать **<Enter>** или сохранить настройки и выйти из меню. На устройстве будет загружена ОС с установочного USB-носителя. По окончании загрузки будет отображено графическое меню установки ОС.

Установка ОС выполняется в одном из следующих режимов:

1) в графическом режиме, в соответствии с 2.6.2.2;

2) в терминальном режиме, в соответствии с 2.6.2.3.

2.6.2.2. Установка в графическом режиме

Для установки ОС на устройство в графическом режиме необходимо:

1) в графическом меню установки ОС выбрать «Install» и нажать **<Enter>**;

2) выбрать устройство внутренней памяти, на которое будет установлена ОС, и нажать **<Enter>**;

3) выбрать модель устройства и нажать **<Enter>**. Начнется копирование файлов ОС во внутреннюю память устройства. По окончании копирования файлов отобразится соответствующее сообщение;

4) для завершения установки и выключения устройства нажать **<Enter>**.

Далее требуется отсоединить установочный USB-носитель, включить устройство и выполнить первоначальную настройку ОС в соответствии с 2.6.3.

2.6.2.3. Установка в терминальном режиме

Для установки ОС в режиме «Мобильный» на устройство в терминальном режиме необходимо:

1) в графическом меню установки ОС выбрать «Exit» и нажать **<Enter>** — будет отображена командная строка и выполнен автоматический вход в систему с учетной записью суперпользователя `root`;

2) для установки ОС выполнить команду копирования:

```
/opt/astra-mobile-install -d /dev/<имя_устройства>
```

где `<имя_устройства>` — имя устройства внутренней памяти, на которое будет установлена ОС. Вывести список устройств внутренней памяти можно командой:

```
fdisk -l
```

Имя устройства отображается в строке `Диск /dev/...`.

Пример

```
Диск /dev/mmcblk1
```

3) в ходе копирования файлов ОС во внутреннюю память устройства на запросы системы о выполнении разметки памяти или удалении имеющейся информации ответить `Y`;

4) выполнить в терминале команду перезагрузки устройства:

```
reboot
```

5) в момент перезагрузки устройства отсоединить установочный USB-носитель.

После перезагрузки устройства при первом запуске ОС требуется выполнить первоначальную настройку в соответствии с 2.6.3.

2.6.3. Настройка установленной ОС

При установке ОС в режиме «Мобильный» на устройство автоматически создается учетная запись администратора `administrator` с паролем по умолчанию. При первом входе в систему будет отображено окно для смены пароля по умолчанию для администратора.

Установленная на устройство ОС в режиме «Мобильный» предоставляет графический интерфейс ОС и набор программ, адаптированных для использования на устройствах с сенсорным экраном. Описание графического интерфейса в режиме «Мобильный» приведено в электронной справке («Документация — Графический интерфейс — Режим «Мобильный»).

Для вызова электронной справки в режиме «Мобильный» требуется на экране приложений перейти в список «Все (неадаптированные)» и запустить приложение «Помощь».

Для использования в режиме «Мобильный» некоторых программ из состава ОС, в том числе не адаптированных для работы на устройствах с сенсорным экраном, может потребоваться установка соответствующих пакетов из репозитория ОС.

2.6.3.1. Начальная настройка ОС

При первом входе в систему необходимо:

1) ознакомиться с лицензионным соглашением, доступным по QR-коду, установить флаг «Я принимаю условия лицензионного соглашения» и нажать **[Начать]**;

2) задать новый пароль системного администратора и нажать **[>]**;

3) в зависимости от приобретенной лицензии выбрать уровень защищенности системы и функции безопасности, доступные на данном уровне, затем нажать **[>]**;

4) нажать **[Сохранить]** и подтвердить сохранение настроек в отобразившемся диалоговом окне. После перезагрузки устройства выполнить вход в систему (см. электронную справку).

После установки ОС отображение меню загрузчика GRUB (и вход в него) по умолчанию заблокировано. Если будет разблокировано отображение меню загрузчика, то необходимо установить пароль на вход в меню загрузчика и на использование режима командной строки загрузчика.

2.6.3.2. Настройка виртуальной клавиатуры

Для приложений в режиме «Мобильный» по умолчанию используется виртуальная клавиатура из состава рабочего стола KDE Plasma. При этом могут возникать ошибки в работе виртуальной клавиатуры с приложениями X11, функционирующими через Xwayland.

Для устранения ошибок возможно использовать виртуальную клавиатуру `fly-vkbd`. Для использования клавиатуры `fly-vkbd` определенным приложением требуется в файле `/etc/xdg/plasmamobilerc` в секции `[FlyVkbd]` для параметра `Applications` добавить название desktop-файла приложения.

Пример

```
[FlyVkbd]
Applications=chromium,yandex-browser
```

2.6.4. Обновление ОС

Обновление ОС в режиме «Мобильный» выполняется одним из следующих способов:

1) из обновленного образа ОС в формате IMG в соответствии с 2.6.4.1;
2) с загрузочного USB-носителя в соответствии с 2.6.4.2;
3) из источников (образов ISO и сетевых репозиториев) в соответствии с 2.6.4.3. Для указания источников рекомендуется использовать файл `/etc/apt/sources.list`.

ВНИМАНИЕ! При обновлении из образа IMG либо с загрузочного USB-носителя будут сохранены только каталоги `/opt` и `/home`, а остальные данные будут удалены. Настройки системы будут установлены по умолчанию, в том числе будет установлен уровень защищенности «Воронеж».

2.6.4.1. Обновление ОС из обновленного образа

Обновление ОС из образа выполняется путем копирования на внутреннюю память устройства файлов из обновленного образа ОС, при этом существующие на устройстве каталоги `/opt` и `/home` будут сохранены. Остальные данные будут удалены. Настройки системы будут установлены по умолчанию, в том числе будет установлен уровень защищенности «Воронеж».

Обновленный образ ОС в формате IMG требуется либо скопировать в каталог `/home/administrator/update/`, либо указать в файле `/etc/astra-mobile-update/update.list` его размещение — в этом случае образ будет автоматически скопирован в каталог `/home/administrator/update/` при выполнении сценария автоматизации обновления.

Обновление выполняется с помощью сценария автоматизации `update`, для запуска которого требуется из каталога `/etc/astra-mobile-update/scripts` выполнить команду:

```
./update
```

При выполнении сценария автоматизации будет выведен нумерованный список доступных образов для обновления. Требуется выбрать нужный образ и указать его номер, затем начнется загрузка образа. По окончании загрузки будут проверены контрольные суммы образа. Если контрольные суммы совпадают, то будет выполнена перезагрузка и обновление ОС.

Если в ходе работы сценария автоматизации `update` будут выявлены ошибки, то сценарий выведет соответствующее сообщение и завершит работу. Необходимо устранить причину ошибки и выполнить сценарий повторно.

Сценарий автоматизации `update` возможно запустить с параметром `-p`, в качестве значения которого указать URI на размещение файла образа.

При обновлении с помощью сценария автоматизации `update` выполняются сценарии из каталога `/etc/astra-mobile-update/scripts`, осуществляющие подготовку устройства к обновлению и установку ОС с обновленного образа:

1) `check-battery` — проверка уровня заряда батареи устройства. Обновление возможно только при заряде батареи больше 70 % или при подключении устройства к источнику электропитания;

2) `check-space` — проверка свободного места на устройстве в каталоге `/home/administrator/update/`. Необходимо минимум 8 ГБ свободного места для загрузки обновленного образа;

3) `list-of-updates` — просмотр списка доступных обновлений из указанных в `/etc/astra-mobile-update/update.list` источников. В качестве источника может использоваться сетевой ресурс по протоколам HTTP, HTTPS, FTP или локальный каталог, в котором размещен файл с обновленным образом в формате IMG;

4) `calculate-remote-update-size` — вычисление размера файла на сетевом ресурсе. Если свободного места на устройстве не достаточно для данного файла, то возможно прервать обновление;

5) `download-update-file` — копирование на устройство файла образа, указанного в качестве параметра. Файл образа будет сохранен в `/home/administrator/update/`. Также автоматически загружаются файлы `*.img.gost` и `*.img.md5` с контрольной суммой образа;

6) `check-sum` — проверка контрольной суммы указанного в параметре файла. Контрольная сумма сравнивается со значением в файле `*.img.gost`;

7) `list-of-other-sessions` — отображение списка других незавершенных сессий. При наличии в системе активных сессий кроме сессии, из которой выполняется обновление, обновление не будет выполнено;

8) `terminate-session` — завершение других сессии по требованию пользователя, в качестве параметра указывается ID из списка незавершенных сессий, полученного при выполнении `list-of-other-sessions`;

9) `run-update-script-from-img` — установка обновления из образа, указанного в качестве первого параметра. Вторым параметром требуется указать каталог расположения сценария обновления `/sbin/astra-mobile-system-update`.
ВНИМАНИЕ! Сценарий обновления `/sbin/astra-mobile-system-update` находится непосредственно в файле образа, указанного в первом параметре, и будет выполнен после распаковки обновления из образа.

После обновления ОС необходимо выполнить ее настройку в соответствии с 2.6.3.

2.6.4.2. Обновление ОС с USB-носителя

Обновление ОС с загрузочного USB-носителя выполняется путем копирования на внутреннюю память устройства файлов с USB-носителя, при этом существующие на устройстве каталоги `/opt` и `/home` будут сохранены. Остальные данные будут удалены. Настройки системы будут установлены по умолчанию, в том числе будет установлен уровень защищенности «Воронеж».

Для обновления ОС с загрузочного USB-носителя требуется:

1) подготовить USB-носитель в соответствии с 2.6.1;
2) загрузиться с USB-носителя в соответствии с 2.6.2.1;
3) в графическом меню установки ОС выбрать «Exit» и нажать **<Enter>**. Будет отображена командная строка и выполнен автоматический вход в систему с учетной записью суперпользователя `root`;
4) для обновления ОС выполнить команду копирования:

```
/opt/astra-mobile-install -u -d /dev/<имя_устройства>
```

где `<имя_устройства>` — имя устройства внутренней памяти, на которое будет установлена ОС. Вывести список устройств внутренней памяти можно командой:

```
fdisk -l
```

Имя устройства отображается в строке `Диск /dev/...`

Пример

```
Диск /dev/mmcblk1
```

5) в ходе копирования файлов ОС во внутреннюю память устройства на запросы системы о выполнении разметки памяти или удалении имеющейся информации ответить `Y`;
6) выполнить в терминале команду перезагрузки устройства:

```
reboot
```

7) в момент перезагрузки устройства отсоединить установочный USB-носитель.

После обновления ОС необходимо выполнить ее настройку в соответствии с 2.6.3.

2.6.4.3. Обновление ОС из источников

При обновлении ОС из источников выполняется обновление из сетевых репозиториев или из образов ISO, указанных в файле `/etc/apt/sources.list`.

Обновление может быть выполнено:

1) с помощью инструмента командной строки `astra-update`, описание инструмента приведено в `man astra-update`;

2) в графическом интерфейсе, порядок обновления приведен в электронной справке («Документация — Графический интерфейс — Режим «Мобильный»).

3. СИСТЕМНЫЕ КОМПОНЕНТЫ

3.1. Управление устройствами

3.1.1. Типы устройств

В ОС существует два типа устройств:

1) блочные устройства с произвольным доступом — данные, записанные в такие устройства, могут быть прочитаны (например, жесткие диски);

2) символьные устройства с последовательным или произвольным доступом — данные, записанные в такие устройства, не могут быть прочитаны (например, последовательные порты).

Каждое поддерживаемое устройство представляется в ФС файлом устройства. При выполнении операций чтения или записи с файлом устройства происходит обмен данными с устройством, на которое указывает этот файл. Данный способ доступа к устройствам позволяет не использовать специальные программы (а также специальные методы программирования, такие как работа с прерываниями).

Файлы устройств располагаются в каталоге `/dev`, для вывода списка файлов выполнить команду `ls`. При выполнении команды с параметром `-l` на экран монитора будет выведен список файлов с указанием в первой колонке типа файла и прав доступа к нему. Первый символ в первой колонке указывает на тип файла:

- `c` — символьное устройство;
- `b` — блочное устройство;
- `d` — каталог;
- `l` — символическая ссылка;
- «`-`» (дефис) — обычный файл.

Пример

Просмотр информации о файле, соответствующем звуковому устройству

```
ls -l /dev/dsp
```

Вывод команды:

```
crw-rw---- 1 root audio 14, 3 июл 1 13:05 /dev/dsp
```

Описание команды `ls` приведено в `man ls`.

Наличие файла устройства не означает, что данное устройство установлено в системе. Например, наличие файла `/dev/sda` не означает, что на компьютере установлен жесткий диск SCSI. Это предусмотрено для облегчения установки программ и нового оборудования, т. к. исключает необходимость поиска нужных параметров и создания файлов для новых устройств.

3.1.2. Жесткие диски

При администрировании дисков могут возникнуть задачи по разделению жесткого диска на разделы, созданию и монтированию ФС, форматированию диска и др.

Разделение жесткого диска может использоваться для хранения разных операционных систем на одном жестком диске, для хранения пользовательских и системных файлов в разных дисковых разделах. Разделение жесткого диска упрощает резервное копирование и восстановление, а также повышает защищенность системных файлов от повреждений.

Для использования диска или раздела необходимо создать на нем ФС.

Для штатного доступа к данным, находящимся в ФС, необходимо выполнить монтирование ФС. Монтирование выполняется с целью формирования единой структуры каталогов, обеспечения буферизации дисков и работы с виртуальной памятью.

Монтирование может выполняться как автоматически, так и вручную. Монтируемые вручную ФС должны быть размонтированы также вручную.

Центральный процессор и жесткий диск обмениваются информацией через дисковый контроллер. Это упрощает схему обращения и работы с диском, т. к. контроллеры для разных типов дисков могут быть построены с использованием единого интерфейса для связи с компьютером.

Каждый жесткий диск представлен отдельным файлом устройства в каталоге `/dev`:

- `/dev/hda` и `/dev/hdb` — для первого и второго диска, подключенного по IDE шине;
- `/dev/sda`, `/dev/sdb` и т. д. — для дисков, использующих SCSI или SATA-интерфейс.

3.1.3. Разделы жесткого диска

Весь жесткий диск может быть разделен на несколько дисковых разделов, при этом каждый раздел в системе представлен как отдельный диск. Разделение используется, например, при работе с двумя операционными системами на одном жестком диске. При этом каждая операционная система использует для работы отдельный дисковый раздел и не взаимодействует с другими. Таким образом, две различные системы могут быть установлены на одном жестком диске.

3.1.3.1. Разбиение жесткого диска

Главная загрузочная запись MBR (Master Boot Record) диска содержит место для четырех основных (первичных) разделов, пронумерованных от 1 до 4.

Если необходимо добавить еще разделы на диск, то следует преобразовать основной раздел в расширенный (extended). Далее расширенный раздел разделяется на один или несколько логических разделов с номерами от 5 до 15. Логические разделы функционируют так же, как и основные, различие состоит в схеме их создания.

При установке ОС разбиение жесткого диска (дисков) осуществляется средствами программы-установщика. При работе с ОС для разбиения жесткого диска на разделы используется инструмент командной строки `fdisk`.

Каждый раздел должен содержать четное количество секторов, т. к. в ОС используются блоки размером в 1 КБ, т. е. два сектора. Нечетное количество секторов приведет к тому, что последний из них будет не использован. Это ни на что не влияет, но при запуске `fdisk` будет выдано предупреждение.

При изменении размера раздела рекомендуется сначала сделать резервную копию раздела, затем удалить раздел, создать новый раздел и восстановить сохраненную информацию в новом разделе.

Описание инструмента `fdisk` приведено в `man fdisk`.

3.1.3.2. Файлы устройств и разделы

Каждому первичному и расширенному разделу соответствует отдельный файл устройства. Существует соглашение для имен подобных файлов, которое заключается в добавлении номера раздела к имени соответствующего файла устройства. Разделы с 1 по 4 являются первичными либо один из этих разделов является расширенным. Разделы с 5 по 15 являются логическими, на которые разбивается расширенный раздел. Например, `/dev/hda1` соответствует первому первичному разделу первого IDE-диска, а `/dev/sdb7` — третьему логическому разделу второго диска с интерфейсом SCSI или SATA.

3.1.4. Форматирование

Форматирование — это процесс записи специальных отметок на магнитную поверхность, которые используются для разделения дорожек и секторов. Новый диск не может использоваться без предварительного форматирования. Для IDE- и некоторых SCSI-дисков форматирование выполняется при их изготовлении и обычно не требуется повторение этой процедуры.

3.1.5. Программная организация дисковых разделов в RAID и тома LVM

В ядро ОС встроена программная реализация технологии RAID (уровни RAID 0, RAID 1, RAID 5 и их сочетания). Команда `mdadm` предоставляет административный интерфейс для создания и управления массивами RAID.

После создания массива RAID его устройство, например `/dev/md0`, используется также, как и `/dev/hda1` или `/dev/sdb7`.

Том LVM, с точки зрения ядра системы, использует унифицированные механизмы VFS и не нуждается в специальных конфигурациях ядра. В ОС обеспечивается полнофункциональное управление томами LVM, которое осуществляется стеком команд управления.

LVM обеспечивает более высокий уровень абстракции, чем традиционные диски и разделы Linux. Это позволяет добиться большей гибкости при выделении пространства для хранения данных. Логические тома можно перемещать с одного физического устройства на другое, а их размер изменять. Физические устройства можно добавлять и удалять. Томам, управляемым посредством LVM, можно назначать любые текстовые названия, например `database` или `home`, а не служебные `sda` или `hda`.

3.1.6. Разделы диска в режиме «Мобильный»

При установке ОС в режиме «Мобильный» разметка внутренней памяти устройства выполняется автоматически, при этом применяется таблица разделов GUID (GPT).

После установки ОС в режиме «Мобильный» на устройстве доступны следующие разделы:

1) `/boot/efi` — загрузочная область EFI. Размер раздела 100 МБ, тип ФС vfat;

2) `/boot` — содержит необходимую информацию для загрузки системы: ядро, образ initrd, файлы загрузчика. Размер раздела 500 МБ, тип ФС ext4;

3) `recovery` — размещение образа для восстановления ОС, который используется при сбросе настроек ОС на значения по умолчанию. Размер раздела 8 ГБ, тип ФС ext4;

4) том LVM с разделами:

   а) корневой каталог (обозначается символом «/». Размер раздела 15 ГБ, тип ФС ext4;

   б) `/opt` — каталог для установки дополнительного ПО (например, текстовые и графические редакторы, средства антивирусной защиты, специальное программное обеспечение и т. п.). Каталоги дополнительного ПО должны иметь имя вида `/opt/<имя_вендора>/<название_ПО>`. Ярлыки дополнительного ПО должны быть установлены в `/opt/astra-mobile/menu`. При обновлении ОС каталог `/opt` не изменяется, соответственно, не требуется переустановка дополнительного ПО. Размер раздела 5 ГБ, тип ФС ext4;

в) /home — рабочие (домашние) каталоги пользователей, в т.ч. для размещения пользовательских данных приложений. При обновлении ОС каталог /home не изменяется, соответственно, пользовательские данные приложений будут сохранены. Размер раздела не менее 10 ГБ (при возможности расширяется после загрузки), тип ФС ext4.

3.2. Управление ФС

3.2.1. Общие сведения

Файловая система — это методы и структуры данных, которые используются ОС для хранения файлов на диске или его разделе.

Перед тем, как раздел или диск могут быть использованы для хранения информации (файлов), он должен быть инициализирован, а требуемые данные перенесены на этот диск. Этот процесс называется созданием ФС.

В ОС рекомендована к применению и используется по умолчанию ФС типа ext4, обеспечивающая поддержку длинных имен, символических связей, хранение мандатных атрибутов, возможность представления имен файлов русскими буквами. Дополнительно могут использоваться ФС ISO9660, FAT (MS-DOS), NTFS и др.

Все данные ОС состоят из множества файлов (программы, библиотеки, каталоги, системные и пользовательские файлы) и располагаются в ФС. Структура ФС имеет вид «перевернутого дерева», вершину которого называют корневым каталогом, в системе обозначается символом «/».

В зависимости от параметров, заданных в процессе установки ОС, каталоги могут относиться к различным ФС.

После установки ОС файловая система может состоять, например, из следующих каталогов:

- /bin (/usr/bin) — содержит исполняемые файлы, необходимые для работы системы. Многие команды ОС являются программами из этого каталога;
- /boot — содержит необходимую информацию для загрузки системы: ядро (ядра), образ initrd, файлы загрузчика;
- /dev — содержит файлы устройств (device files). С их помощью осуществляется доступ к физическим устройствам, установленным в системе;
- /root — рабочий (домашний) каталог суперпользователя;
- /tmp — используется для хранения временных файлов, создаваемых программами в процессе своей работы. При работе с программами, создающими много больших временных файлов, рекомендуется иметь отдельную ФС;

- /etc — содержит конфигурационные файлы ОС, в т. ч. файл паролей passwd и список ФС fstab, монтируемых при начальной загрузке. В этом же каталоге хранятся сценарии загрузки (startup scripts), список узлов (hosts) с их IP-адресами и другие данные о конфигурации;

- /lib (/usr/lib) — содержит разделяемые библиотеки, используемые программами во время своей работы. Применяя разделяемые библиотеки, хранящиеся в общедоступном месте, можно уменьшить размер программ за счет повторного использования одного и того же кода;

- /proc — является псевдофайловой системой и используется для чтения из памяти информации о системе;

- /sbin (/usr/sbin) — содержит исполняемые файлы, используется для системного администрирования и требующие для запуска права суперпользователя);

- /usr — содержит программы и данные, не подлежащие изменению. Каталог /usr и его подкаталоги необходимы для функционирования ОС, т. к. содержат наиболее важные программы. Данный каталог почти всегда является отдельной ФС;

- /var — содержит изменяемые файлы, например log-файлы и др.;

- /home — содержит рабочие (домашние) каталоги пользователей. Рекомендуется создавать в качестве отдельной ФС, чтобы обеспечить пользователям достаточное пространство для размещения своих файлов. Если пользователей в системе много, возможно разделить этот каталог на несколько ФС. Например, можно создать подкаталоги /home/staff и /home/admin соответственно для персонала и администраторов, установить каждый подкаталог как отдельную ФС и уже в них создавать рабочие каталоги пользователей.
В рабочих каталогах пользователей также содержатся некоторые конфигурационные файлы, которые являются скрытыми и изменяются редко. Файл становится скрытым, если поставить точку в начале имени файла. При выводе списка файлов командой ls для отображения в том числе скрытых файлов использовать параметр -a:

```
ls -a
```

Для обеспечения совместной работы пользователей в ОС создаются автоматически при установке совместно используемые каталоги, доступ к которым разрешен всем пользователям:

- /tmp — каталог временных файлов. Содержимое каталога не сохраняется после перезагрузки ОС;

- /var/tmp — каталог временных файлов. Содержимое каталога сохраняется после перезагрузки ОС;

- /dev/shm — каталог разделяемой памяти, используется для обмена временными рабочими данными через разделяемую оперативную память. Содержимое каталога не сохраняется после перезагрузки ОС;

- `/run/mount` — каталог временного монтирования пользовательских устройств, используется для автоматического монтирования с помощью сценариев подключаемых пользовательских устройств;
- `/var/cache` — каталог для кеширования данных.

Также при установке дополнительного ПО могут создаваться собственные совместно используемые каталоги данного ПО.

Совместно используемые каталоги предназначены для создания в них файловых объектов, доступных всем пользователям. Применяемый в ОС механизм дискреционного управления доступом позволяет всем пользователям выполнять создание файловых объектов в совместно используемые каталогах, а также поиск принадлежащих другим пользователям файловых объектов в совместно используемых каталогах. Чтение и изменение не принадлежащих пользователю файловых объектов ограничивается дискреционными правами доступа, заданным для данного объекта. Дополнительно применяется специальное ограничение дискреционного доступа sticky-бит, запрещающее удалять и переименовывать не принадлежащие пользователю файловые объекты.

3.2.2. Создание

ФС создается при помощи команды `mkfs`. Команда запускает требуемую программу в зависимости от типа создаваемой ФС. Тип ФС задается параметром `-t`.

Пример

```
mkfs -t ext2 /dev/hdb1
```

Описание команды приведено в `man mkfs`.

3.2.3. Монтирование

Перед началом работы с ФС она должна быть смонтирована. Так как все файлы в ОС принадлежат одной структуре каталогов, то монтирование обеспечивает работу с ФС как с каталогом, называемым точкой монтирования. При этом ОС выполняет действия, обеспечивающие функционирование монтируемой ФС.

Перед монтированием ФС к дереву каталогов ОС необходимо убедиться, что существует каталог (точка монтирования), в который будет осуществляться монтирование ФС, иначе монтирование завершится неудачно.

После успешного монтирования ФС в каталог в нем появятся все файлы и подкаталоги ФС. В противном случае каталог будет пустым.

Если использовать в качестве точки монтирования непустой каталог, то его содержимое станет недоступно до размонтирования ФС. Поэтому рекомендуется для монтирования разделов/устройств создавать отдельные каталоги. Обычно они располагаются в `/mnt` и `/media`.

Для получения информации об имеющихся в ОС файловых системах используется инструмент командной строки `df`. Описание инструмента приведено в `man df`.

3.2.3.1. mount

В ОС для монтирования ФС используется инструмент командной строки `mount`. По умолчанию в целях обеспечения безопасности информации использовать инструмент `mount` может только администратор.

Синтаксис:

```
mount [параметр[параметр]] [<устройство>] [<точка_монтирования>]
```

где `<устройство>` — устройство, которое необходимо примонтировать;
`<точка_монтирования>` — имя каталога, в который требуется примонтировать устройство.

Параметры, дополнительно используемые с инструментом `mount`, приведенные в таблице 6.

Таблица 6

Параметр	Описание
`-f`	Имитировать монтирование ФС. Выполняются все действия, кроме системного вызова для монтирования ФС
`-v`	Вывести подробный отчет о действиях, выполняемых командой
`-w`	Примонтировать ФС с доступом для чтения и записи
`-r`	Примонтировать ФС с доступом только для чтения
`-n`	Выполнить монтирование без записи в файл `/etc/mtab`
`-t <тип_ФС>`	Задать тип монтируемой ФС
`-a`	Подключить все ФС, перечисленные в `/etc/fstab`
`-o <параметр>`	Задать параметры монтирования ФС, параметры в списке разделяются запятыми. Список возможных параметров приведен в `man mount`

Если необходимый параметр не указан, `mount` попытается определить его по файлу `/etc/fstab`.

Примеры:

1. Монтирование раздела жесткого диска `/dev/hdb3` в каталог `/mnt`:

```
mount /dev/hdb3 /mnt
```

2. Монтирование всех ФС типа NFS, перечисленных в файле `/etc/fstab`:

```
mount -vat nfs
```

Если правильно примонтировать ФС не удается, то для получения отчета о результатах выполнения команды `mount` выполнить команду:

```
mount -vf <устройство> <точка_монтирования>
```

В данном случае команда выполняет все действия, кроме монтирования, и выводится подробный отчет о каждом шаге выполнения команды.

Описание команды `mount` приведено в `man mount`.

3.2.3.2. fstab

В конфигурационном файле `/etc/fstab` указываются ФС для монтирования и перечисляются параметры их монтирования.

В файле `/etc/fstab` каждой ФС соответствует запись в одной строке. Поля в строках разделяются пробелами или символами табуляции. В таблице 7 приведено описание полей файла `/etc/fstab`.

Таблица 7

Поле	Описание
`<file system>` (файловая система)	Подключаемое блочное устройство или удаленная ФС
`<mount point>` (точка монтирования)	Каталог монтирования ФС. Чтобы сделать систему невидимой в дереве каталогов (например, для файлов подкачки), используется слово `none`
`<type>` (тип)	Указывает тип монтируемой ФС
`<options>` (параметры монтирования)	Список разделенных запятыми параметров для монтируемой ФС. Должен содержать, по крайней мере, тип монтирования. Более подробную информацию см. в руководстве `man` команды `mount`
`<dump>` (периодичность резервного копирования)	Указывает, как часто следует выполнять резервное копирование с помощью команды `dump`. Если в поле стоит значение `0`, то резервное копирование ФС не выполняется

Окончание таблицы 7

Поле	Описание
<pass> (номер прохода)	Задает порядок проверки целостности ФС при загрузке с помощью команды fsck. Для корневой ФС следует указывать значение 1, для остальных — 2. Если значение не указано, целостность ФС при загрузке проверяться не будет

Рекомендуется монтировать ФС во время загрузки через /etc/fstab, без использования команды mount. Далее приведен пример файла fstab.

Пример

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda11 during installation
UUID=a50cefb7-a198-4240-b198-581200027898 / ext4 errors=remount-ro,
     secdel=2 0 1
# /home was on /dev/sda10 during installation
UUID=c94bba8d-95d4-467b-b3e0-2cd7f92c3355 /home  ext4  usrquota,secdelrnd
     0 2
# swap was on /dev/sda5 during installation
UUID=ce71b251-2405-4eed-8130-5f92a56b67ac none    swap    sw       0 0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
```

Комментарии в файле начинаются с символа #.

В файле fstab параметр defaults поля <options> указывает, что при монтировании ФС будет применен набор параметров по умолчанию, а именно — ФС будет примонтирована с разрешенным доступом для чтения и записи; она должна рассматриваться как отдельное блочное устройство; весь файловый ввод-вывод должен выполняться асинхронно; разрешено выполнение программных файлов; ФС может монтироваться с помощью команды mount -a; биты UID и GID файлов в ФС интерпретируются; обычным пользователям не разрешено подключать данную ФС.

Раздел подкачки (в примере /dev/sda5) используется ядром ОС для организации виртуальной памяти. Он должен присутствовать в файле /etc/fstab для информирования системы о его местонахождении. Чтобы он не отображался в дереве каталогов, точка монтирования в

файле fstab указывается none. Кроме того, разделы подкачки монтируются с параметром sw.

Псевдофайловая система /proc указывает на информационное пространство процессов в памяти. Соответствующий физический раздел для нее отсутствует.

ФС VFAT также можно монтировать автоматически. Раздел /dev/sdb1 — это первый раздел второго жесткого диска SCSI. Он монтируется как раздел VFAT, где vfat указывается в качестве типа ФС, /win — в качестве точки монтирования.

Для получения полной информации о допустимых в файле /etc/fstab параметрах см. руководство man fstab.

3.2.4. Размонтирование

Для размонтирования ФС используется инструмент командной строки umount. Размонтирование может понадобиться для проверки и восстановления ФС с помощью команды fsck. Удаленные ФС размонтируются в случае неполадок в сети.

Инструмент umount имеет следующий синтаксис:

```
umount <устройство>
umount <точка_монтирования>
umount -a
umount -t <тип_ФС>
```

где <устройство> — устройство, которое необходимо размонтировать;
<точка_монтирования> — имя каталога, от которого необходимо отмонтировать;
-a — размонтировать все ФС;
-t — размонтировать только ФС указанного типа <тип_ФС>.

Инструмент umount не размонтирует ФС, если она используется в текущий момент. Например, если ФС смонтировать в /mnt и выполнить команды:

```
cd /mnt
umount /mnt
```

то появится сообщение об ошибке, т. к. ФС занята. Перед размонтированием /mnt необходимо перейти в каталог другой ФС.

Для принудительного размонтирования устройства, независимо от его использования, можно воспользоваться параметром -f:

```
umount -f /cdrom
```

Для размонтирования и извлечения из устройств сменных носителей информации используется инструмент командной строки `eject`.

Инструмент командной строки `fuser` отображает сведения о процессах, использующих ФС:

```
fuser -v <точка_монтирования>
```

Для завершения всех процессов, использующих ФС, можно воспользоваться командой:

```
fuser -km <точка_монтирования>
```

Описание инструментов `umount`, `eject` и `fuser` приведено, соответственно, в `man umount`, `man eject` и `man fuser`.

3.3. Управление пользователями

3.3.1. Работа с пользователями

Управление пользователями заключается в добавлении и удалении пользователей, а также в определении их привилегий и предусматривает:

- добавление имен пользователей для возможности их работы в системе;
- создание или изменение паролей пользователей;
- определение их привилегий;
- создание и назначение рабочих каталогов;
- определение групп пользователей;
- удаление имен пользователей.

Каждый пользователь должен иметь уникальное регистрационное имя, дающее возможность идентифицировать пользователя и избежать ситуации, когда один пользователь может стереть файлы другого. Кроме того, каждый пользователь должен иметь свой пароль для входа в систему.

3.3.1.1. Добавление пользователя

Для добавления пользователя применяется инструмент командной строки `adduser` с указанием в качестве параметра имени добавляемого пользователя:

```
adduser <имя_пользователя>
```

Команда `adduser` добавляет пользователя, создает рабочий каталог пользователя, создает почтовый ящик, а также копирует файлы, имена которых начинаются с точки, из каталога `/etc/skel` в рабочий каталог пользователя. Каталог `/etc/skel` должен содержать все

файлы-шаблоны, которые необходимы каждому пользователю. Обычно это персональные конфигурационные файлы для настройки оболочки, например `.profile`, `.bashrc` и `.bash_logout`.

При добавлении пользователя в систему в файле `/etc/passwd` добавляется запись вида:

```
login_name:encrypted_password:user_ID:group_ID:user_information:
     login_directory:login_shell
```

Описание полей записи приведено в таблице 8.

Таблица 8

Поле	Описание
`login_name`	Регистрационное имя учетной записи пользователя (имя пользователя)
`encrypted_password`	Указатель на теневой файл паролей (`shadow`)
`user_ID`	Уникальный номер, используемый ОС для идентификации пользователя. Для локальных пользователей не должен превышать 2499
`group_ID`	Уникальный номер или имя, используемые для идентификации первичной группы пользователя. Если пользователь является членом нескольких групп, то он может в процессе работы менять группу (если это разрешено администратором)
`user_information`	Информация о пользователе, например его фамилия, имя и должность
`login_directory`	Рабочий каталог пользователя (в котором он оказывается после входа в систему)
`login_shell`	Оболочка, используемая пользователем после входа в систему (например, `/bin/bash`)

Описание файла `/etc/passwd` приведено в `man 5 passwd`.

Команда `adduser` представляет собой файл сценария `bash`, находящийся в каталоге `/usr/sbin`.

Для изменения информации о пользователе используется инструмент командой строки `chfn`.

Описание `adduser` и `chfn` приведено, соответственно, в `man adduser` и `man chfn`.

ВНИМАНИЕ! Для обеспечения штатной работы пользователя с сетевыми службами в системе должна быть явно задана его классификационная метка (диапазон уровней конфиденциальности и категорий конфиденциальности), даже если ему недоступны уровни и категории выше 0. Описание классификационной метки и порядок ее использования приведены в РУСБ.10015-01 97 01-1.

3.3.1.2. Установка пароля пользователя

Для установки пароля пользователя предназначена команда `passwd`. Задавать пароль необходимо для каждого пользователя. После входа в систему пользователь может изменить свой пароль. Для установки пароля пользователя выполнить следующее:

1) ввести команду и имя пользователя, например:

```
passwd ivanov
```

и нажать клавишу **<Enter>**;

2) после появления приглашения:

```
Новый пароль:
```

ввести пароль и нажать клавишу **<Enter>**;

3) ввести повторно пароль после появления соответствующего сообщения и нажать клавишу **<Enter>**.

Пароль будет преобразован и внесен в файл `/etc/shadow`.

ВНИМАНИЕ! Пароль рекомендуется создавать способом, максимально затрудняющем его подбор. Наиболее безопасный пароль состоит из случайной (псевдослучайной) последовательности букв, знаков препинания, специальных символов и цифр.

Описание команды приведено в `man passwd`.

Пример

Запись в файле `/etc/passwd`

```
ivanov:x:123:121:Petr Ivanov:/home/ivanov:/bin/bash
```

Второе поле записи содержит ссылку на пароль в преобразованном виде.

Примечание. Пароль пользователя не хранится в явном виде. Если пользователь забыл свой пароль, то администратор системы не может его восстановить. Для восстановления доступа пользователя в систему администратор может задать новый пароль для пользователя с помощью команды `passwd`.

3.3.1.3. Удаление пользователя

В ОС доступно несколько вариантов удалить пользователя:

- лишить пользователя возможности входа в систему;
- удалить учетную запись пользователя;

- удалить учетную запись пользователя и все его файлы и каталоги.

Лишение пользователя возможности входа в систему может быть использовано в случае его длительного перерыва в работе. На время отсутствия пользователя можно заблокировать его учетную запись с помощью команды:

```
usermod -L <имя_пользователя>
```

После выполнения команды вход в систему от имени указанного пользователя будет недоступен, при этом все пользовательские файлы и каталоги сохраняются.

Для разблокировки учетной записи необходимо выполнить команду:

```
usermod -U <имя_пользователя>
```

Одним из вариантов лишения пользователя возможности входа в систему может быть смена имени пользователя. При этом вход в систему под старым именем становится невозможным. Для этого необходимо выполнить команду:

```
usermod -l <новое_имя_пользователя> <имя_пользователя>
```

Примечание. Имена домашнего каталога и почтового ящика при изменении имени пользователя не меняются. Эти параметры должны быть изменены вручную.

Удаление учетной записи пользователя производится либо путем непосредственного редактирования файла `/etc/passwd`, либо с помощью команды:

```
deluser <имя_пользователя>
```

По умолчанию учетная запись удаляется без удаления домашнего каталога и файлов, принадлежащих удаляемому пользователю. Для удаления домашнего каталога может использоваться дополнительный параметр `--remove-home`, а для поиска и удаления всех файлов, принадлежащих удаляемому пользователю, — параметр `--remove-all-files`.

Также удаление пользователя, его домашнего каталога и файлов могут быть выполнены вручную путем последовательного выполнения следующих команд:

1) для полного удаления пользователя и всех его файлов из системы выполнить команду:

```
find / -user <имя_пользователя> -exec rm -r {} \;
```

2) удалить запись о пользователе из файла `/etc/passwd`;

3) для удаления файлов, не принадлежащих ни одному пользователю в системе, выполнить команду:

```
find / -nouser -exec rm -r {} \;
```

Описание команд приведено в `man usermod`, `man deluser` и `man find`.

3.3.1.4. Неудачный вход в систему

Инструмент командной строки `faillog` используется для управления счетчиком неудачных попыток и их ограничения. Также инструмент отображает журнал неудачных попыток входа в систему (файл `/var/log/faillog`).

При запуске `faillog` без параметров выводятся записи `faillog` только тех пользователей, у которых имеется хотя бы одна неудачная попытка входа.

Предельное число попыток входа для каждой учетной записи равно 10. Для сброса счетчика неудачных попыток входа необходимо использовать параметр `-r`.

Описание команды `faillog` и файла `/var/log/faillog` приведено, соответственно, в `man faillog` и `man 5 faillog`.

3.3.2. Работа с группами

Каждый пользователь является членом группы. Разным группам можно назначить разные возможности и привилегии.

Пользователь может состоять в нескольких группах и переходить из одной в другую в процессе работы.

Информация о группах содержится в файле `/etc/group`. Описание файла `/etc/group` приведено в `man 5 group`.

3.3.2.1. Добавление

Информация о группах в файле `/etc/group` содержится в формате:

```
Admin:x:21:user1,user2,user3
```

где `Admin` — имя группы;
`x` — пароль в преобразованном виде. Если поле пустое, то пароль не требуется;
`21` — уникальный идентификатор группы;
`user1, user2, user3` — участники группы.

Добавление группы производится с помощью команды:

```
addgroup <имя_группы>
```

Также новую группу можно создать путем редактирования файла `/etc/group`, добавив в нем строку с необходимой информацией о группе.

ВНИМАНИЕ! Каждой группе присваивается уникальный идентификационный номер и ОС при работе учитывает номер группы, а не имя. Поэтому, если присвоить двум группам одинаковый номер, ОС будет воспринимать две группы как одну и ту же.

Описание инструмента `addgroup` и файла `/etc/group` приведено, соответственно, в `man addgroup` и `man 5 group`.

3.3.2.2. Удаление

Удаление группы производится с помощью команды:

```
delgroup <имя_группы>
```

Также удалить группу можно путем редактирования файла `/etc/group`, удалив записи о группе.

Описание команды `delgroup` приведено в `man delgroup`.

3.3.3. Рабочие каталоги пользователей

Рабочие каталоги пользователей на одном компьютере следует размещать в отдельном каталоге верхнего уровня (по умолчанию — `/home`). Если пользователей много, то оптимально разделить их домашние каталоги по группам (подразделениям), например, `/home/hr` (отдел персонала) `/home/admins`, `/home/buhg` и т. д.).

Таким образом, рабочие каталоги будут логически сгруппированы, что в дальнейшем облегчит администрирование системы.

3.4. Перезагрузка и выключение

Перезагрузка необходима в следующих случаях:

1) при подключении нового устройства или если работающее устройство «зависает» и его невозможно сбросить;

2) при модификации файла конфигурации, который используется только при начальной загрузке, т. к. для того чтобы изменения вступили в силу, необходимо загрузить систему заново;

3) если система «не отвечает» и невозможно зарегистрироваться и определить причину ошибки.

Перезагрузку можно выполнить одним из способов:

1) использовать команду `shutdown` с параметром `-r` в соответствии с 3.4.1;
2) использовать команду `reboot` в соответствии с 3.4.2;
3) использовать команду `init 6`.

Выключение компьютера предполагает корректное завершение работы системы (останов), позволяющее избежать потерь информации и сбоев ФС.

Выключение компьютера можно выполнить несколькими способами:

1) использовать команду `shutdown` (см. 3.4.1);
2) использовать команду `halt` (см. 3.4.2);
3) использовать команду `init 0`.

Работая с ОС, не рекомендуется выключать питание компьютера без предварительного завершения работы с использованием соответствующих инструментов ОС, т. к. ОС хранит информацию ФС в оперативной памяти и при отключении питания информация может быть потеряна, а ФС повреждена.

Выключение питания также может повредить жесткий диск, если установленный в системе жесткий диск перед отключением питания требует установки в соответствующее положение защитный переключатель либо выполнения парковки головок.

3.4.1. shutdown

Команда `shutdown` позволяет безопасно и корректно инициировать завершение работы системы, выключение, перезагрузку или возврат в однопользовательский режим.

В качестве параметра команды `shutdown` возможно задать время ожидания перед завершением работы системы. Во время ожидания команда посылает зарегистрированным пользователям через постепенно укорачивающиеся промежутки времени предупреждения о завершении работы системы. По умолчанию сообщения содержат информацию о завершении работы и времени, оставшемся до завершения работы. При желании администратор может добавить собственное сообщение, например с информацией о причине останова и о времени, в течение которого вход в систему будет невозможен.

Параметры команды `shutdown` позволяют задать определенное действие для компьютера: остановиться, перейти в однопользовательский режим или перезагрузиться. Дополнительно возможно указать, следует ли перед перезагрузкой проверить диски с помощью команды `fsck`.

Синтаксис команды:

```
shutdown [<параметр>] [<время>] [<сообщение>]
```

где <параметр> — параметр, определяющий действие команды (без параметра команда выполняет выключение компьютера);
<время> — время завершения работы системы в формате чч:мм. Значение может быть также задано в формате +m, где m — количество минут ожидания до завершения работы. Значение +0 может быть заменено словом now;
<сообщение> — сообщение, посылаемое всем пользователям, зарегистрированным в системе в момент запуска команды.

В таблице 9 перечислены основные параметры команды shutdown.

Таблица 9

Параметр	Описание
-k	Послать предупреждение без реального завершения работы системы
-r	Перезагрузить компьютер после завершения работы
-h	Выключить компьютер после завершения работы
-n	Не синхронизировать диски. Этот параметр следует использовать осторожно, т. к. могут быть потеряны или повреждены данные
-f	«Быстрая» перезагрузка. Создается файл /etc/fastboot, при наличии которого во время загрузки ОС не запускается программа fsck
-c	Отменить уже запущенный процесс завершения работы. Параметр <время> при этом не может быть использован

Описание команды приведено в man shutdown.

Команда shutdown посылает всем пользователям предупреждающее сообщение, затем ожидает заданное в командной строке время и посылает всем процессам сигнал SIGTERM. Далее вызывается команда halt или reboot — в зависимости от параметров командной строки.

3.4.2. halt и reboot

Команда halt выполняет все основные операции, необходимые для останова системы. Для вызова команды выполнить в командной строке:

```
halt
```

или

```
shutdown -H
```

Команда регистрирует останов, уничтожает несущественные процессы, осуществляет системный вызов `sync`, ожидает завершения операций записи ФС, а затем прекращает работу ядра.

При выполнении команды `halt` с параметром `-n`:

```
halt -n
```

вызов `sync` подавляется. Данная команда используется после исправления корневого раздела программой `fsck` для того, чтобы ядро не могло затереть исправления старыми версиями суперблока.

При выполнении команды `halt` с параметром `-q`:

```
halt -q
```

инициируется немедленный останов, без синхронизации, уничтожения процессов и записи в файлы регистрации.

Команда `reboot` выполняет все основные операции, необходимые для останова системы (аналогично команде `halt`), а затем перезагружает компьютер с нуля. Для вызова команды выполнить в командной строке:

```
reboot
```

или

```
shutdown -r
```

Описание команд `halt` и `reboot` приведено в `man halt` и `man reboot` соответственно.

4. СИСТЕМНЫЕ СЛУЖБЫ, СОСТОЯНИЯ И КОМАНДЫ

4.1. Системные службы

Службы — это специальные программы, выполняющие различные служебные функции. Обычно службы запускаются автоматически при наступлении определенного события (например, при загрузке ОС) и выполняются в фоновом режиме.

4.1.1. Управление службами

В среде ОС для управления службами, точками монтирования и т. п. применяется системный менеджер systemd. Менеджер systemd обеспечивает параллельный запуск служб в процессе загрузки ОС, использует сокеты и активацию D-Bus для запускаемых служб, предлагает запуск демонов по необходимости, отслеживает запуск служб, поддерживает мгновенные снимки и восстановление состояния системы, монтирование и точки монтирования, а также внедряет основанную на зависимостях логику контроля процессов сложных транзакций.

Отличительной особенностью systemd является использование контрольных групп Linux, обеспечивающих иерархическую структуризацию служб: любая запущенная служба помещается в отдельную контрольную группу с уникальным идентификатором. Служба, запуская другую зависимую службу, становится родительской службой, а зависимая служба — дочерней. Дочерняя служба автоматически включается в группу с тем же идентификатором, что и родительская. Непривилегированные службы не могут изменить свое положение в иерархии. При штатном завершении работы родительской службы будут завершены и все ее дочерние службы.

Информация о менеджере systemd также приведена в `man systemd`.

Описание использования менеджера systemd для управления доступом приведено в РУСБ.10015-01 97 01-1.

Менеджер systemd оперирует специально оформленными файлами конфигурации — юнитами (unit). Каждый юнит отвечает за конкретную служб (`*.service`), точку монтирования (`*.mount`), устройство (`*.device`), файл подкачки (`*.swap`), сокет (`*.socket`) и т. д.

Юниты менеджера systemd располагаются в каталогах `/etc/systemd/system`, `/run/systemd/system`, `/usr/lib/systemd/system`, а также в пользовательских каталогах.

Приоритет выполнения юнитов зависит от их расположения:

- `/usr/lib/systemd/system/` — юниты из установленных пакетов, имеют минимальный приоритет;

- /run/systemd/system/ — юниты, созданные в режиме рантайм. Данные юниты имеют приоритет выше, чем юниты из установленных пакетов;

- /etc/systemd/system/ — юниты, созданные и управляемые администратором. Данные юниты имеют приоритет выше, чем юниты, созданные в режиме рантайм.

Также в ОС доступен механизм управления службами systemV, сохраненный для обеспечения совместимости. Менеджер systemV управляет сценариями запуска в каталогах /etc/init.d, /etc/rc{0-6,S}.d.

Таким образом, администратор ОС может использовать два инструмента для управления службами:

1) /usr/sbin/service (команда service) — устаревший инструмент, работающий только со службами, сценарии управления которых находятся в каталогах /etc/init.d, /etc/rc{0-6,S}.d;

2) /bin/systemctl (команда systemctl) — инструмент для управления всеми службами.

Инструменты обеспечивают интерфейс пользователя с юнитами/сценариями. Юниты/сценарии обеспечивают интерфейс управления службами, предоставляя администратору параметры для запуска, остановки, перезапуска, запроса состояния, а также для других действий со службой.

Сценарии systemV могут иметь произвольный набор параметров управления, поэтому предусмотрена возможность проверки доступных параметров с помощью команды service, выполненной с названием сценария в качестве параметра.

Пример

Команда запроса доступных параметров для службы cron:

```
sudo /usr/sbin/service cron
```

Результат выполнения команды:

```
[info] Usage: /etc/init.d/cron {start|stop|status|restart|reload|
     force-reload}
```

Команда service выводит информацию только о службах, сценарии которых находятся в каталоге /etc/init.d. Проверить текущее состояние служб можно с помощью параметра --status-all команды service:

```
sudo /usr/sbin/service --status-all
```

Пример

Вывод команды `/usr/sbin/service --status-all` проверки состояния служб:

```
[ + ]  acpi-support
[ + ]  acpid
[ - ]  anacron
...
```

Юниты systemd имеют фиксированный набор параметров, оформленных в виде параметров команды `systemctl`, например, `start, stop, reload, restart` и т. д.

Для просмотра списка установленных юнитов выполнить команду:

```
sudo systemctl list-unit-files
```

Для просмотра списка запущенных юнитов выполнить команду:

```
systemctl list-units
```

или для просмотра списка запущенных юнитов определенного типа использовать данную команду с параметром `-t <тип_юнита>`:

```
systemctl list-units -t  <тип_юнита>
```

Для получения списка юнитов, которые менеджер systemd загрузил и пробовал загрузить, не зависимо от их состояния в текущий момент, используется команда `systemctl` с параметром `-a`.

Пример

Для получения списка юнитов типа `service`, которые загрузил и пробовал загрузить менеджер systemd, выполнить команду:

```
systemctl -t service -a
```

Результат выполнения команды:

```
UNIT                    LOAD      ACTIVE    SUB     DESCRIPTION
acpi-support.service    loaded    active    exited  LSB: Start some power
? apache2.service       masked    inactive  dead    apache2.service
? apparmor.service      not-found inactive  dead    apparmor.service
assistant.service       loaded    active    running Assistant remote
     control
```

Основные параметры для использования с инструментом командной строки `systemctl` приведены в таблице 10.

Таблица 10

Параметр	Описание
`systemctl start <юнит>`	Незамедлительно запустить юнит
`systemctl stop <юнит>`	Незамедлительно остановить юнит
`systemctl restart <юнит>`	Перезапустить юнит
`try-restart <юнит>`	Перезапустить (не запускать неработающие) юниты
`systemctl reload <юнит>`	Перезагрузить настройки юнита
`systemctl status`	Вывести общую информацию о состоянии системы и список юнитов, которым соответствуют запущенные процессы. При запуске команды с именем юнита будет выведена информация о статусе данного юнита
`systemctl cat <юнит>`	Показать содержимое юнита
`systemctl is-enabled <юнит>`	Проверить включение юнита в автозапуск при загрузке системы
`systemctl enable <юнит>`	Добавить юнит в автозапуск при загрузке системы
`systemctl disable <юнит>`	Удалить юнит из автозапуска при загрузке системы
`systemctl mask <юнит>`	Маскировать юнит для исключения возможности его запуска
`systemctl unmask <юнит>`	Снять маску юнита
`systemctl help <юнит>`	Показать страницу руководства `man` юнита (при наличии поддержки данной функции для указанного юнита)
`systemctl daemon-reload`	Перезагрузить systemd для поиска новых или измененных юнитов
`systemctl --failed`	Показать список юнитов, которые не были запущены из-за ошибки
`isolate <юнит или цель>`	Если указано имя юнита, то запускает этот юнит и все его зависимости, остановив все остальные службы. Если указано имя целевого состояния выполнения, то переводит систему в указанное состояние выполнения (имя состояния указывается без расширения `.target`)

Полное описание команды `systemctl` приведено в `man systemctl`.

4.1.2. Конфигурационные файлы systemd

При использовании менеджера systemd возможно как корректировать существующие юниты, так и создавать новые.

Юнит представляет собой `ini`-подобный файл, имя которого состоит из имени юнита и суффикса, определяющего тип юнита. В общем случае юнит-файл включает секции `[Unit]` и `[Install]`, а также секции, соответствующие конкретному типу юнита.

Секция `[Unit]` содержит описание юнита, а также информацию о зависимостях при запуске юнита. Основные параметры секции:

1) `Description=` — описание юнита;

2) `Wants=` — зависимость требования запуска. Требование исходного юнита запустить юнит, указанный в параметре. При этом результат запуска юнита, указанного в параметре, не влияет на запуск исходного юнита. При отсутствии параметров `After=` и `Before=` юниты будут запущены одновременно;

3) `Requires=` — зависимость требования запуска. Требование исходного юнита запустить юнит, указанный в параметре. При этом ошибка запуска юнита, приведенного в параметре, приведет к ошибке запуска исходного юнита. При отсутствии параметров `After=` и `Before=` юниты будут запущены одновременно;

4) `After=` — зависимость порядка запуска. Дополнительный, но не обязательный параметр к параметрам `Wants=` и `Requires=`, указывающий на необходимость запуска исходного юнита только после запуска юнита, указанного в параметре. При этом если данный параметр используется с параметром `Wants=`, то исходный юнит будет запущен вне зависимости от результата запуска юнита, указанного в параметре;

5) `Before=` — аналогичен параметру `After=`, только определяет запуск исходного юнита до запуска юнита, указанного в параметре.

Секция `[Install]` содержит информацию об установке юнита. Используется командами `systemctl enable <юнит>` и `systemctl disable <юнит>`. Может содержать следующие параметры:

1) `Alias=` — список альтернативных имен юнита, разделенных пробелом. Имена должны иметь тот же суффикс, что и имя файла юнита. При использовании команды `systemctl enable` будут созданы символические ссылки из перечисленных имен на данный юнит.
ВНИМАНИЕ! Не все типы юнитов могут иметь альтернативные имена. Для типов `*.mount`, `*.slice`, `*.swap` и `*.automount` данный параметр не поддерживается;

2) `WantedBy=` — указывает на целевое состояние (см. 4.2), при котором запускается данный юнит. При использовании команды `systemctl enable` будет добавлена символическая ссылка в `<имя_состояния>.target`;

3) `Also=` — определяет список юнитов, которые будут добавлены в автозапуск или удалены из автозапуска вместе с данным юнитом.

Секция [Service] присутствует в юнитах службы и может содержать следующие параметры, определяющие запуск службы:

1) Type= — определяет тип запуска службы:

   а) simple — служба считается запущенной, когда завершился основной процесс службы (процесс, определенный в ExecStart=, считается основным процессом). Не рекомендуется использовать данный тип, если другие службы зависят от очередности при запуске данной службы. Исключение — активация сокета;

   б) forking — служба считается запущенной, когда основной (родительский) процесс службы создал дочерний процесс, при этом родительский процесс завершился. Дочерний процесс продолжает функционировать в качестве основного. Рекомендуется использовать данный тип для запуска классических демонов. Потребуется также задать значение параметра PIDFile= для отслеживания основного процесса;

   в) oneshot — похож на тип simple, используется для сценариев, которые завершаются после выполнения одного задания;

   г) notify — похож на тип simple, но служба запускается после отправки менеджеру systemd сигнала о своей готовности;

   д) dbus — похож на тип simple, но ожидает появления в системной шине DBus шины, указанной в BusName=;

   е) idle — менеджер systemd отложит выполнение службы и запустит ее после запуска остальных служб;

2) PIDFile= — расположение pid-файла службы;

3) ExecStart= — указывает на команду, которая должна быть выполнена при запуске службы;

4) ExecStop= — указывает на команды, которые должны быть выполнены для завершения службы, запущенной в ExecStart=;

5) ExecReload= — указывает на команду, которая должна быть выполнена для перезапуска службы;

6) Restart= — определяет перезапуск службы в случае самостоятельного или принудительного завершения основного процесса или при возникновении ошибки;

7) RemainAfterExit — позволяет считать службу активной даже в случае, если все ее процессы завершились. Значение по умолчанию no (нет).

Общие параметры, которые могут содержаться в секциях [Service], [Socket], [Mount], [Swap]:

1) WorkingDirectory= — рабочий каталог службы;

2) User= — пользователь, от имени которого будет запущена служба;

3) Group= — группа, от имени которой будет запущена служба;

4) `OOMScoreAdjust=` — приоритет завершения процесса при нехватке памяти, где 1000 — максимальное значение, означающее полный запрет на завершение процесса;

5) `KillMode=` — указывает на порядок завершения процессов данного юнита.

4.2. Системные (целевые) состояния

В systemd уровни запуска файлов реализованы в виде сгруппированных юнитов, представляющих целевое состояние (цель). Файлы, определяющие целевые состояния, хранятся в каталоге `/lib/systemd/system/` и имеют расширение имени `.target`. Для совместимости в ОС сохранено понятие «уровней выполнения». В стандартно установленной системе предусмотрено наличие шести системных уровней выполнения, каждому из которых соответствует целевое состояние.

Одна из целей назначается в качестве состояния по умолчанию, в которое переходит система после включения. В стандартно установленной ОС состоянием по умолчанию является `graphical.target` (уровень выполнения 5) — многопользовательский режим с графической оболочкой. Уровням выполнения 2, 3 и 4 соответствует цель `multi-user.target` (многопользовательский режим без графической оболочки), а целям `poweroff.target` (уровень выполнения 0) и `reboot.target` (уровень выполнения 6) соответствуют выключение и перезагрузка системы соответственно.

Проверить список соответствия состояний и уровней выполнения можно командой:

```
ls -la /lib/systemd/system/runlevel*
```

Пример вывода команды:

```
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel6.target -> reboot.target
```

Каждая цель имеет собственное имя вида `<имя_состояния>.target` и предназначена для конкретных задач. Одновременно могут быть активны несколько целей. Цели могут наследовать все службы других целей, добавляя к ним свои. В systemd также имеются цели, имитирующие общие уровни выполнения SystemV, поэтому для переключения между целевыми юнитами можно использовать команду:

```
telinit RUNLEVEL
```

Для определения доступных целевых состояний используется команда:

```
systemctl list-unit-files --type=target
```

Для определения активных целевых состояний используется команда:

```
systemctl list-units --type=target
```

Для перехода в целевое состояние используется команда:

```
systemctl isolate <имя_состояния>.target
```

или команда:

```
sudo init <уровень_выполнения>
```

Данные команды изменят только текущий уровень выполнения и их действие не повлияет на последующие загрузки системы.

Пример

Для перехода в целевое состояние командой `systemctl` выполнить:

```
systemctl isolate multi-user.target
```

Для перехода в целевое состояние командой `init` выполнить:

```
sudo init 3
```

Обе команды переведут систему в состояние `multi-user` (многопользовательский режим без графической оболочки), что соответствует третьему уровню выполнения. При этом будут запущены/остановлены все службы, указанные в соответствующем описании состояния.

Для просмотра целевого состояния по умолчанию, которое systemd использует сразу после загрузки системы, используется команда:

```
systemctl get-default
```

Для просмотра дерева зависимостей юнитов от цели выполнить команду:

```
systemctl list-dependencies <имя_состояния>.target
```

Для проверки текущего уровня выполнения выполнить команду:

```
sudo runlevel
```

Для изменения состояния системы, заданного по умолчанию, выполнить команду:

```
sudo systemctl set-default <имя_состояния>.target
```

В новое состояние по умолчанию система будет переведена после перезагрузки. Для принудительного перевода системы в нужное состояние без перезагрузки используется команда `systemctl` с параметром `isolate` и именем целевого состояния (имя состояния может быть указано без расширения `.target`). или команда `init` с указанием уровня выполнения.

Для обеспечения совместимости с более ранними реализациями помимо запуска/остановки юнитов, определенных в файлах `.target`, при переводе системы в другое целевое состояние systemd проверяет все файлы управления службами, имеющиеся в соответствующем целевому уровню выполнения каталоге `/etc/rc{0-6}.d/`, и запускает/останавливает соответствующие этим файлам собственные юниты или, если соответствующий юнит не обнаружен, автоматически генерирует юнит из файла управления и выполняет его.

Подробное описание данных команд и служб приведено на страницах руководства `man systemctl`, `man init`.

4.3. Системные команды

Основные системные команды ОС приведены в таблице 11.

Таблица 11

Команда	Назначение
`addgroup`	Создание новой учетной записи группы
`adduser`	Создание новой учетной записи пользователя
`ar`	Создание и работа с библиотечными архивами
`at`	Формирование или удаление отложенного задания
`awk`	Язык обработки строковых шаблонов
`bc`	Строковый калькулятор
`chfn`	Управление информацией учетной записи пользователя (имя, описание)
`chsh`	Управление выбором командного интерпретатора (по умолчанию — для учетной записи)
`cut`	Разбивка файла на секции, задаваемые контекстными разделителями

Продолжение таблицы 11

Команда	Назначение
delgroup	Удаление учетной записи группы
deluser	Удаление учетной записи пользователя и соответствующих файлов окружения
df	Вывод отчета об использовании дискового пространства
dmesg	Вывод содержимого системного буфера сообщений
du	Вычисление количества использованного пространства элементов ФС
echo	Вывод содержимого аргументов на стандартный вывод
egrep	Поиск строки (в т. ч. в файлах), содержащей заданное регулярное выражение
fgrep	Поиск строки (в т. ч. в файлах), содержащей заданный фиксированный шаблон
file	Определение типа файла
find	Поиск файла по различным признакам в иерархии каталогов
gettext	Получение строки интернационализации из каталогов перевода
grep	Поиск строки (в т. ч. в файлах), содержащей шаблон поиска
groupadd	Создание новой учетной записи группы
groupdel	Удаление учетной записи группы
groupmod	Изменение учетной записи группы
groups	Вывод списка групп
gunzip	Распаковка файла
gzip	Упаковка файла
hostname	Вывод и задание имени хоста
install	Копирование файла с установкой атрибутов
ipcrm	Удаление средства IPC
ipcs	Вывод информации о средствах IPC
kill	Отправка процессу сигнала прекращения выполнения
killall	Отправка всем процессам с указанным именем сигнала прекращения выполнения
lpr	Система печати
ls	Вывод содержимого каталога
lsb_release	Вывод информации о дистрибутиве
mknod	Создание файла специального типа
mktemp	Генерация уникального имени файла
more	Постраничный вывод содержимого файла
mount	Монтирование ФС
msgfmt	Создание объектного файла сообщений из файла сообщений
newgrp	Смена идентификатора группы
nice	Изменение приоритета процесса перед его запуском
nohup	Работа процесса после выхода из системы

Окончание таблицы 11

Команда	Назначение
od	Вывод содержимого файла в восьмеричном и других видах
passwd	Смена пароля учетной записи
patch	Применение файла описания изменений к оригинальному файлу
pidof	Вывод идентификатора процесса по его имени
ps	Вывод информации о процессах
renice	Изменение уровня приоритета процесса
sed	Строковый редактор
sendmail	Транспорт системы электронных сообщений
sh	Командный интерпретатор
shutdown	Команда останова системы
su	Изменение идентификатора запускаемого процесса
sync	Сброс системных буферов на носители
tar	Файловый архиватор
umount	Размонтирование ФС
useradd	Создание новой учетной записи пользователя или обновление существующей
userdel	Удаление учетной записи пользователя и соответствующих файлов окружения
usermod	Модификация информации об учетной записи пользователя
w	Список пользователей, работающих в настоящий момент в системе, и ресурсов, с которым осуществляется работа
who	Вывод списка пользователей системы

Описание команд приведено на страницах руководства man.

4.3.1. Планирование запуска команд

4.3.1.1. at

Для запуска одной или более команд в заранее определенное время используется команда at. В ней можно определить время и/или дату запуска той или иной команды. Команда at требует двух (или большего числа) параметров. Как минимум, следует указать время запуска и какая команда должна быть запущена.

Команды для запуска с помощью команды at вводятся как список в строках, следующих за ней. Ввод каждой строки завершается нажатием клавиши **<Enter>**. По окончании ввода всей команды нажать клавиши **<Ctrl+D>** для ее завершения.

Примеры:

1. Запустить команды `lpr /usr/sales/reports/.` и `echo "Files printed"` в 8:00

```
at 8:00
lpr /usr/sales/reports/.
echo "Files printed"
```

После ввода всей команды отобразится следующая запись:

```
job 756603300.a at Tue Jul 8 08:00:00 2014
```

означающая, что указанные команды будут запущены в 8:00, идентификатор задания `756603300.a` (может понадобится, если необходимо отменить задание командой `at -d`)

В результате выполнения команды в 8:00 будут распечатаны все файлы каталога `/usr/sales/reports`, и пользователю будет выведено сообщение на экран монитора.

2. Для запуска всех команд, перечисленных в файле `getdone`, в 17:30 следует воспользоваться одной из двух форм команды `at`:

```
at 17:30 < getdone
```

или

```
at 10:30 -f getdone
```

Обе приведенные команды эквивалентны. Разница заключается в том, что в первой команде используется механизм перенаправления потоков ввода-вывода, во второй команде — дисковый файл.

Кроме времени в команде `at` может быть определена дата.

Пример

```
at 10:00 Jul 14
lp /usr/sales/reports/
echo "Files printed"
```

Задания, определяемые администратором системы, помещаются в очередь, которую ОС периодически просматривает. Администратору необязательно находиться в системе для того, чтобы `at` отработала задания. В данном случае команда работает в фоновом режиме.

Для просмотра очереди заданий ввести:

```
at -l
```

Если предыдущие примеры были запущены, то будет выведено:

```
job 756603300.a at Sat Jul 8 08:00:00 2014 job 756604200.a at Sat Jul 14
  17:00:00 2014
```

Администратор системы видит только свои задания по команде `at`.

Для удаления задания из очереди следует запустить `at` с параметром `-d` и номером удаляемого задания:

```
at -d 756604200.a
```

В таблице 12 показаны варианты использования команды `at`.

Таблица 12

Формат команды	Назначение
`at hh:mm`	Выполнить задание во время `hh:mm` в 24-часовом формате
`at hh:mm месяц день год`	Выполнить задание во время `hh:mm` в 24-часовом формате в соответствующий день
`at -l`	Вывести список заданий в очереди; псевдоним команды — `atq`
`at now+count time-units`	Выполнить задание через определенное время, которое задано параметром `count` в соответствующих единицах — неделях, днях, часах или минутах
`at -d job_ID`	Удалить задание с идентификатором `job_ID` из очереди; псевдоним команды — `atrm`

Администратор системы может применять все эти команды. Для других пользователей права доступа к команде `at` определяются файлами `/etc/at.allow` и `/etc/at.deny`. Если существует файл `/etc/at.allow`, то применять команду `at` могут только перечисленные в нем пользователи. Если же такого файла нет, проверяется наличие файла `/etc/at.deny`, в котором отражено, кому запрещено пользоваться командой `at`. Если ни одного файла нет, значит, команда `at` доступна только суперпользователю.

Подробное описание команды приведено в `man at`.

4.3.1.2. cron

Для регулярного запуска команд в ОС существует команда `cron`. Администратор системы определяет для каждой программы время и дату запуска в минутах, часах, днях месяца, месяцах года и днях недели.

Команда `cron` запускается один раз при загрузке системы. Отдельные пользователи не должны иметь к ней непосредственного доступа. Кроме того, запуск `cron` никогда не осуществляется вручную путем ввода имени программы в командной строке, а только из сценария загрузки ОС.

При запуске `cron` проверяет очередь заданий команды `at` и задания пользователей в файлах `crontab`. Если команд для запуска нет, `cron` «засыпает» на одну минуту и затем вновь приступает к поискам команды, которую следует запустить в этот момент. Большую часть времени команда `cron` проводит в «спящем» состоянии, и для ее работы используется минимум системных ресурсов.

Чтобы определить список заданий для `cron` используется команда `crontab`. Для каждого пользователя с помощью данной команды создается файл `crontab` со списком заданий, находящийся в каталоге `/var/spool/cron/crontabs` и имеющий то же имя, что и имя пользователя.

Примечание. Пользователи, которым разрешено устанавливать задания командой `cron`, перечислены в файле `/etc/cron.allow`. Файл заданий для команды `cron` можно создать с помощью обычного текстового редактора, но при этом нельзя просто заменить им существующий файл задания в каталоге `/var/spool/cron/crontabs`. Для передачи `cron` сведений о новых заданиях обязательно должна использоваться команда `crontab`.

Каждая строка в файле `crontab` содержит шаблон времени и команду. Можно создать любое количество команд для `cron`. Команда выполняется тогда, когда текущее время соответствует приведенному шаблону. Шаблон состоит из пяти частей, разделенных пробелами или символами табуляции.

Синтаксис команд в файле `crontab`:

```
<минуты> <часы> <день_месяца> <месяц> <день_недели> <задание>
```

Первые пять полей представляют шаблон времени и должны присутствовать в файле. Для того чтобы `cron` игнорировала то или иное поле шаблона времени, следует поставить в поле символ * (звездочка).

Примечание. Символ * означает соответствие любому корректному значение.

Пример

Шаблон:

```
02 00 01 * *
```

определяет, что команда должна быть запущена в 00 часов 2 минуты каждого первого числа любого месяца (символ * в четвертом поле) независимо от дня недели (символ * в пятом поле).

В таблице 13 приведены допустимые значения полей записей crontab.

Таблица 13

Поле	Диапазон
<минуты>	00–59
<часы>	00–23 (полночь — 00)
<день_месяца>	01–31
<месяц>	01–12
<день_недели>	01–07 (понедельник — 01, воскресенье — 07)

Пример

Запись команды в файле crontab, выполняющая сортировку и отправку пользователю pav файла /usr/sales/weekly каждый понедельник в 7:30

```
30 07 * * 01 sort /usr/sales/weekly | mail -s"Weekly Sales" pav
```

Поле команд может содержать все, что может быть в команде, вводимой в командной строке оболочки. В нужное время cron для выполнения команд запустит стандартную оболочку (bash) и передаст ей команду для выполнения.

Для того чтобы определить несколько значений в поле используется запятая в качестве разделяющего символа. Например, если программа chkquotes должна выполняться в 9, 11, 14 и 16 часов по понедельникам, вторникам и четвергам 10 марта и 10 сентября, то запись выглядит так:

```
. 09,11,14,16 10 03,09 01,02,04 chkquotes
```

Параметры команды crontab приведены в таблице 14.

Таблица 14

Параметр	Описание
`-e`	Позволяет редактировать компоненты файла (при этом вызывается редактор, определенный в переменной `EDITOR` оболочки)
`-r`	Удаляет текущий файл `crontab` из каталога
`-l`	Используется для вывода списка текущих заданий `cron`

Команда `crontab` работает с файлом согласно регистрационному имени.

За корректное использование команды `cron` ответственность несут как администратор системы, так и пользователи, например, использование программы не должно вызвать перегрузку системы.

Подробное описание команд и файла `crontab` приведено в `man cron`, `man crontab` и `man 5 crontab`.

4.3.2. Администрирование многопользовательской и многозадачной среды

4.3.2.1. who

Для получения списка пользователей, работающих в ОС, используется инструмент командной строки `who`. Результатом выполнения команды является список, содержащий идентификаторы активных пользователей, терминалы и время входа в систему.

Пример

Результат выполнения команды `who`:

```
root console May 19 07:00
```

Основные параметры команды `who`:

1) `-u` — вывести список пользователей с указанием времени бездействия (символ «`.`» (точка) означает, что пользователь активно работал в последнюю минуту, `old` — что последний раз нажатие клавиш было более суток назад);

2) `-H` — вывести подробную информацию о пользователях. При этом выводится строка заголовка таблицы пользователей, описание столбцов приведено в таблице 15.

Таблица 15

Поле	Описание
ИМЯ	Имя пользователя
ЛИНИЯ	Использованные линии и терминалы
ВРЕМЯ	Время, прошедшее после регистрации пользователя в системе
IDLE	Время, прошедшее со времени последней активной работы пользователя
PID	Идентификатор процесса входной оболочки пользователя
КОММЕНТАРИЙ	Комментарий

Пример

Выполнение команды who с параметрами -u и -H:

```
who -uH
```

Результат выполнения команды:

```
ИМЯ   ЛИНИЯ    ВРЕМЯ           IDLE   PID     КОММЕНТАРИЙ
root console Dec 12 08:00   .      10340
```

Подробное описание команды приведено в man who.

4.3.2.2. ps

Для получения информации о состоянии запущенных процессов используется команда ps. Команда выводит следующую информацию о процессах:

- выполненные процессы;
- процессы, вызвавшие проблемы в системе;
- как долго выполняется процесс;
- какие системные ресурсы затребовал процесс;
- идентификатор процесса (который будет необходим, например, для прекращения работы процесса с помощью команды kill) и т. д.

Данная информация полезна как для пользователя, так и для системного администратора. Запущенная без параметров командной строки ps выдает список процессов, порожденных администратором.

Наиболее распространенное применение команды ps — отслеживание работы фоновых и других процессов в системе. Поскольку в большинстве случаев фоновые процессы не взаимодействуют с экраном и с клавиатурой, команда ps остается основным средством наблюдения за ними.

В таблице 16 приведены четыре основных поля информации для каждого процесса, выводимые командой `ps`.

Таблица 16

Поле	Описание
`PID`	Идентификатор процесса
`TTY`	Терминал, с которого был запущен процесс
`TIME`	Время работы процесса
`CMD`	Имя выполненной команды

Подробное описание команды приведено в `man ps`.

4.3.2.3. nohup

Обычно дочерний процесс завершается после завершения родительского. Таким образом, если запущен фоновый процесс, он будет завершен при выходе из системы. Для того чтобы процесс продолжал выполняться после выхода из системы, применяется команда `nohup`, указанная в начале командной строки:

```
nohup sort sales.dat &
```

Команда `nohup` заставляет ОС игнорировать выход из нее и продолжать выполнение процесса в фоновом режиме, пока он не закончится. Таким образом, будет запущен процесс, который будет выполняться длительное время, не требуя контроля со стороны администратора системы.

Подробное описание команды приведено в `man nohup`.

4.3.2.4. nice

Команда `nice` позволяет предопределять приоритет выполнения процесса (фонового или переднего плана) во время его запуска.

При запуске все процессы имеют одинаковый приоритет и ОС равномерно распределяет между ними процессорное время. С помощью команды `nice` можно понизить приоритет выбранного процесса, предоставив другим процессам больше процессорного времени.

Приоритет выполнения процесса может изменяться от -20 (наивысший приоритет) до 19 (наименьший приоритет). По умолчанию приоритет каждого процесса равен 10.

Повышение приоритета процесса осуществляется от имени администратора.

Синтаксис команды:

```
nice -<число> <команда>
```

Параметр <число> определяет на какое значение должен быть изменен приоритет выбранного процесса. Чем больше значение параметра <число>, тем меньше будет приоритет выбранного процесса.

Пример

Для процесса сортировки, запущенного командой:

```
sort sales.dat > sales.srt &
```

необходимо повысить приоритет над процессом печати.

Для этого необходимо запустить процесс печати с уменьшенным приоритетом, выполнив команду:

```
nice -5 lp mail_list &
```

Или назначить процессу печати самый низкий приоритет, выполнив команду:

```
nice -10 lp mail_list &
```

Для назначения процессу максимального приоритета -20 необходимо от имени администратора выполнить команду:

```
nice --30 <команда> &
```

Подробное описание команды приведено в `man nice`.

4.3.2.5. renice

Команда `renice` позволяет изменить приоритет запущенного процесса. Повышение приоритета процесса осуществляется от имени администратора.

Синтаксис команды:

```
renice -<число> <PID>
```

где `PID` — идентификатор процесса.

Определить PID можно с помощью команды ps:

```
ps -e | grep <имя_процесса>
```

Команда grep отфильтрует записи по имени нужного процесса.

Возможно изменить приоритет всех процессов пользователя или группы пользователей, для этого в команде renice используется идентификатор пользователя или группы.

Пример

Для изменения приоритета процесса текущего пользователя (pav) необходимо:

1) отобразить идентификаторы всех процессов, запущенных текущим пользователем, выполнив команду:

```
ps -ef | grep $LOGNAME
```

Результат выполнения команды:

```
pav 11805 11804 0 Dec 22 ttysb 0:01 sort sales.dat > sales srt
pav 19955 19938 4 16:13:02 ttypo 0:00 grep pav
pav 19938
1 0 16:11:04 ttypo 0-00 bash
pav 19940 19938 42 16:13:02 ttypo 0:33 find . -name core -exec
     nn {};
```

2) уменьшить приоритет процесса find с идентификатором 19940, выполнив команду:

```
renice -5 19940
```

Подробное описание команды приведено в man renice.

4.3.2.6. kill

Команда kill отправляет сигнал указанному процессу или процессам. Каждый сигнал имеет номер и название. Для просмотра всех сигналов необходимо выполнить команду:

```
kill -l
```

Синтаксис команды:

```
kill [-<сигнал>] <PID_1> [<PID_2> [...]]
```

где <сигнал> — номер сигнала или его название. Если параметр не задан, то по умолчанию будет применен сигнал с номером 15 (SIGTERM) на завершение выполнения процесса;
<PID_n> — идентификатор процесса.

С помощью параметра <сигнал> можно, например, дать указание процессу перечитать конфигурационные файлы без прекращения работы.

Если процесс работает не в фоновом режиме, нажатие комбинации клавиш **<Ctrl+C>** должно прервать его выполнение. Фоновый процесс прервать возможно только с помощью команды kill, посылающей процессу сигнал завершения.

Примеры:

1. Завершить процесс с идентификатором 127:

```
kill -SIGTERM 127
```

или:

```
kill -15 127
```

2. Завершить процессы с идентификаторами 127 и 240:

```
kill 127 240
```

Для завершения процесса, только что запущенного в фоновом режиме, необходимо выполнить команду:

```
kill $!
```

Для завершения всех фоновых процессов необходимо выполнить команду:

```
kill 0
```

При успешном завершении процесса сообщение не выводится. Сообщение появится при попытке завершения процесса без наличия соответствующих прав доступа или при попытке завершить несуществующий процесс.

Завершение родительского процесса приводит к завершению дочерних (кроме запущенных с помощью nohup). Однако для полной уверенности в завершении всех процессов, связанных с данным, следует указывать их в команде kill.

Некоторые процессы могут игнорировать посылаемые им сигналы, включая сигнал 15 (SIGTERM). Сигнал с номером 9 (SIGKILL) не может быть проигнорирован процессом, и процесс будет принудительно завершен. Например, если процесс не завершился после выполнения команды:

```
kill <PID_процесса>
```

то необходимо выполнить команду:

```
kill -9 <PID_процесса>
```

После выполнения команды процесс завершится без возможности корректно закрыть файлы, что может привести к потере данных.

Преимущественное право контроля над процессом принадлежит владельцу. Права владельца могут отменяться только суперпользователем.

Ядро назначает каждому процессу четыре идентификатора: реальный и эффективный UID, реальный и эффективный GID. Реальные ID используются для учета использования системных ресурсов, а эффективные — для определения прав доступа. Как правило, реальные и эффективные ID совпадают. Владелец процесса может посылать в процесс сигналы, а также понижать приоритет процесса.

Процесс, приступающий к выполнению другого программного файла, осуществляет один из системных вызовов семейства exec. Когда такое случается, эффективные UID и GID процесса могут быть установлены равными UID и GID файла, содержащего образ новой программы, если у этого файла установлены биты смены идентификатора пользователя и идентификатора группы. Системный вызов exec — это механизм, с помощью которого такие команды, как passwd, временно получают права суперпользователя (команде passwd они нужны для того, чтобы изменить /etc/passwd).

Подробное описание команды приведено в man kill.

5. УПРАВЛЕНИЕ ПРОГРАММНЫМИ ПАКЕТАМИ

В ОС используются программные пакеты (далее по тексту — пакеты) в формате DEB (файлы с расширением `.deb`). Для управления пакетами в режиме командной строки (или в эмуляторе терминала в графическом режиме) предназначены набор команд нижнего уровня `dpkg` и комплекс программ высокого уровня `apt`, `apt-cache` и `aptitude`. В графическом режиме управлять пакетами можно с помощью программы `synaptic` (универсальная графическая оболочка для `apt`).

По умолчанию обычный пользователь не имеет права использовать эти инструменты. Для всех операций с пакетами (за исключением некоторых случаев получения информации о пакетах) необходимы права суперпользователя, которые администратор может получить через механизм `sudo`.

Примечание. Права доступа к исполняемым файлам позволяют всем пользователям запускать их на выполнение, но удалять или модифицировать такие файлы может только суперпользователь. Обычно приложения устанавливаются в каталог с правами чтения всеми пользователями, но без права записи в него.

Средства управления пакетами обеспечивают возможность автоматизированной установки обновлений ОС.

5.1. dpkg

Инструмент командной строки `dpkg` предназначен для операций с пакетами на локальном уровне. С помощью `dpkg` можно устанавливать и удалять пакеты, собирать пакеты из исходных текстов, получать информацию о конкретном пакете и об установленных в системе пакетах.

Для установки пакета необходимо выполнить команду:

```
dpkg -i <полное_имя_пакета>
```

Пример

Для установки пакета `iptables_1.4.21-2_amd64.deb`, расположенного в домашнем каталоге пользователя `/home/user1`, выполнить следующую команду:

```
dpkg -i /home/user1/iptables_1.4.21-2_amd64.deb
```

В случае нарушения зависимостей будет выведено сообщение об ошибке, в котором будут перечислены все необходимые пакеты, которые следует установить для разрешения обязательных зависимостей.

Для удаления пакета с сохранением его конфигурационных, пользовательских и других файлов (в случае, если данный пакет не связан зависимостями с другими установленными пакетами) следует выполнить команду:

```
dpkg -r <имя_пакета>
```

Пример

Для удаления пакета `iptables_1.4.21-2_amd64.deb` необходимо выполнить команду:

```
dpkg -r iptables
```

Для удаления пакета и его конфигурационных, пользовательских и других файлов (в случае, если данный пакет не связан зависимостями с другими установленными пакетами) следует выполнить команду:

```
dpkg -P <имя_пакета>
```

Пример

Для удаления пакета `iptables_1.4.21-2_amd64.deb` необходимо выполнить команду:

```
dpkg -P iptables
```

При удалении пакета с зависимостями с другими пакетами будет отображено сообщение об ошибке с перечнем зависимостей.

Подробное описание команды приведено в `man dpkg`.

5.2. apt

Инструмент командной строки `apt` предназначен для выполнения операций с пакетами (при наличии доступа к сетевым или локальным репозиториям): устанавливать, удалять, обновлять, разрешать зависимости. А также искать пакеты по заданным критериям и просматривать подробную информацию о пакете.

5.2.1. Настройка доступа к репозиториям

Информация о сетевых и локальных репозиториях содержится в файле `/etc/apt/sources.list`. В файле указывается список источников пакетов, который используется программами для определения местоположения репозиториев. Список

источников разрабатывается для поддержки любого количества активных источников и различных видов этих источников. Источники перечисляются по одному в строке в порядке убывания их приоритета.

Описание файла `/etc/apt/sources.list` приведено в `man sources.list`.

Пример

Файл `sources.list`

```
# deb cdrom:[OS Astra Linux 1.8_x86-64 DVD ]/ 1.8_x86-64 contrib main
     non-free
deb https://dl.astralinux.ru/astra/stable/1.8_x86-64/repository-main/
     1.8_x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/stable/1.8_x86-64/repository-update/
     1.8_x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/stable/1.8_x86-64/uu/last/
     repository-update/ 1.8_x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/stable/1.8_x86-64/repository-base/
     1.8_x86-64 main contrib non-free
```

При установке ОС с DVD-диска строка `deb cdrom...` автоматически записывается в файл `sources.list`. Добавить данную строку в список источников также можно при помощи команды:

```
apt-cdrom add
```

при этом DVD-диск с дистрибутивом ОС должен находиться в устройстве чтения CD/DVD-дисков (монтировать его не обязательно).

Строки, соответствующие источникам остальных типов, добавляются в файл при помощи любого редактора.

5.2.2. Установка и удаление пакетов

После установки ОС создается локальная БД с информацией обо всех пакетах, которые находились на DVD-диске с дистрибутивом, и репозиторий установленных пакетов. Информацию о каждом установленном пакете можно просмотреть.

Пример

Для просмотра информации о пакете `iptables` выполнить команду:

```
apt show iptables
```

Обновить содержимое локальной БД можно при помощи команды:

```
apt update
```

Данную команду необходимо выполнять при каждом изменении списка источников пакетов или при изменении содержимого этих источников.

Полное обновление всех установленных в системе пакетов производится при помощи команды:

```
apt dist-upgrade
```

Установка отдельного пакета (если он отсутствовал в системе) выполняется командой:

```
apt install <имя_пакета>
```

При этом будут проверены и разрешены все обязательные зависимости и, при необходимости, установлены необходимые дополнительные пакеты.

Удаление пакета (с сохранением его конфигурационных файлов) выполняется командой:

```
apt remove <имя_пакета>
```

Для удаления пакета вместе с его конфигурационными файлами (кроме конфигурационных файлов из домашних каталогов пользователей) применяется команда:

```
apt remove --purge <имя_пакета>
```

Полное описание инструмента `apt` приведено в `man apt`.

6. БАЗОВЫЕ СЕТЕВЫЕ СЛУЖБЫ

6.1. Протокол TCP/IP

6.1.1. Пакеты и сегментация

Данные передаются по сети в форме сетевых пакетов, каждый из которых состоит из заголовка и полезной нагрузки. Заголовок содержит сведения о том, откуда прибыл пакет и куда он направляется. Заголовок, кроме того, может включать контрольную сумму, информацию, характерную для конкретного протокола, и другие инструкции по обработке. Полезная нагрузка — это данные, подлежащие пересылке.

6.1.2. Адресация пакетов

Сетевые пакеты могут достичь пункта назначения только при наличии правильного сетевого адреса. Протокол TCP/IP использует сочетание нескольких схем сетевой адресации.

Самый нижний уровень адресации задается сетевыми аппаратными средствами.

На следующем, более высоком, уровне используется адресация Интернет (которую чаще называют «IP-адресацией»). Каждому включенному в сеть устройству присваивается один четырехбайтовый IP-адрес (в соответствии с протоколом IPv4). IP-адреса глобально уникальны и не зависят от аппаратных средств.

IP-адреса идентифицируют компьютер, но не обеспечивают адресацию отдельных процессов и служб. Протоколы TCP и UDP расширяют IP-адреса, используя порты. Порт в данном случае представляет собой двухбайтовое число, добавляемое к IP-адресу и указывающее конкретного адресата той или иной сетевой службы. Все стандартные UNIX-службы связываются с известными портами, которые определены в файле `/etc/services`. Для того чтобы предотвратить попытки нежелательных процессов замаскироваться под эти службы, установлено, что порты с номерами до 1024 могут использоваться только суперпользователем. Описание файла `/etc/services` приведено в `man services`.

6.1.3. Маршрутизация

6.1.3.1. Таблица

Маршрутизация — это процесс направления пакета по ряду сетей, находящихся между источником и адресатом.

Данные маршрутизации хранятся в таблице маршрутизации. Каждый элемент этой таблицы содержит несколько параметров, включая поле метрики, в котором указано значение приоритета маршрута на определенном сетевом интерфейсе (если таблица содержит противоречивую информацию). Для направления пакета по конкретному адресу подбирается

наиболее подходящий маршрут. Если нет такого маршрута и нет маршрута по умолчанию, то отправителю возвращается ошибка «network unreachable» (сеть недоступна).

Таблицу маршрутизации компьютера можно вывести на экран монитора с помощью команды `route`.

6.1.3.2. Организация подсетей

Организация подсетей задается маской подсети, в которой биты сети включены, а биты компьютера выключены. Маска подсети задается во время начальной загрузки, когда конфигурируется сетевой интерфейс командой `ifconfig`. Ядро, как правило, использует сам класс IP-адресов для того, чтобы выяснить, какие биты относятся к сетевой части адреса; если задать маску явно, то эта функция просто отменяется.

При организации подсетей необходимо учесть, что если вычислительная сеть имеет более одного соединения с сетью Интернет, то другие сети должны уметь отличать подсети сети пользователя, чтобы определить в какой маршрутизатор следует послать пакет.

6.1.4. Создание сети TCP/IP

Процесс создания сети TCP/IP состоит из следующих этапов:

- планирование сети;
- назначение IP-адресов;
- настройка сетевых интерфейсов;
- настройка статических маршрутов.

6.1.4.1. Планирование сети

Планирование сети включает:

- определение сегментов сети;
- определение технических и программных средств, с помощью которых сегменты объединяются в сеть;
- определение серверов и рабочих станций, которые будут установлены в каждом сегменте;
- определение типа среды (витая пара и др.).

6.1.4.2. Назначение IP-адресов

Адреса назначают сетевым интерфейсам, а не компьютерам. Если у компьютера есть несколько физических интерфейсов, то у него будет несколько сетевых адресов.

Существует возможность создания виртуального сетевого интерфейса (loopback), который реализован программно и не связан с оборудованием, но при этом полностью интегрирован во внутреннюю сетевую инфраструктуру компьютерной системы.

Для того, чтобы можно было обращаться к компьютерам по их именам рекомендуется использовать службу DNS, также соответствие между именем и IP-адресом может быть задано в файле `/etc/hosts` на компьютере, с которого выполняется обращение.

6.1.4.3. Настройка сетевых интерфейсов

Инструмент `ifconfig` используется для включения и выключения сетевого интерфейса, задания IP-адреса, широковещательного адреса и связанной с ним маски подсети, а также для установки других параметров. Он обычно используется при первоначальной настройки, но может применяться и для внесения изменений в дальнейшем.

В большинстве случаев команда имеет следующий формат:

```
ifconfig интерфейс [семейство] <адрес> up <параметр> ...
```

Пример

```
ifconfig eth0 128.138.240.1 up netmask 255.255.255.0 \
     broadcast 128.138.240.255
```

Здесь интерфейс обозначает аппаратный интерфейс, к которому применяется команда. Как правило, это двух-трехсимвольное имя устройства, за которым следует число. Примеры распространенных имен `eth1`, `lo0`, `ppp0` образуются из имени драйвера устройства, используемого для управления им. Для того чтобы выяснить, какие интерфейсы имеются в системе, можно воспользоваться командой:

```
netstat-i
```

Параметр `up` включает интерфейс, а параметр `down` выключает его.

Описание инструмента приведено в `man ifconfig`.

6.1.4.4. Настройка статических маршрутов

Инструмент `route` определяет статические маршруты — явно заданные элементы таблицы маршрутизации, которые обычно не меняются даже в тех случаях, когда запускается серверный процесс маршрутизации.

Маршрутизация выполняется на уровне IP. Когда поступает пакет, предназначенный для другого компьютера, IP-адрес пункта назначения пакета сравнивается с маршрутами, ука-

занными в таблице маршрутизации ядра. Если номер сети пункта назначения совпадает с номером сети какого-либо маршрута, то пакет направляется по IP-адресу следующего шлюза, связанного с данным маршрутом.

Существующие маршруты можно вывести на экран.

Описание инструмента приведено в `man route`.

6.1.5. Проверка и отладка сети

6.1.5.1. ping

Инструмент `ping` служит для проверки соединений в сетях на основе TCP/IP.

После команды запуска он работает в бесконечном цикле, если не задан параметр `-c`, определяющий количество пакетов, после передачи которого команда завершает свое выполнение. Чтобы прекратить работу `ping`, необходимо нажать **<Ctrl+C>**.

Описание инструмента приведено в `man ping`.

6.1.5.2. netstat

Инструмент `netstat` выдает информацию о состоянии, относящуюся к сетям:

- проверка состояния сетевых соединений;
- анализ информации о конфигурации интерфейсов;
- изучение таблицы маршрутизации;
- получение статистических данных о различных сетевых протоколах.

Команда запуска `netstat` без параметров выдает информацию о состоянии активных портов TCP и UDP. Неактивные серверы, ожидающие установления соединения, как правило, не показываются (их можно просмотреть командой `netstat -a`).

Основные параметры `netstat`:

1) `-i` — показывает состояние сетевых интерфейсов;
2) `-r` — выдает таблицу маршрутизации ядра;
3) `-s` — выдает содержимое счетчиков, разбросанных по сетевым программам.

Описание инструмента приведено в `man netstat`.

6.1.5.3. arp

Инструмент `arp` обращается к таблице ядра, в которой задано соответствие IP-адресов аппаратным адресам. В среде Ethernet таблицы ведутся с помощью протокола ARP и не требуют администрирования.

Команда:

```
arp -a
```

выводит содержимое таблицы соответствий.

Описание инструмента приведено в `man arp`.

6.2. Протокол FTP

В ОС передача файлов обеспечивается с помощью интерактивного инструмента `lftp`, запускаемого на клиентской стороне, и серверной службы `vsftpd`, которая запускается на компьютере, выполняющем функцию сервера FTP. Инструмент и служба реализуют протокол передачи файлов FTP. Для копирования файлов клиенту обычно необходимо знание имени и пароля пользователя (хотя существует и вариант анонимного доступа), которому принадлежат файлы на сервере FTP.

6.2.1. Клиентская часть

Клиентская часть может быть установлена командой:

```
apt install lftp
```

Запуск инструмента `lftp` осуществляется командой:

```
lftp <имя_сервера>
```

Интерактивный доступ к серверу службы FTP обеспечивается следующими основными внутренними командами `lftp`:

1) `open`, `user`, `close` — связь с удаленным компьютером;
2) `lcd`, `dir`, `mkdir`, `lpwd` — работа с каталогами в FTP-сервере;
3) `get`, `put`, `ftpcopy` — получение и передача файлов;
4) `ascii`, `binary`, `status` — установка параметров передачи.

Выход из инструмента `lftp` осуществляется по команде `exit`.

Описание инструмента приведено в `man lftp`.

6.2.2. Служба vsftpd сервера FTP

В ОС служба `vsftpd` устанавливается командой:

```
apt install vsftpd
```

После установки службы `vsftpd` по умолчанию в конфигурационном файле `/etc/vsftpd.conf` указаны параметры для работы с включенной IPv4- и IPv6-адресацией — для параметров `listen` и `listen_ipv6` установлены следующие значения:

```
listen=NO
listen_ipv6=YES
```

Для приема соединения как от клиентов IPv4, так и от клиентов IPv6 достаточно, чтобы для параметра `listen_ipv6` было установлено значение `YES`, при этом значение параметра `listen` всегда будет интерпретироваться как `YES`.

Если в системе включена IPv6-адресация, то служба `vsftpd` запускается автоматически без дополнительных настроек.

Если в системе отключена IPv6-адресация или необходимо использовать только IPv4-адресацию, то для запуска службы `vsftpd` требуется отредактировать файл `/etc/vsftpd.conf` — для параметров `listen` и `listen_ipv6` должны быть установлены следующие значения:

```
listen=YES
listen_ipv6=NO
```

Для настройки `vsftpd` не требуется указывать все доступные параметры, а достаточно указать только те, значения которых следует переопределить. Параметры, не указанные явным образом в файле `/etc/vsftpd.conf`, будут принимать значения по умолчанию. Значения, принимаемые по умолчанию, приведены в `man vsftpd.conf`.

В конфигурационном файле `/etc/vsftpd.conf` присутствуют параметры, которые зависят от других параметров. Если один параметр, от которого зависит другой, отключен, то и зависимый параметр также будет отключен. Например, если параметр `local_enable`, позволяющий авторизоваться локальным пользователям, будет отключен, то зависящий от него параметр `local_umask` также будет отключен.

Для запуска службы `vsftpd` с параметрами, отличными от указанных в файле `/etc/vsftpd.conf`, необходимо использовать инструмент командной строки `vsftpd`. Таким образом, если значение параметра, переданное в командной строке, не совпадает с

указанным в конфигурационном файле, то будет применено значение параметра, указанное в командной строке, так как оно имеет приоритет над указанным в конфигурационном файле. Значения параметров, указанных в командной строке, применяются последовательно.

После установки службы `vsftpd` создается каталог с документацией службы `/usr/share/doc/vsftpd`, где каталог `examples` содержит примеры конфигурационного файла `vsftpd.conf`.

Описание службы `vsftpd` и файла `/etc/vsftpd.conf` приведено в `man vsftpd` и `man vsftpd.conf` соответственно.

6.3. Протокол DHCP

На компьютере, выполняющем роль сервера динамической конфигурации сети, должна быть установлена служба DHCP-сервера.

DHCP-сервер представлен пакетом `isc-dhcp-server` и графической утилитой `fly-admin-dhcp` для его быстрой настройки.

Для установки DHCP-сервера выполнить команду от имени администратора с использованием механизма `sudo`:

```
sudo apt install isc-dhcp-server
```

Установка графической утилиты `fly-admin-dhcp` выполняется командной:

```
sudo apt install fly-admin-dhcp
```

При установке `fly-admin-dhcp` также автоматически будет установлен пакет `isc-dhcp-server`.

Запуск службы DHCP-сервера осуществляется с помощью команды:

```
systemctl start isc-dhcp-server
```

или автоматически путем включения в список служб, запускаемых при старте системы.

Настройки службы DHCP-сервера задаются в файлах `/etc/default/isc-dhcp-server` и `/etc/dhcp/dhcpd.conf`.

В файле /etc/default/isc-dhcp-server для параметров INTERFACES указываются протоколы и сетевые интерфейсы, с которыми будет работать служба, например:

```
INTERFACESv4="eth0"
#INTERFACESv6=" "
```

При необходимости возможно указать несколько сетевых интерфейсов, разделенных пробелом.

В файле /etc/dhcp/dhcpd.conf указывается топология сети и параметры выдаваемой через DHCP-сервер информации.

ВНИМАНИЕ! Для запуска службы DHCP-сервера указанному в файле /etc/default/isc-dhcp-server сетевому интерфейсу должен быть присвоен IP-адрес и данный IP-адрес должен быть назначен вручную в файле /etc/dhcp/dhcpd.conf.

Сервер динамически назначает IP-адреса DHCP-клиентам обеих подсетей и осуществляет поддержку нескольких клиентов BOOTP. Первые несколько активных строк файла определяют ряд параметров и режимов, действующих для всех обслуживаемых сервером подсетей и клиентов. Каждая строка задана шаблоном «параметр — значение». Параметр может быть общим или стоять перед option. Параметр, следующий за option, — это ключ настройки, он состоит из имени ключа и его значения.

Кроме общих параметров, существуют т. н. «операторы топологии сети» или «объявления».

Описание некоторых параметров настройки DHCP-сервера, содержащихся в файле /etc/dhcp/dhcpd.conf, приведено в таблице 17.

Таблица 17

Параметр	Описание
max-lease-time	Определяет максимально допустимое время аренды. Независимо от длительности аренды, фигурирующей в запросе клиента, этот срок не может превышать значение, заданное данным параметром
get-lease-hostnames	Предписывает dhcpd предоставлять каждому клиенту наряду с динамическим адресом имя узла. Имя узла должно быть получено от DNS. Данный параметр — логический. При значении FALSE назначается адрес, но не имя узла. Значение TRUE используется только в сетях с небольшим количеством хостов, которым выделяются имена, т. к. поиск имен в DNS замедляет запуск демона
hardware type address	Параметр определяет аппаратный адрес клиента. Значение type может быть ethernet или token-ring. address должен быть соответствующим устройству физическим адресом. Параметр должен быть связан с оператором host. Он необходим для распознавания клиента BOOTP

Продолжение таблицы 17

Параметр	Описание
`filename file`	Указывает файл загрузки для бездисковых клиентов. `file` — это ASCII-строка, заключенная в кавычки
`range [dynamic-bootp]`	Данный параметр указывает диапазон адресов. После него через пробел указывается нижний адрес диапазона и опционально верхний адрес. Если верхний адрес не указан, занимается весь теоретически возможный диапазон от нижнего адреса. Этот параметр всегда связан с оператором `subnet`. Все адреса должны принадлежать этой подсети. Флаг `dynamic-bootp` указывает, что адреса могут автоматически назначаться клиентам BOOTP также, как и клиентам DHCP. Если оператор `subnet` не содержит параметра `range`, для такой подсети динамическое распределение адресов не действует
`server-name name`	Имя сервера DHCP, передаваемое клиенту. `name` — это ASCII-строка, заключенная в кавычки
`next-server name`	Имя узла или адрес сервера, с которого следует получить загрузочный файл
`fixed-address`	Назначает узлу один или несколько фиксированных адресов. Действителен только в сочетании с параметром `host`. Если указано несколько адресов, выбирается адрес, корректный для данной сети, из которой выполняет загрузку клиент. Если такого адреса нет, никакие параметры не передаются
`dynamic-bootp-lease-cuttof date`	Устанавливает дату завершения действия адресов, назначенных клиентам BOOTP. Клиенты BOOTP не обладают способностью обновлять аренду и не знают, что срок аренды может истечь. Этот параметр меняет поведение сервера и используется только в особых случаях
`dynamic-bootp-lease-length`	Длительность аренды в секундах для адресов, автоматически назначаемых клиентам BOOTP. Данный параметр используется в особых ситуациях, когда клиенты используют образ загрузки BOOTP PROM. В ходе загрузки клиент действует в качестве клиента BOOTP, а после загрузки работает с протоколом DHCP и умеет обновлять аренду
`use-host-decl-names`	Предписывает передавать имя узла, указанное в операторе `host`, клиенту в качестве его имени. Логический параметр, может иметь значения `TRUE` или `FALSE`
`server-identifier hostname`	Определяет значение, передаваемое в качестве идентификатора сервера. По умолчанию передается первый IP-адрес сетевого интерфейса
`authoritative not authoritative`	Указывает, является ли сервер DHCP компетентным. `not authoritative` используется, когда в компетенцию сервера не входит распределение адресов клиентам
`use-lease-addr-for-default-route`	Логический параметр (`TRUE` или `FALSE`). Предписывает передавать клиенту арендованный адрес в качестве маршрута по умолчанию. Параметр используется только тогда, когда локальный маршрутизатор является сервером-посредником ARP. Оператор настройки `routers` имеет более высокий приоритет

Окончание таблицы 17

Параметр	Описание
always-replay-rfc1048	Логический параметр. Предписывает посылать клиенту BOOTP ответы в соответствии с RFC 1048
allow keyword deny keyword	Определяет необходимость отвечать на запросы различных типов. Ключевое слово keyword указывает тип разрешенных и запрещенных запросов. Существуют следующие ключевые слова: 1) unknown-clients — определяет возможность динамического назначения адресов неизвестным клиентам; 2) bootp — определяет необходимость отвечать на запросы BOOTP (по умолчанию обслуживаются); 3) booting — используется внутри объявления host для указания необходимости отвечать тому или иному клиенту. По умолчанию сервер отвечает всем клиентам

Каждый из операторов топологии может многократно встречаться в файле настройки. Операторы определяют иерархическую структуру. Операторы топологии, встречающиеся в файле /etc/dhcp/dhcpd.conf, приведены в таблице 18.

Таблица 18

Оператор	Описание
group {[parameters] [options]}	Группирует операторы shared-network, subnet, host и другие операторы group. Позволяет применять наборы параметров ко всем элементам группы
shared-network name {[parameters] [options]}	Используется только в случае, когда несколько подсетей находятся в одном физическом сегменте. В большинстве случаев различные подсети находятся в различных физических сетях. В качестве имени name может использоваться любое описательное имя. Оно используется только в отладочных сообщениях. Параметры, связанные с общей сетью, объявляются внутри фигурных скобок и действуют на все подсети общей сети. Каждый оператор shared-network содержит не менее двух операторов subnet, в противном случае нет необходимости использовать группирование

Общеупотребительные параметры, следующие за ключевым словом option в файле /etc/dhcp/dhcpd.conf, приведены в таблице 19.

Таблица 19

Параметр	Описание
subnet-mask	Определяет маску подсети в формате десятичной записи через точку. Если subnet-mask отсутствует, dhcpd использует маску подсети из оператора subnet
time-offset	Указывает разницу данного часового пояса с временем UTC в секундах

Окончание таблицы 19

Параметр	Описание
`routers`	Перечисляет адреса доступных клиентам маршрутизаторов в порядке предпочтения
`domain-name-servers`	Перечисляет адреса доступных клиентам серверов DNS в порядке предпочтения
`lpr-servers`	Перечисляет адреса доступных клиентам серверов печати LPR в порядке предпочтения
`host-name`	Указывает имя узла для клиента
`domain-name`	Определяет имя домена
`interface-mtu`	Определяет значение MTU для клиента в байтах. Минимально допустимое значение — 68
`broadcast-address`	Определяет широковещательный адрес для подсети клиента
`static-routes destination gateway`	Перечисляет доступные клиенту статические маршруты. Маршрут по умолчанию не может быть указан таким способом. Для его указания используется параметр `routers`
`trailer-encapsulation`	Определяет, следует ли клиенту выполнять инкапсуляцию завершителей (оптимизация, основанная на изменении порядка данных). Значение 0 означает, что инкапсуляцию выполнять не следует, 1 — выполнять
`nis-domain string`	Строка символов, определяющая имя домена NIS
`dhcp-client-identifier string`	Используется в операторе `host` для определения идентификатора клиента DHCP. `dhcpd` может использовать данное значение для идентификации клиента вместо аппаратного адреса

ВНИМАНИЕ! Для корректной работы DHCP-сервера требуется в файле `/etc/dhcp/dhcpd.conf` раскомментировать параметр `authoritative`.

После завершения настроек следует перезапустить службу DHCP-сервера с помощью команды:

```
sudo systemctl restart isc-dhcp-server
```

Описание службы DHCP-сервера и файла `/etc/dhcp/dhcpd.conf` приведено на страницах руководств `man dhcpd` и `man dhcpd.conf`.

6.4. Протокол NFS

Протокол NFS обеспечивает общий доступ к файлам и каталогам *nix-систем (в т. ч. Linux), что позволяет использовать ФС удаленных компьютеров.

В ОС используется реализация службы NFS, работающая на уровне ядра и представленная пакетом `nfs-kernel-server`.

Доступ к ФС удаленных компьютеров обеспечивается с помощью программ на сторонах сервера и клиента.

При работе с сетевой ФС любые операции над файлами, производимые на локальном компьютере, передаются через сеть на удаленный компьютер.

6.4.1. Установка и настройка сервера

Для установки сервера выполнить от имени администратора команды:

```
apt update
apt install nfs-kernel-server
```

Для нормального запуска и возобновления работы службы сервера NFS требуется после установки пакета и перезагрузки компьютера внести изменения в UNIT-файл `/etc/systemd/system/multi-user.target.wants/nfs-server.service`, добавив следующие строки в секцию `unit`:

```
[Unit]
Requires=rpcbind.service
After=rpcbind.service
```

Затем перезапустить службу, выполнив команды:

```
systemctl daemon-reload
systemctl restart nfs-kernel-server
```

На стороне сервера существуют следующие программы, используемые для обеспечения службы NFS:

- `rpc.idmapd` — перенаправляет обращения, сделанные с других компьютеров к службам NFS;
- `rpc.nfsd` — переводит запросы к службе NFS в действительные запросы к локальной ФС;
- `rpc.svcgssd` — поддерживает создание защищенного соединения;
- `rpc.statd` — поддерживает восстановление соединения при перезагрузке сервера;
- `rpc.mountd` — запрашивается для монтирования и размонтирования ФС.

Описание программ приведено на страницах руководства `man`.

Запросы монтирования поступают от клиентских компьютеров к серверу монтирования mountd, который проверяет правильность клиентского запроса на монтирование и разрешает серверу службы NFS (nfsd) обслуживать запросы клиента, выполнившего монтирование. Клиенту разрешается выполнять различные операции с экспортированной ФС в пределах своих полномочий. Для получения хорошего качества обслуживания клиентов рекомендуется на сервере службы NFS одновременно запускать несколько копий процесса nfsd.

На стороне сервера выполняется экспортирование ФС. Это означает, что определенные поддеревья, задаваемые каталогами, объявляются доступными для клиентских компьютеров. Информация об экспортированных ФС заносится в файл /etc/exports, в котором указывается, какие каталоги доступны для определенных клиентских компьютеров, а также какими правами доступа обладают клиентские компьютеры при выполнении операций на сервере. В конфигурационный файл /etc/exports информация заносится строкой вида:

```
<общий_каталог> <IP-адрес_клиента>(<параметр>)
```

Параметр определяет правила монтирования общего ресурса для клиента. Если параметров несколько, то они указываются через запятую. Перечень параметров и их описание приведены в таблице 20.

Таблица 20

Параметр	Описание
rw	Предоставляет права на чтение и запись
ro	Предоставляет права только на чтение
no_root_squash	По умолчанию в общих ресурсах NFS пользователь root становится обычным пользователем (nfsnobody). Таким образом, владельцем всех файлов, созданных root, становится nfsnobody, что предотвращает загрузку на сервер программ с установленным битом setuid. Если указан параметр no_root_squash, то удаленные пользователи root могут изменить любой файл в разделяемой файловой системе и внести вредоносный код для других пользователей. В целях безопасности рекомендуется этот параметр не использовать
nohide	Служба NFS автоматически не показывает нелокальные ресурсы (например, примонтированные с помощью mount --bind). Данный параметр включает отображение таких ресурсов
sync	Синхронный режим доступа. Указывает, что сервер должен отвечать на запросы только после записи на диск изменений, выполненных этими запросами
async	Асинхронный режим доступа. Указывает серверу не ждать записи информации на диск и давать ответ на запрос сразу. Использование этого режима повышает производительность, но понижает надежность, т.к. в случае обрыва соединения или отказа оборудования возможна потеря данных

Окончание таблицы 20

Параметр	Описание
noaccess	Запрещает доступ к указанному каталогу. Применяется, если доступ к определенному каталогу выдан всем пользователям сети, но при этом необходимо ограничить доступ для отдельных пользователей
all_squash	Подразумевает, что все подключения будут выполняться от анонимного пользователя
subtree_check	Выполняет контроль поддерева — позволяет экспортировать не весь раздел, а лишь его часть. При этом сервер NFS выполняет дополнительную проверку обращений клиентов для проверки, что они предпринимают попытку доступа к файлам, находящимся в соответствующих подкаталогах. Параметр subtree_check включен по умолчанию
no_subtree_check	Отменяет контроль поддерева. Не рекомендуется использовать данный параметр, т. к. может быть нарушена безопасность системы. Параметр может применяться в том случае, если экспортируемый каталог совпадает с разделом диска
anonuid=1000	Привязывает анонимного пользователя к локальному UID
anongid=1000	Привязывает анонимную группу пользователя к локальной группе GID

Пример

Описание в конфигурационном файле /etc/exports экспорта совместно используемого каталога /nfsshare

```
/srv/nfsshare 192.168.1.20/255.255.255.0(rw,nohide,all_squash,
     anonuid=1000,anongid=1000,no_subtree_check)
```

ВНИМАНИЕ! Использование пробелов между IP-адресом/именем клиента и правами его доступа в файле /etc/exports влечет изменение трактовки прав доступа. Например, строка:

```
/tmp/nfs/ master.astralinux.ru(rw)
```

предоставляет ресурсу master.astralinux.ru права на доступ и чтение, в то время как строка:

```
/tmp/nfs/ master.astralinux.ru (rw)
```

предоставляет ресурсу master.astralinux.ru права только на чтение, а всем остальным — на чтение и запись.

После внесения изменений в файл /etc/exports необходимо выполнить команду:

```
exportfs -ra
```

6.4.2. Установка и настройка клиента

Для установки клиента выполнить на компьютере от имени администратора команды:

```
apt update
apt install nfs-common
```

После установки пакета `nfs-common` на клиенте возможно примонтировать совместно используемые ресурсы. Список доступных ресурсов можно проверить, выполнив команду:

```
showmount -e <IP-адрес_сервера>
```

Для монтирования совместно используемого ресурса на клиенте выполнить команду:

```
mount <IP-адрес_сервера>:<общий_каталог> <каталог_монтирования>
```

где `<IP-адрес_сервера>` — имя сервера NFS;
`<общий_каталог>` — экспортированный каталог сервера NFS;
`<каталог_монтирования>` — каталог монтирования на клиенте.

На стороне клиента для поддержки службы NFS используется модифицированная команда `mount` (если указывается ФС NFS4, то автоматически вызывается команда `mount.nfs4`). Команда модифицирована таким образом, чтобы она могла понимать запись:

```
<IP-адрес_сервера>:<общий_каталог>
```

Для удаленных ФС, которые являются частью постоянной конфигурации клиента и должны автоматически монтироваться во время начальной загрузки клиента, должны присутствовать соответствующие строки в файле `/etc/fstab` клиента, например:

```
192.168.1.10:/srv/nfsshare/    /mnt/share nfs rw,sync,hard,intr 0 0
```

Кроме того, для поддержки защищенных соединений на клиентской стороне должна запускаться команда `rpc.gssd`.

6.5. DNS

Система доменных имен DNS (Domain Name System) представляет собой иерархическую распределенную систему для получения информации о компьютерах, службах и ресурсах, входящих в глобальную или приватную компьютерную сеть. Чаще всего используется

для получения IP-адреса по имени компьютера или устройства, получения информации о маршрутизации почты и т. п.

Основой DNS является представление об иерархической структуре доменного имени и зонах. Распределенная база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определенному протоколу. Каждый сервер, отвечающий за имя, может делегировать ответственность за дальнейшую часть домена другому серверу, что позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени.

Основными важными понятиями DNS являются:

- домен (область) — именованная ветвь или поддерево в дереве имен. Структура доменного имени отражает порядок следования узлов в иерархии; доменное имя читается справа налево от младших доменов к доменам высшего уровня (в порядке повышения значимости);
- полное имя домена (FQDN) — полностью определенное имя домена. Включает в себя имена всех родительских доменов иерархии DNS;
- зона — часть дерева доменных имен (включая ресурсные записи), размещаемая как единое целое на некотором сервере доменных имен;
- DNS-запрос — запрос от клиента (или сервера) серверу для получения информации.

Служба доменных имен `named` предназначена для генерации ответов на DNS-запросы. Существуют два типа DNS-запросов:

- прямой — запрос на преобразование имени компьютера в IP-адрес;
- обратный — запрос на преобразование IP-адреса в имя компьютера.

6.5.1. Установка DNS-сервера

В ОС используется DNS-сервер BIND9. Для установки службы DNS-сервера выполнить в терминале команду:

```
apt install bind9
```

При установке пакета `bind9` будет автоматически уставновлен пакет инструментов командной строки `bind9utils`, включающий:

- `named-checkconf` — инструмент проверки синтаксиса файлов конфигурации;
- `named-checkzone` — инструмент проверки файлов зон DNS;
- `rndc` — инструмент управления службой DNS.

Дополнительно также рекомендуется установить пакет инструментов командной строки для работы с DNS `dnsutils`, выполнив команду:

```
apt install dnsutils
```

В составе пакета `dnsutils` будут установлены следующие инструменты:

- `dig` — инструмент для опроса DNS-серверов и проверки их ответа;
- `nslookup` — инструмент для проверки преобразования имен в IP-адреса (разрешение имен);
- `nsupdate` — инструмент для динамического обновления записей DNS.

ВНИМАНИЕ! При установке службы DNS-сервера будут автоматически созданы учетная запись пользователя `bind` и группа `bind`. Соответственно, служба будет работать от имени `bind:bind`.

6.5.2. Настройка сервера службы доменных имен `named`

Конфигурационные параметры службы `named` хранятся в файлах каталога `/etc/bind/`, перечень конфигурационных файлов приведен в таблице 21.

Таблица 21 – Конфигурационные файлы службы доменных имен `named`

Файл	Описание
`/etc/bind/named.conf`	Основной конфигурационный файл. Содержит значения конфигурационных параметров для всего сервера и ссылки на другие конфигурационные файлы
`/etc/bind/named.conf.options`	Конфигурационный файл основных параметров сервера, основным из которых является параметр `directory`, содержащий каталог конфигурационных файлов зон. Значение по умолчанию `/var/cache/bind`
`/etc/bind/named.conf.local`	Конфигурационный файл описания локальных зон сервера. Для каждой зоны указываются пути к конфигурационным файлам для прямого и обратного разыменования (как правило, в указанном ранее каталоге `/var/cache/bind`)
`/etc/bind/named.conf.default-zones`	Конфигурационный файл зон по умолчанию. В частности, в этом файле содержатся ссылки на автоматически созданные файлы конфигурации `/etc/bind/db.local` и `/etc/bind/127.db` зоны `localhost`. В большинстве случаев не требует правки

Настройка сервера доменных имен является сложной задачей. Перед использованием DNS следует ознакомиться с существующей документацией, файлами помощи и страницами

руководства man службы named, конфигурационного файла named.conf и сопутствующих утилит.

Далее приведен типовой пример настройки службы доменных имен named, обслуживающей одну доменную зону. Пример достаточен для демонстрации функционирующего домена ЕПП ОС.

Примечание. Обновление конфигурации сервера может выполняться без перезапуска самой службы доменных имен named вызовом:

```
rndc reload
```

Пример

Настройка сервера DNS домена my.dom подсети 192.168.1.

В конфигурационный файл /etc/bind/named.conf.local необходимо добавить следующие строки:

```
zone "my.dom" {
        type master;
        file "/var/cache/bind/db.my.dom";
};

zone "1.168.192.in-addr.arpa" {
        type master;
        file "/var/cache/bind/db.192.168.1";
};
```

Примечание. Имена конфигурационных файлов следует выбирать так, чтобы было понятно для какой конфигурации они используются. В приведенном примере имя конфигурационного файла для зоны обратного просмотра может быть, например, /var/cache/bind/1.168.192.in-addr.arpa.zone или /var/cache/bind/db.my.dom.inv.

Конфигурационный файл /var/cache/bind/db.my.dom содержит информацию зоны прямого просмотра:

```
;
; BIND data file for my.dom zone
;
$TTL    604800
@       IN      SOA     my.dom. root.my.dom. (
                         2014031301     ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
```

```
                          604800 )       ; Negative Cache TTL
;
@       IN      NS      server.my.dom.
@       IN      A       192.168.1.100
@       IN      MX      1       server.my.dom.

server          IN      A       192.168.1.100
client1         IN      A       192.168.1.101
client2         IN      A       192.168.1.102
client3         IN      A       192.168.1.103

ns      IN      CNAME   server
;gw CNAMEs
ftp     IN      CNAME   server
repo    IN      CNAME   server
ntp     IN      CNAME   server

_https._tcp  IN SRV     10 10 443 server.my.com.

client1      IN TXT    "MAKS"
```

Конфигурационный файл `/var/cache/bind/db.192.168.1` содержит информацию зоны обратного просмотра:

```
;
; BIND reverse data file for my.dom zone
;
$TTL    86400
@       IN      SOA my.dom. root.my.dom. (
                         2014031301     ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                         2419200        ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      server.my.dom.

100     IN      PTR     server.my.dom.
101     IN      PTR     client1.my.dom.
102     IN      PTR     client2.my.dom.
103     IN      PTR     client3.my.dom.
```

Описание зон может содержать следующие основные типы записей:

- NS — имя DNS сервера;
- A — связь имени с IP-адресом;
- CNAME — связь псевдонима с другим именем (возможно псевдонимом);
- PTR — обратная связь IP-адреса с именем;
- SRV — запись о сетевой службе;
- TXT — текстовая запись.

ВНИМАНИЕ! Перевод строки в конце конфигурационных файлов зон обязателен. В большинстве применений необходимо указание точки в конце имен компьютеров для предотвращения вывода корневого суффикса имени вида «1.168.192.in-addr.arpa».

6.5.3. Настройка клиентов для работы со службой доменных имен

Для работы со службой доменных имен на компьютерах необходимо наличие конфигурационного файла /etc/resolv.conf, содержащего информацию о доменах и именах серверов DNS, например:

```
domain my.dom
search my.dom
nameserver 192.168.1.100
```

Также может быть рассмотрена установка системы поддержки работы со службой доменных имен, содержащейся в пакете resolvconf.

ВНИМАНИЕ! Для взаимодействия DNS-сервера с клиентами, функционирующими в разных мандатных контекстах, требуется дополнительная настройка механизма privsock. Описание настройки сетевых служб для работы с использованием механизма privsock приведено в документе РУСБ.10015-01 97 01-1.

6.6. Настройка SSH

SSH — это клиент-серверная система для организации защищенных туннелей между двумя и более компьютерами. В туннелях защищаются все передаваемые данные, в т. ч. пароли.

В поставляемую в составе дистрибутива версию пакета ssh встроены алгоритмы защитного преобразования ГОСТ grasshopper-ctr (в соответствии с ГОСТ Р 34.13-2015) и имитовставки hmac-gost2012-256-etm (на основе ГОСТ Р 34.11-2012). Эти алгоритмы используются по умолчанию, их использование не требует специальной настройки.

При этом в список алгоритмов защитного преобразования (параметр конфигурации `Ciphers`) и выработки имитовставки (параметр конфигурации `MACs`), допустимых к использованию, по умолчанию включены следующие алгоритмы защитного преобразования (перечислены в порядке убывания приоритетов применения):

```
grasshopper-ctr,aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,
     aes128-cbc,3des-cbc
```

и алгоритмы выработки имитовставки (перечислены в порядке убывания приоритетов применения):

```
hmac-gost2012-256-etm,hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
```

В конфигурационных файлах клиента (файл `/etc/ssh/ssh_config`) и сервера (файл `/etc/ssh/sshd_config`) имеются закомментированные строчки `Ciphers` и `MACs`, справочно отражающие список алгоритмов, принятых по умолчанию. Если требуется изменить набор допустимых алгоритмов или приоритеты их применения, следует раскомментировать данную строчку и указать нужные алгоритмы в порядке приоритета их выполнения.

Например, для приоритетного выбора более быстрых (а значит и более простых) алгоритмов можно использовать следующие параметры конфигурации:

```
Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc
MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
```

ВНИМАНИЕ! Использование простых алгоритмов снижает безопасность системы.

Проверить списки поддерживаемых алгоритмов можно следующими командами:

```
# список алгоритмов защитного преобразования:
ssh -Q cipher
# список алгоритмов выработки имитовставки:
ssh -Q mac
```

Дополнительная информация по применению `ssh` доступна на официальном сайте разработчика `wiki.astralinux.ru`.

6.6.1. Служба ssh

Служба `ssh` (синоним `sshd`) может быть установлена при установке ОС. При этом служба будет запущена автоматически после завершения установки и перезагрузки, что обеспечит удаленный доступ к установленной ОС для выполнения дальнейших настроек.

При необходимости служба может быть установлена отдельно:

```
apt install ssh
```

Проверить состояние службы:

```
systemctl status ssh
```

Служба берет свои конфигурации сначала из командной строки, затем из файла `/etc/ssh/sshd_config`. Синтаксис:

```
sshd [-deiqtD46] [-b bits] [-f config_file] [-g login_grace_time]
[-h host_key_file] [-k key_gen_time] [-o option] [-p port] [-u len]
```

Параметры, которые могут присутствовать в файле `/etc/ssh/sshd_config`, описаны в таблице 22. Пустые строки, а также строки, начинающиеся с `#`, игнорируются. Названия параметров не чувствительны к регистру символов.

Таблица 22

Параметр	Описание
`AllowGroups`	Задает список групп, разделенный пробелами, которые будут допущены в систему
`DenyGroups`	Действие, противоположное действию параметра `AllowGroups`: записанные в данный параметр группы не будут допущены в систему
`AllowUsers`	Задает разделенный пробелами список пользователей, которые получат доступ в систему. По умолчанию доступ разрешен всем пользователям
`DenyUsers`	Действие, противоположное действию параметра `AllowUsers`: записанные в данный параметр пользователи не получат доступ в систему
`AFSTokenPassing`	Указывает на то, может ли маркер `AFS` пересылаться на сервер. Значение по умолчанию `yes`
`AllowTCPForwarding`	Указывает на то, разрешены ли запросы на переадресацию портов. Значение по умолчанию `yes`
`Banner`	Отображает полный путь к файлу сообщения, выводимого перед аутентификацией пользователя
`ChallengeResponseAuthentication`	Указывает на то, разрешена ли аутентификация по методу «оклик — ответ». Значение по умолчанию `yes`
`Ciphers`	Задает разделенный запятыми список методов защиты соединения, разрешенных для использования

Продолжение таблицы 22

Параметр	Описание
CheckMail	Указывает на то, должна ли служба sshd проверять почту в интерактивных сеансах регистрации. Значение по умолчанию no
ClientAliveInterval	Задает интервал ожидания в секундах, по истечении которого клиенту посылается запрос на ввод данных
ClientAliveCountMax	Задает число напоминающих запросов, посылаемых клиенту. Если по достижении указанного предела от клиента не поступит данных, сеанс завершается и сервер прекращает работу. Значение по умолчанию 3
HostKey	Полный путь к файлу, содержащему личный ключ компьютера. Значение по умолчанию /etc/ssh/ssh_host_key
GatewayPorts	Указывает на то, могут ли удаленные компьютеры подключаться к портам, для которых клиент запросил переадресацию. Значение по умолчанию no
HostbasedAuthentication	Указывает на то, разрешена ли аутентификация пользователей с проверкой файлов .rhosts и /etc/hosts.equiv и открытого ключа компьютера. Значение по умолчанию no
IgnoreRhosts	Указывает на то, игнорируются ли файлы $HOME/.rhosts и $HOME/.shosts. Значение по умолчанию yes
IgnoreUserKnownHosts	Указывает на то, игнорируется ли файл $HOME/.ssh/known_hosts в режимах аутентификации RhostsRSAAuthentication и HostbasedAuthentication. Значение по умолчанию no
KeepAlive	Если установлено значение yes (по умолчанию), демон sshd будет периодически проверять наличие связи с клиентом. В случае неуспешного завершения проверки соединение разрывается. Для выключения данного механизма задать значение параметра no в файле конфигурации сервера и клиента
KerberosAuthentication	Указывает на то, разрешена ли аутентификация с использованием Kerberos. Значение по умолчанию no
KerberosOrLocalPasswd	Указывает на то, должна ли использоваться локальная парольная аутентификация в случае неуспешной аутентификации на основе Kerberos
KerberosTgtPassing	Указывает на то, может ли структура TGT системы Kerberos пересылаться на сервер. Значение по умолчанию no
KerberosTicketCleanup	Указывает на то, должен ли при выходе пользователя удаляться кэш-файл его пропуска Kerberos

Продолжение таблицы 22

Параметр	Описание
ListenAddress	Задает интерфейс, к которому подключается служба sshd. Значение по умолчанию 0.0.0.0, т. е. любой интерфейс
LoginGraceTime	Задает интервал времени в секундах, в течение которого должна произойти аутентификация пользователя. Если процесс аутентификации не успевает завершиться вовремя, сервер разрывает соединение и завершает работу. Значение по умолчанию 600 с
LogLevel	Задает степень подробности журнальных сообщений. Возможные значения: QUIET, FATAL, ERROR, INFO (по умолчанию), VERBOSE, DEBUG (не рекомендуется)
MACs	Задает разделенный запятыми список доступных алгоритмов MAC (код аутентификации сообщений), используемых для обеспечения целостности данных
MaxStartups	Задает максимальное число одновременных неаутентифицированных соединений с демоном sshd
PAMAuthenticationViaKbdInt	Указывает на то, разрешена ли парольная аутентификация с использованием PAM. Значение по умолчанию no
PasswordAuthentication	Если установлено значение yes (по умолчанию) и ни один механизм беспарольной аутентификации не приносит положительного результата, тогда пользователю выдается приглашение на ввод пароля, который проверяется самим демоном sshd. Если значение параметра no, парольная аутентификация запрещена
PermitEmptyPasswords	Если установлено значение yes, пользователи, не имеющие пароля, могут быть аутентифицированы службой sshd. Если установлено значение no (по умолчанию), пустые пароли запрещены
PermitRootLogin	Указывает на то, может ли пользователь root войти в систему с помощью команды ssh. Возможные значения: no (по умолчанию), without-password, forced-command-only и yes
PidFile	Задает путь к файлу, содержащему идентификатор главного процесса. Значение по умолчанию /var/run/sshd.pid
Port	Задает номер порта, к которому подключается sshd. Значение по умолчанию 22
PrintLastLog	Указывает на то, должна ли служба sshd отображать сообщение о времени последнего доступа. Значение по умолчанию yes
PrintMotd	Указывает на то, следует ли после регистрации в системе отображать содержимое файла /etc/motd. Значение по умолчанию yes

Окончание таблицы 22

Параметр	Описание
Protocol	Задает разделенный запятыми список версий протокола, поддерживаемых службой sshd
PubKeyAuthentication	Указывает на то, разрешена ли аутентификация с использованием открытого ключа. Значение по умолчанию yes
ReverseMappingCheck	Указывает на то, должен ли выполняться обратный поиск имен. Значение по умолчанию no
StrictModes	Если равен yes (по умолчанию), sshd будет запрещать доступ любому пользователю, чей начальный каталог и/или файл .rhosts принадлежат другому пользователю либо открыты для записи
Subsystem	Предназначается для конфигурирования внешней подсистемы. Аргументами является имя подсистемы и команда, выполняемая при поступлении запроса к подсистеме
SyslogFacility	Задает название средства, от имени которого регистрируются события в системе Syslog. Возможны значения: DAEMON, USER, AUTH (по умолчанию), LOCAL0-7
UseLogin	Указывает на то, должна ли применяться команда login для организации интерактивных сеансов регистрации. Значение по умолчанию no
X11Forwarding	Указывает на то, разрешена ли переадресация запросов к системе X Window. Значение по умолчанию no
X11DisplayOffset	Задает номер первого дисплея (сервера) системы X Window, доступного демону sshd для переадресации запросов. Значение по умолчанию 10
XAuthLocation	Задает путь к команде xauth. Значение по умолчанию /usr/X11R6/bin/xauth

6.6.2. Клиент ssh

В роли клиента выступает инструмент командной строки ssh. Синтаксис команды:

```
ssh [-afgknqstvxACNTX1246] [-b bind_address] [-c cipher_spec] [-e escape_char]
[-i identity_file] [-login_name] [-m mac_spec] [-o option] [-p port]
[-F configfile] [-L port:host:hostport] [-R port:host:hostport]
[-D port] hostname | user@hostname [command]
```

Подробное описание параметров инструмента приведено man ssh. В простом варианте инициировать соединение с сервером sshd можно командой:

```
ssh 10.1.1.170
```

где 10.1.1.170 — IP-адрес компьютера с запущенной службой sshd. При этом sshd будет считать, что пользователь, запрашивающий соединение, имеет такое же имя, под каким он аутентифицирован на компьютере-клиенте.

Клиент ssh может заходить на сервер sshd под любым именем, используя параметр:

```
-l <имя_клиента>
```

Однако сервер будет согласовывать ключ сеанса (например, при беспарольной аутентификации по открытому ключу пользователя), проверяя открытые ключи в домашнем каталоге пользователя именно с этим именем на компьютере-клиенте. Если же используется парольная аутентификация, на компьютере-сервере должна существовать учетная запись с таким именем. Использовать беспарольную аутентификацию по открытым ключам компьютера настоятельно не рекомендуется, т. к. при этом способе в системе должны существовать потенциально опасные файлы: /etc/hosts.equiv, /etc/shosts.equiv, $HOME/.rhosts, $HOME/.shosts.

Инструментssh берет свои конфигурационные установки сначала из командной строки, затем из пользовательского файла $HOME/.ssh/config и из общесистемного файла /etc/ssh/ssh_config. Если идентичные параметры заданы по-разному, выбирается самое первое значение.

В таблице 23 описаны параметры, которые могут присутствовать в файле $HOME/.ssh/config или /etc/ssh/ssh_config. Пустые строки и комментарии игнорируются.

Таблица 23

Параметр	Описание
CheckHostIP	Указывает на то, должна ли команда ssh проверять IP-адреса в файле known_hosts. Значение по умолчанию yes
Ciphers	Задает разделенный запятыми список методов защиты сеанса, разрешенных для использования. По умолчанию aes122b-cbc, 3des-cbc, blowfish-cbc, cast128-cbc, arcfour, aes192-cbc, aes256-cbc
Compression	Указывает на то, должны ли данные сжиматься с помощью команды gzip. Значение по умолчанию no. Эта установка может быть переопределена с помощью параметра командной строки -C
ConnectionAttempts	Задает число неудачных попыток подключения (одна в секунду), после чего произойдет завершение работы. Значение по умолчанию 4

Продолжение таблицы 23

Параметр	Описание
EscapeChar	Задает escape-символ, используемый для отмены специального назначения следующего символа в сеансах с псевдотерминалом. Значение по умолчанию ~. Значение none запрещает использование escape-символа
ForwardAgent	Указывает на то, будет ли запрос к команде ssh-agent переадресован на удаленный сервер. Значение по умолчанию no
ForwardX11	Указывает на то, будут ли запросы к системе X Window автоматически переадресовываться через SSH-туннель с одновременной установкой переменной среды DISPLAY. Значение по умолчанию no
GatewayPorts	Указывает на то, могут ли удаленные компьютеры подключаться к локальным портам, для которых включен режим переадресации. Значение по умолчанию no
GlobalKnownHostsFile	Задает файл, в котором хранится глобальная база ключей компьютера. По умолчанию глобальная база ключей компьютера хранится в файле /etc/ssh/ssh_known_hosts
HostbasedAuthentication	Указывает на то, разрешена ли аутентификация пользователей с проверкой файлов .rhosts, /etc/hosts.equiv и открытого ключа компьютера. Этот параметр рекомендуется установить в значение no
HostKeyAlgorithm	Задает алгоритмы получения ключей компьютеров в порядке приоритета. Значение по умолчанию ssh-rsa, ssh-dss
HostKeyAlias	Задает псевдоним, который должен использоваться при поиске и сохранении ключей компьютера
HostName	Задает имя или IP-адрес компьютера, на котором следует регистрироваться. По умолчанию выбирается имя, указанное в командной строке
IdentityFile	Задает файл, содержащей личный ключ пользователя. Значение по умолчанию $HOME/.ssh/identity. Вместо имени начального каталога пользователя может стоять символ ~. Разрешается иметь несколько таких файлов. Все они будут проверены в указанном порядке
KeepAlive	Если равен yes (по умолчанию), команда ssh будет периодически проверять наличие связи с сервером. В случае неуспешного завершения проверки (в т. ч. из-за временных проблем с маршрутизацией) соединение разрывается. Чтобы выключить этот механизм, следует задать данный параметр, равным no, в файлах /etc/ssh/sshd_config и /etc/ssh/ssh_config либо в файле $HOME/.ssh/config
KerberosAuthentication	Указывает на то, разрешена ли аутентификация с применением Kerberos
KerberosTgtPassing	Указывает на то, будет ли структура TGT системы Kerberos пересылаться на сервер

Продолжение таблицы 23

Параметр	Описание
LocalForward	Требует значения в формате порт:узел:удаленный_порт. Указывает на то, что запросы к соответствующему локальному порту перенаправляются на заданный порт удаленного узла
LogLevel	Задает степень подробности журнальных сообщений команды ssh. Возможные значения: QUIET, FATAL, ERROR, INFO (по умолчанию), VERBOSE, DEBUG
MACs	Задает разделенный запятыми список доступных алгоритмов аутентификации сообщений для обеспечения целостности данных. Стандартный выбор: hmac-md5, hmac-sha1, hmac-ripemd160@openssh.com, hmac-sha1-96, hmac-md5-96
NumberOfPasswordPrompts	Задает число допустимых попыток ввода пароля. Значение по умолчанию 3
PasswordAuthentication	Если равен yes (по умолчанию), то в случае необходимости команда ssh пытается провести парольную аутентификацию
Port	Задает номер порта сервера. Значение по умолчанию 22
PreferredAuthentications	Задает порядок применения методов аутентификации. Значение по умолчанию: publickey, password, keyboard-interactive
Protocol	Задает в порядке приоритета версии протокола SSH
ProxyCommand	Задает команду, которую следует использовать вместо ssh для подключения к серверу. Эта команда выполняется интерпретатором /bin/sh. Спецификация %p соответствует номеру порта, а %h — имени удаленного узла
PubkeyAuthentication	Указывает на то, разрешена ли аутентификация с использованием открытого ключа. Значение по умолчанию yes
RemoteForward	Требует значения в формате удаленный_порт:узел:порт. Указывает на то, что запросы к соответствующему удаленному порту перенаправляются на заданный порт заданного узла. Переадресация запросов к привилегированным портам разрешена только после получения прав суперпользователя на удаленной системе. Эта установка может быть переопределена с помощью параметра командной строки -R
StrictHostKeyCheking	Если равен yes, команда не будет автоматически добавлять ключи компьютера в файл $HOME/.ssh/known_hosts и откажется устанавливать соединение с компьютерами, ключи которых изменились. Если равен no, команда будет добавлять непроверенные ключи сервера в указанные файлы. Если равен ask (по умолчанию), команда будет спрашивать пользователя о том, следует ли добавлять открытый ключ сервера в указанные файлы
UsePrivilegedPort	Указывает на то, можно ли использовать привилегированный порт для установления исходящих соединений. Значение по умолчанию no

Окончание таблицы 23

Параметр	Описание
`User`	Задает пользователя, от имени которого следует регистрироваться в удаленной системе. Эта установка может быть переопределена с помощью параметра командной строки `-l`
`UserKnownHostsFile`	Задает файл, который используется для автоматического обновления открытых ключей
`XAuthLocation`	Задает путь к команде `xauth`. Значение по умолчанию `/usr/X11R6/bin/xauth`

Клиентские конфигурационные файлы бывают глобальными, на уровне системы (`/etc/ssh/ssh_config`), и локальными, на уровне пользователя (`$HOME/.ssh/config`). Следовательно, пользователь может полностью контролировать конфигурацию клиентской части SSH.

Конфигурационные файлы разбиты на разделы, установки которых относятся к отдельному компьютеру, группе компьютеров или ко всем компьютерам. Установки разных разделов могут перекрывать друг друга.

6.7. Службы точного времени

В состав ОС входят следующие службы точного времени:

1) службы, использующие протокол синхронизации времени NTP:
   а) `systemd-timesyncd` — клиентская служба синхронизации времени, используется в ОС по умолчанию. Описание службы приведено в 6.7.1;
   б) `chronyd` — клиент и сервер протокола точного времени NTP. Описание службы приведено в 6.7.2;
2) служба времени высокой точности PTP (Precision Time Protocol) — описание службы приведено в 6.7.3.

При настройке служб времени используются термины для обозначения времени, приведенные в таблице 24.

Таблица 24

Термин	Описание	Пример
Universal time, UTC	UTC (Coordinated Universal Time) — всемирное координированное время. Не зависит от местоположения компьютера, используется в качестве системного времени: времени в ядре ОС, для отметок времени записи журналов и для синхронизации времени службами времени	Universal time: Ср 2019-02-20 07:51:49 UTC

Окончание таблицы 24

Термин	Описание	Пример
Time Zone	Временная зона. Определяет временное смещение и параметры сезонного (зимнего/летнего) времени.	Time zone: Europe/Moscow (MSK, +0300)
Local time	Локальное время (местное время). Получается из всемирного координированного времени добавлением временного смещения. Используется в основном для взаимодействия с пользователями системы	Local time: Ср 2019-02-20 10:51:49 MSK
RTC time	Аппаратное время, установленное в аппаратных часах компьютера (Real Time Clock, RTC, также CMOS или BIOS time). Используется для первоначальной установки времени при загрузке ОС. Аппаратные часы могут быть настроены как на всемирное координированное, так и на местное время. При установке системного времени на основании показаний аппаратных часов ОС принимает решение о том, какое именно время (UTC или местное) показывают аппаратные часы, на основании собственных внутренних настроек (см. `man timedatectl`)	RTC time: Ср 2019-02-20 07:51:49

6.7.1. Служба systemd-timesyncd

Служба `systemd-timesyncd` предназначена для использования в роли клиента и не может выполнять функции сервера точного времени. Подходит для синхронизации времени с доверенным сервером времени в локальной сети. Может применяться в системах, где не требуется высокая точность синхронизации времени. Поддерживает только упрощенный протокол передачи времени.

6.7.1.1. Установка и настройка

Служба `systemd-timesyncd` устанавливается автоматически, если при установке ОС был выбран для установки компонент «Консольные утилиты».

Служба синхронизации запускается автоматически, если при установке ОС для настройки времени была выбрана синхронизация времени по сети.

Для запуска службы синхронизации времени вручную и для ее добавления в автозапуск выполнить команду:

```
systemctl enable systemd-timesyncd
```

Служба не может работать одновременно со службами `ntpd` или `chronyd` (не будет выполняться синхронизация времени). Служба завершает свою работу без сообщений об ошибке, если обнаружит на компьютере:

- установленную службу `ntpd` (даже незапущенную);

- установленную службу chronyd (даже незапущенную);
- для виртуальных машин — установленные гостевые дополнения Oracle Virtual Box.

Запись о завершении работы systemd-timesyncd будет внесена в системный журнал /var/log/syslog.

Состояние службы systemd-timesyncd можно проверить командой:

```
systemctl status systemd-timesyncd
```

Также для проверки статуса службы синхронизации времени можно использовать команду:

```
timedatectl status
```

Примечание. Необходимо учитывать, что timedatectl может использоваться и с другими службами синхронизации времени. Таким образом, если просматривать статус службы времени, то отображается статус запущенной в данной момент службы. А при использовании timedatectl, например, для запуска службы времени будет запущена установленная в системе служба.

Пример

Вывод команды timedatectl status

```
Local time: Ср 2018-12-26 11:08:12 MSK
Universal time: Ср 2018-12-26 08:08:12 UTC
RTC time: Ср 2018-12-26 08:08:12
Time zone: Europe/Moscow (MSK, +0300)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

Автоматический запуск службы отключается командой:

```
systemctl disable systemd-timesyncd
```

или

```
timedatectl set-ntp false
```

6.7.1.2. Выбор серверов времени

Основные и резервные серверы времени указываются в конфигурационных файлах службы `systemd-timesyncd`:

1) `/etc/systemd/timesyncd.conf`
2) `/etc/systemd/timesyncd.conf.d/*.conf`;
3) `/run/systemd/timesyncd.conf.d/*.conf`;
4) `/usr/lib/systemd/timesyncd.conf.d/*.conf`.

Основные параметры в конфигурационном файле:

1) `NTP=` — список имен основных серверов единого времени, разделенный пробелами. Объединяется со списком имен, полученных от службы `systemd-networkd`. По умолчанию список пустой и используются резервные серверы времени, указанные в параметре `FallbackNTP=`;
2) `FallbackNTP=` — список имен резервных серверов единого времени, разделенный пробелами.

Служба `systemd-timesyncd` перебирает по очереди серверы из основного списка и, если не удалось связаться ни с одним из серверов, обращается к серверам из резервного списка.

По умолчанию для службы `systemd-timesyncd` указаны российские серверы точного времени ВНИИФТРИ.

Дополнительно служба `systemd-timesyncd` может получать имена серверов времени от службы `systemd-networkd`, если в конфигурационных файлах этой службы (каталоги `/lib/systemd/network/`, `/run/systemd/network/`, `/etc/systemd/network/` или файл `/lib/`) указаны серверы единого времени, привязанные к сетевым интерфейсам.

Более подробная информация о службе `systemd-networkd` приведена в `man systemd.network`.

6.7.2. Служба chronyd

Служба точного времени `chronyd` рекомендована к применению вместо службы `ntpd`. Служба `chronyd` может выступать в роли клиента и сервера протокола сетевого времени NTP и позволяет:

- быстрее синхронизировать системные часы;
- использовать аппаратные метки времени, что обеспечивает более точную синхронизацию времени;
- не прекращать работу службы синхронизации, обнаружив слишком большое отклонение времени, а попытаться выполнить коррекцию времени;

- работать, если порт 123 закрыт для исходящих запросов.

Службы chronyd обеспечивает надежную работу синхронизации при нестабильных сетевых соединениях, частичной доступности или перегрузки сети.

6.7.2.1. Установка

При установке ОС служба chronyd автоматически не устанавливается. Для установки службы требуется установить пакет chrony (при этом будет удален пакет установленной ранее службы systemd-timesyncd):

```
apt install chrony
```

ВНИМАНИЕ! При установке контроллера домена FreeIPA пакет chrony будет установлен автоматически, при этом будет удален пакет systemd-timesyncd.

6.7.2.2. Настройка

В режиме клиента служба chronyd может запускаться с настройками по умолчанию без конфигурационного файла.

По умолчанию для службы chronyd указаны российские серверы точного времени ВНИИФТРИ.

Для настройки работы службы chronyd в режиме сервера времени (т. е. чтобы служба отвечала клиентам на запросы) необходимо отредактировать (или при его отсутствии — создать) конфигурационный файл /etc/chrony/chrony.conf. В конфигурационном файле требуется добавить строку с разрешениями клиентам подключаться к серверу NTP.

Пример

Настройка разрешений подключаться к NTP-серверу:

1) разрешить всем клиентам:

```
allow
```

2) разрешить только клиенту с определенным IP-адресом:

```
allow 10.10.12.5
```

3) разрешить клиентам определенной сети:

```
allow 10.10.12
```

Более подробная информация о настройке конфигурационного файла /etc/chrony/chrony.conf приведена в man chrony.conf.

После редактирования конфигурационного файла перезапустить службу chronyd:

```
systemctl restart chronyd
```

6.7.3. Служба времени высокой точности PTP

Служба времени высокой точности PTP включает следующие службы:

- ptp4l — служба протокола времени высокой точности, реализующая работу по протоколу времени высокой точности PTP в соответствии со стандартом IEEE 1588. Точность протокола зависит от способа установки отметок времени (time stamping) в пакетах IEEE 1588. При программном методе установки отметок времени обеспечивается точность 1-100 микросекунд, на точность влияют прерывания, загрузка процессора и иные факторы. Аппаратная поддержка обеспечивает точность до единиц микросекунд;
- phc2sys — служба синхронизации часов;
- timemaster — служба координации, обеспечивающая совместную работу службы времени NTP (chronyd) и службы времени высокой точности PTP.

6.7.3.1. Проверка оборудования

Служба времени высокой точности ориентирована на использование аппаратных средств точного времени, в частности, аппаратных возможностей сетевых карт (аппаратные отметки времени).

Проверить, поддерживает ли сетевая карта аппаратные отметки времени, можно из командной строки с помощью инструмента ethtool. Для этого необходимо:

1) установить пакет ethtool, если он не был установлен ранее, выполнив команду:

```
apt install ethtool
```

2) проверить оборудование, выполнив команду:

```
ethtool -T eth0
```

Если сетевая карта не поддерживает аппаратные отметки времени, возможно настроить и использовать службу времени высокой точности на основе программных отметок времени, но это повлечет снижение точности. Настройка использования сетевых карт без аппаратной поддержки отметок времени приведена в 6.7.3.3.

6.7.3.2. Установка службы PTP

Служба времени высокой точности PTP устанавливается из пакета `linuxptp` командой:

```
apt install linuxptp
```

6.7.3.3. Настройка службы ptp4l

Для включения службы `ptp4l` необходимо раскомментировать в конфигурационном файле `/etc/linuxptp/timemaster.conf` секцию домена точного времени `[ptp_domain 0]` и указать данные сетевой карты.

Пример

Настройки домена точного времени, использующего интерфейс `eth0`:

```
[ptp_domain 0]
interfaces eth0
delay 10e-6
```

Домен точного времени обслуживается службой `ptp4l`.

Настройка службы `tp4l` осуществляется с помощью конфигурационного файла `/etc/linuxptp/ptp4l.conf`.

При использовании сетевых карт без аппаратной поддержки отметок времени необходимо в конфигурационном файле `/etc/linuxptp/ptp4l.conf` в параметре `time_stamping` заменить аппаратную поддержку (`hardware`) на программную (`software`):

```
time_stamping software
```

Подробное описание настроек конфигурационного файла приведено в `man pdp4l`.

6.7.3.4. Настройка службы timemaster

Настройка службы `timemaster` осуществляется с помощью конфигурационного файла `/etc/linuxptp/timemaster.conf`.

Необходимо разрешить автоматический запуск службы `timemaster` при старте ОС, выполнив команду:

```
systemctl enable timemaster
```

Подробно параметры настройки описаны в `man timemaster`.

6.7.3.5. Настройка службы phc2sys

Служба `phc2sys` не требует настройки. Если в системе установлена сетевая карта, поддерживающая аппаратные отметки времени, которую необходимо синхронизировать с системными часами RTC, служба `phc2sys` запускается автоматически с нужными параметрами.

При работе с сетевыми картами, не поддерживающими аппаратные отметки времени, служба `phc2sys` не запускается.

6.7.3.6. Запуск службы PTP

После завершения настройки запуск всех служб осуществляется командой:

```
systemctl start timemaster
```

Служба `timemaster` запустит все остальные службы.

Пример

Проверка состояния службы `timemaster`:

```
systemctl status timemaster
```

Результат выполнения команды при штатном функционировании и наличии аппаратной поддержки:

```
timemaster.service - Synchronize system clock to NTP and PTP time sources
Loaded: loaded (/lib/systemd/system/timemaster.service; enabled; preset:
      enabled)
Active: active (running) since Thu 2024-02-22 12:33:42 MSK; 2s ago
Docs: man:timemaster
Main PID: 32390 (timemaster)
Tasks: 3 (limit: 4001)
Memory: 1.1M
CPU: 12ms
CGroup: /system.slice/timemaster.service
   32390 /usr/sbin/timemaster -f /etc/linuxptp/timemaster.conf
   32391 /usr/sbin/chronyd -n -f /var/run/timemaster/chrony.conf
   32392 /usr/sbin/ptp4l -l 5 -f /var/run/timemaster/ptp4l.0.conf -H -i
        eth0
   32393 /usr/sbin/phc2sys -l 5 -a -r -R 1.00 -z /var/run/timemaster/
         ptp4l.0.socket -n 0 -E ntpshm -M 0
```

6.7.3.7. Настройка режима интерпретации показаний аппаратных часов

Чтобы исключить проблемы с коррекцией времени и сменой сезонного локального времени, рекомендуется настраивать аппаратные часы на всемирное координированное время (UTC). По умолчанию ОС настроена так, чтобы показания аппаратных часов трактовались как время UTC.

Режим интерпретации показаний аппаратных часов может быть включен при установке ОС. После установки ОС режим интерпретации показаний аппаратных часов включается с помощью графической утилиты `fly-admin-date` путем установки во вкладке «Дата и время» флага «Системные часы установлены на UTC».

Проверка показаний системного, локального и аппаратного времени выполняется командой:

```
timedatectl
```

Если ОС настроена так, что показания аппаратных часов трактуются как локальное время, при выполнении команды `timadatectl` будет выдано соответствующее предупреждение.

Настройка аппаратных часов на время UTC с одновременной синхронизацией с системным временем выполняется командой:

```
timedatectl set-local-rtc 0
```

Для настройки с одновременной синхронизацией системного времени по показаниям часов RTC следует использовать параметр `--adjust-system-clock`.

Настройка аппаратных часов на локальное время выполняется командой:

```
timedatectl set-local-rtc 1
```

6.7.4. Ручная синхронизация времени ntpdate

Инструмент `ntpdate` применяется для проверки работы сервера времени и/или синхронизации с ним системного времени.

Ручная синхронизация времени может применяться для:

1) проверки, независимой от запущенных служб времени, степени рассинхронизации времени;
2) проверки доступности серверов через порт 123.

Инструмент устанавливается в ОС по умолчанию.

Запускать необходимо с правами `root`. Возможен запуск как из командной строки (вручную), так и из стартового сценария, выполняемого при загрузке ОС. Возможно выполнение `ntpdate` по расписанию из сценария `cron` для периодической коррекции времени.

Для установки инструмента выполнить команду:

```
sudo apt install ntpdate
```

Синтаксис команды:

```
ntpdate [-параметры] <NTP-сервер>
```

Основные параметры инструмента приведены в таблице 25.

Таблица 25

Параметр	Описание
`-a <ключ>`	Разрешение аутентификации и указание ключа для использования. По умолчанию аутентификация отключена
`-d`	Проверка доступности сервера времени запросом времени с подробной диагностикой без коррекции показаний локальных часов
`-q`	Проверка доступности сервера времени запросом времени без коррекции показаний локальных часов
`-u`	Предписывает использовать для запроса времени IP-порт, отличный от 123. По умолчанию `ntpdate` использует тот же IP-порт (123) что и служба `ntpd`, и, если служба `ntpd` запущена, то `ntpdate` при запуске выдаст ошибку, что порт занят. Также IP-порт 123 может быть закрыт для обеспечения безопасности
`-b`	Принудительное пошаговая коррекция времени с помощью вызова функции `settimeofday()`. Параметр следует использовать при вызове из файла запуска во время начальной загрузки

Например, для осуществление периодической коррекции времени выполнить команду:

```
ntpdate -ubv 0.ru.pool.ntp.org
```

Более подробная информация приведена в `man ntpdate`.

6.8. Программный коммутатор Open vSwitch

Open vSwitch (OVS) — это многоуровневый программный коммутатор, который поддерживает стандартные интерфейсы управления. Open vSwitch используется для работы в качестве виртуального коммутатора в средах виртуальных машин. В дополнение к стандартным интер-

фейсам управления и видимости на уровне виртуальной сети, Open vSwitch поддерживает распределение между несколькими физическими серверами.

Коммутатор Open vSwitch поддерживает несколько технологий виртуализации на базе Linux, включая KVM и VirtualBox.

В коммутаторе OVS реализованы следующие функции:

- стандартная модель VLAN 802.1Q с магистральными портами доступа;
- NIC teaming (bonding) — объединение интерфейсов сетевых карт (как поддерживающих LACP, так и не поддерживающих) на порте вышестоящего коммутатора;
- протоколы NetFlow, sFlow(R) для мониторинга и анализа состояния сети;
- зеркалирование для повышения видимости;
- конфигурация QoS (качество обслуживания) и его контроль;
- туннелирование Geneve, GRE, VXLAN, STT и LISP;
- управление сбоями подключения 802.1ag;
- OpenFlow 1.0.

Коммутатор OVS может полностью работать как на уровне ядра, так и в пользовательском пространстве без модуля ядра. OVS в пользовательском пространстве может получить доступ к устройствам Linux или DPDK.

Коммутатор OVS поддерживает фильтрацию сетевого потока на основе классификационных меток при включенном в ОС мандатном управлении доступом. Порядок настройки параметров фильтрации описан в РУСБ.10015-01 97 01-1.

6.8.1. Основные модули

Основными компонентами OVS являются:

1) `ovs-vswitchd` — демон, обеспечивающий работу OVS;
2) `ovsdb-server` — сервер баз данных для хранения конфигурации `ovs-vswitchd`;
3) `ovs-dpctl` — инструмент командной строки для настройки коммутатора;
4) сценарии и спецификации для создания пакетов `deb`;
5) `ovs-vsctl` — инструмент командной строки для запроса и обновления конфигурации `ovs-vswitchd`;
6) `ovs-appctl` — инструмент командной строки для отправки команд запущенным демонам OVS;
7) `ovs-ofctl` — инструмент командной строки для управления коммутаторами и контроллерами OpenFlow;

8) `ovs-pki` — инструмент командной строки для создания и управления инфраструктурой открытых ключей для коммутаторов OpenFlow;

9) `ovs-testcontroller` — контроллер OpenFlow, может использоваться для тестирования;

10) модуль для `tcpdump`, который позволяет анализировать сообщения OpenFlow.

6.8.2. Установка и настройка Open vSwitch

Для установки OVS выполнить команду:

```
sudo apt install openvswitch-switch openvswitch-common
```

Пакеты `openvswitch-switch` и `openvswitch-common` содержат основные компоненты пользовательского пространства коммутатора. Также доступны дополнительные пакеты с документацией на OVS, поддержкой IPsec, PKI, VTEP и Python.

Для переключения пользовательского пространства требуется установить пакет `openvswitch-switch-dpdk`, предоставляющий Open vSwitch с поддержкой DPDK.

6.8.3. Добавление сетевого моста и портов

Добавление сетевого моста осуществляется командой:

```
sudo ovs-vsctl add-br <имя_сетевого_моста>
```

Добавление порта осуществляется командой:

```
sudo ovs-vsctl add-port <имя_сетевого_моста> <порт> [<параметры>]
```

Указываемый в команде сетевой мост должен присутствовать в системе.

Пример

Для создания сетевого моста в OVS требуется:

1) запустить OVS для создания БД:

```
sudo /usr/share/openvswitch/scripts/ovs-ctl start
```

2) сконфигурировать порты, выполнив команды:

```
sudo ip tuntap add mode tap tap0
sudo ifconfig tap0 up
sudo ip tuntap add mode tap tap1
sudo ifconfig tap1 up
```

3) создать сетевой мост и добавить на него порты, выполнив команды:

```
sudo ovs-vsctl add-br br0
sudo ovs-vsctl add-port br0 tap0
sudo ovs-vsctl add-port br0 tap1
```

4) запустить сетевой мост:

```
sudo ifconfig br0 up
```

5) добавить физический интерфейс, выполнив команды:

```
sudo ovs-vsctl add-port br0 eth0
sudo ifconfig eth0 0
sudo dhclient br0
```

Проверить добавление портов возможно командой:

```
sudo ovs-vsctl show
```

Результат выполнения команды:

```
517c8376-7b07-45e9-9f6f-d0844efd0207
    Bridge br0
        Port eth0
            Interface eth0
        Port tap1
            Interface tap1
        Port br0
            Interface br0
                type: internal
        Port tap0
            Interface tap0
    ovs_version: "3.1.0"
```

6.8.4. Конфигурирование правил обработки пакетов

Правила обработки пакетов реализуются через OpenFlow — протокол управления процессом обработки данных, передающихся по сети маршрутизаторами и коммутаторами, реализующий технологию программно-конфигурируемой сети.

Примеры:

1. Правило, блокирующее пакеты, которые в своем уровне IP имеют `ip_dst=87.250.250.242`:

```
ovs-ofctl add-flow br0 ip,nw_dst=87.250.250.242,actions=drop
```

2. Правило, отбрасывающее все IP-пакеты:

```
ovs-ofctl add-flow br0 ip,actions=drop
```

3. Правило, отправляющее пакеты обратно на порт с индексом 2:

```
ovs-ofctl add-flow br0 in_port=2,actions=in_port
```

4. Правило, устанавливающее новые MAC-адреса в пакете и возвращающее этот пакет обратно на тот же порт:

```
ovs-ofctl add-flow br0 in_port=vport1,
actions=mod_dl_dst=08:00:27:8e:fc:42,
mod_dl_src=08:00:27:03:6a:e3,in_port
```

Правила OpenFlow, помимо блокировки и пропуска пакетов, также могут менять поля в пакете.

Пример

Правило, устанавливающее новые MAC-адреса в пакете и возвращающее этот пакет обратно на тот же порт:

```
ovs-ofctl add-flow br0 in_port=vport1,
actions=mod_dl_dst=08:00:27:8e:fc:42,
mod_dl_src=08:00:27:03:6a:e3,in_port
```

Возможно задание правил, учитывающих порт, VLAN и протоколы, также правилу можно задать приоритет.

6.8.5. Регистрация событий

В программном коммутаторе OVS присутствуют встроенные средства регистрации событий. Также коммутатор OVS из состава ОС доработан для реализации возможности регистрации событий безопасности и аудита IP-пакетов.

6.8.5.1. Встроенные средства регистрации

Настройка встроенных средств регистрации событий осуществляется с помощью инструмента командной строки `ovs-vswitchd`.

Для управления регистрацией событий используется команда:

```
sudo ovs-vswitchd -v <модуль>:<способ>:<уровень>
```

```
[-v <модуль>:<способ>:<уровень> ...]
```

где <модуль> — модуль из состава OVS, для которого настраивается регистрация событий;
<способ> — способ регистрации событий;
<уровень> — уровень регистрируемых событий.

Коммутатор OVS состоит из модулей и настройка регистрации событий выполняется для каждого из модулей отдельно.

Список модулей OVS с настроенными для них уровнями регистрации можно вывести командой:

```
sudo ovs-appctl vlog/list
```

Встроенные средства регистрации событий коммутатора OVS обеспечивают регистрацию следующими способами: запись в системный журнал /var/log/syslog, запись в заданный файл, вывод в терминал, а также отправка информации о событиях на удаленный компьютер.

Если настраивается регистрация событий в файл, то должна быть включена возможность регистрации событий в файл командой:

```
sudo ovs-vswitchd --log-file[=<имя_файла>]
```

Если файл не указан, то по умолчанию события будут регистрироваться в файл /var/log/openvswitch/ovs-vswitchd.log.

Для настройки отправки информации о событиях на удаленный компьютер выполнить команду:

```
sudo ovs-vswitchd --syslog-target=<IP-адрес>:<порт>
```

Вывод информации о событиях в терминал по умолчанию отключен.

Также регистрируемые события разделены на уровни, определяющие серьезность события. При указании уровня выполняется регистрация событий заданного уровня и более высоких уровней. Наивысший уровень off, при котором регистрация событий не выполняется. Информация об уровнях регистрации приведена в man ovs-appctl.

Регистрация событий в системном журнале /var/log/syslog по умолчанию выполняется автоматически для уровней emer и err.

Дополнительная информация по использованию инструмента командной строки `ovs-vswitchd` для настройки регистрации событий приведена в `man ovs-vswitchd`.

Настройка регистрации событий после запуска службы `ovs-vswitchd` также может осуществляться с помощью инструмента командной строки `ovs-appctl`:

```
sudo ovs-appctl vlog/set <модуль>:<способ>:<уровень>
     [<модуль>:<способ>:<уровень> ...]
```

Дополнительная информация по использованию инструмента командной строки `ovs-appctl` приведена в `man ovs-appctl`.

Для просмотра системного журнала `/var/log/syslog` может использоваться графическая утилита `ksystemlog`, описание утилиты приведено в электронной справке.

6.8.5.2. Регистрация событий безопасности

В коммутаторе OVS из состава ОС реализована регистрация следующих событий безопасности:

- запуск и остановка OVS;
- изменение конфигурации OVS;
- действия с правилами OpenFlow;
- изменение классификационных меток.

Регистрация событий безопасности выполняется подсистемой регистрации событий, описание которой приведено в 17.2.

Также регистрация событий безопасности выполняется автоматически в системном журнале `/var/log/syslog`, запись о событии имеет метку `ovs_audit`. Для просмотра системного журнала `/var/log/syslog` может использоваться графическая утилита `ksystemlog`, описание утилиты приведено в электронной справке.

6.8.5.3. Аудит IP-пакетов

В коммутаторе OVS из состава ОС реализован аудит IP-пакетов, к которым была применена фильтрация на основе классификационных меток (см. РУСБ.10015-01 97 01-1). Регистрируется информация о пропуске или блокировке IP-пакетов на основе правил OpenFlow.

Для настройки аудита с выводом информации о событиях в журнал `/var/log/syslog` выполнить команду:

```
sudo ovs-appctl vlog/set syslog:ovs_mac:dbg
```

Для настройки аудита с выводом информации о событиях в терминал выполнить команду:

```
sudo ovs-appctl vlog/set console:ovs_mac:dbg
```

Сообщения аудита IP-пакетов имеют метку `ovs_mac`. Для просмотра системного журнала `/var/log/syslog` может использоваться графическая утилита `ksystemlog`, описание утилиты приведено в электронной справке.

6.9. Сетевая защищенная файловая система

6.9.1. Назначение и возможности

Для организации защищенных файловых серверов предназначена сетевая защищенная ФС (СЗФС), в основу которой положена CIFS, работающая по протоколу SMB/CIFS. Протокол СЗФС содержит в себе сообщения, которые передают информацию о мандатном контексте (метке безопасности и дополнительных мандатных атрибутах управления доступом) субъекта доступа. Подробное описание мандатого контекста приведено в документе РУСБ.10015-01 97 01-1.

Условием корректного функционирования СЗФС является использование механизма ЕПП, обеспечивающее в рамках данной ЛВС однозначное соответствие между логическим именем пользователя и его идентификатором (а также именем группы и ее идентификатором) на всех компьютерах (рабочих станциях и серверах), на которых данный пользователь может работать. Для корректной работы СЗФС необходима синхронизация UID/GID в системах клиента и сервера, т. к. информация о пользователях и группах передается в сеть в численных значениях.

СЗФС предоставляет следующие базовые возможности:

- разделение операционной системой типа Windows файловой системы ОС и наоборот;
- совместное использование принтеров, подключенных к ОС, операционной системой типа Windows и наоборот.

6.9.2. Состав

Основой СЗФС является клиент-серверная архитектура.

Сервер представляет собой расширенный сервер Samba и выполняет следующие задачи:

1) управление совместно используемые ресурсами;
2) контроль доступа к совместно используемым ресурсам. При подключении клиента сервер устанавливает метку безопасности процесса, обслуживающего запросы

клиента, в соответствии с меткой безопасности этого клиента. Этим обеспечивается мандатный контроль доступа к совместно используемым файлам на стороне сервера.

Клиент представляет собой сетевую ФС в составе системы управления файлами ядра ОС и реализует интерфейс между виртуальной ФС ядра и сервером СЗФС. Клиент СЗФС выполняет следующие задачи:

1) отображение каталогов и файлов смонтированного сетевого ресурса;

2) передача на сервер дополнительной информации о классификационной метке пользователя (процесса), работающего с совместно используемым ресурсом.

С точки зрения пользователя, СЗФС выглядит как стандартная ФС, поддерживающая все механизмы защиты ОС и позволяющая работать с удаленной ФС с помощью стандартных команд.

В состав СЗФС входят следующие компоненты:

- `smbd` — служба сервера, которая обеспечивает работу службы печати и разделения файлов для клиентов операционной системы типа Windows. Конфигурационные параметры службы `smbd` описываются в файле `smb.conf`;

- `nmbd` — служба сервера, которая обеспечивает работу службы имен NetBIOS, а также может использоваться для запроса других служб имен;

- `smbclient` — служба, которую реализует клиент, используемый для доступа к другим серверам и для печати на принтерах, подключенных к серверам;

- `testparm` — команда, позволяющая протестировать конфигурационный файл `smb.conf`;

- `smbstatus` — команда, выводящая информацию о том, кто в настоящее время пользуется сервером Samba.

6.9.3. Настройка

СЗФС устанавливается в процессе установки ОС.

Основная настройка СЗФС в ОС осуществляется путем редактирования конфигурационного файла `/etc/samba/smb.conf`.

Файл `/etc/samba/smb.conf` состоит из основных именованных разделов `[global]`, `[homes]` и `[printers]`, возможно добавление пользовательских разделов. Внутри каждого раздела находится ряд параметров вида `<имя_параметра> = <значение>`.

В разделе `[global]` описаны параметры, управляющие сервером Samba в целом, а также находятся значения параметров по умолчанию для других разделов.

Примеры:

1. Фрагмент конфигурационного файла, определяющий рабочую группу WORKGR1, к которой относится компьютер, а также описывающий саму систему:

```
[global];
;workgroup = NT-Domain-Name или Workgroup-Name
workgroup = WORKGR1
;comment эквивалентен полю описания NT (Description field)
comment = Сервер СЗФС
```

2. Фрагмент конфигурационного файла, описывающий тип системы печати, доступной на сервере администратора, а также местонахождение конфигурационного файла принтера. Последняя строка говорит о том, что все принтеры, определенные в файле printcap, должны быть доступны в сети:

```
;printing = BSD или SYSV или A1X (и т.д.)
printing = bsd
printcap name = /etc/printcap
load printers = yes
```

3. Фрагмент конфигурационного файла, определяющий поддержку сервером гостевого входа. Следующие два параметра определяют работу с журнальными файлами. Параметр m сообщает службе Samba, что для каждого клиента ведется свой файл, а последняя строка говорит о том, что максимальный размер создаваемого журнального файла — 50 КБ:

```
;Раскомментируйте это поле, если вам нужен гостевой вход
;guest = pcguest
log file = /var/log/samba-log.%m
max log size = 50
```

Раздел [homes] позволяет подключаться к рабочим каталогам пользователей без их явного описания. При запросе клиентом определенной службы ищется соответствующее ей описание в файле и, если такового нет, просматривается раздел [homes]. Если этот раздел существует, просматривается файл паролей для поиска рабочего каталога пользователя, сделавшего запрос, и, найдя его, он становится доступным по сети. Основные параметры раздела [homes]:

1) comment — значение параметра выводится для клиента при запросе о доступных ресурсах;

2) browseable — определяет, как выводить ресурс в списке просмотра;

3) `read only` — определяет, может ли пользователь создавать и изменять файлы в своем рабочем каталоге при подключении по сети;

4) `create mask` — определяет права доступа для вновь создаваемых файлов в рабочем каталоге пользователя.

Пример

```
[homes]
comment = Home Directories
browseable = no
case sensitive = yes
read only = yes
create mask = 0700
directory mask = 0700
ea support = yes
```

Раздел `[printers]` используется для предоставления доступа к принтерам, определенным в файле `/etc/printcap`. В разделе `[printers]` описываются параметры управления печатью при отсутствии иного явного описания. Параметры `comment`, `browseable`, `read only`, `create mask` аналогичны параметрам раздела `[homes]`, остальные параметры:

1) `path` — определяет местонахождение файла спулера при печати через SMB;

2) `printable` — определяет, может ли использоваться данный ресурс для печати;

3) `guest ok` — определяет, может ли воспользоваться принтером гостевой пользователь.

Пример

```
[printers]
comment = All Printers
browseable = no
path = /var/spool/samba
printable = no
guest ok = no
read only = yes
create mask = 0700
```

После настройки параметров сервера по умолчанию можно создать совместно используемые каталоги, доступ к которым могут получать определенные пользователи, группы пользователей или все пользователи.

Пример

Создание совместно используемого каталога с доступом только для одного пользователя. Для этого необходимо создать отдельный раздел файла `smb.conf` и заполнить его необходимой информацией (обычно это пользователь, каталог и конфигурационная информация)

```
[User1]
comment = User1’ s remote source code directory
path = /usr/locaI/src
valid users = victor
browseable = yes
public = no
writeable = yes
create mode = 0700
```

В данном разделе создается совместно используемый каталог с именем `User1`. На локальном сервере его путь — `/usr/local/src`, `browseable = yes`, поэтому ресурс будет виден в списках ресурсов сети, но т. к. `public = no`, получить доступ к нему сможет только пользователь `victor`. Предоставить доступ другим пользователям можно, поместив их в запись `valid users`.

По умолчанию сервер Samba поддерживает подключение по протоколу SMB всех версий, а клиент при подключении начинает процедуру согласования протокола подключения со старшей версии. Для принудительного определения диапазона возможных протоколов используются параметры конфигурационного файла `/etc/samba/smb.conf`, приведенные в таблице 26.

Таблица 26

Имя параметра	Синоним параметра	Значение по умолчанию	Описание
`server min protocol`	`min protocol`	`NT1`	Минимальная версия протокола сервера
`server max protocol`	`max protocol`, `protocol`	`SMB3_11`	Максимальная версия протокола сервера
`client min protocol`		`NT1`	Минимальная версия протокола клиента
`client max protocol`		`SMB3_11`	Максимальная версия протокола клиента
`client ipc min protocol`		`NT1` (значение параметра `client min protocol`)	Минимальная версия протокола клиента для межпроцессного взаимодействия

Окончание таблицы 26

Имя параметра	Синоним параметра	Значение по умолчанию	Описание
`client ipc max protocol`		`SMB3_11`	Максимальная версия протокола клиента для межпроцессного взаимодействия

Допустимые значения параметров, указанных в таблице 26, для каждой версии протокола приведены в таблице 27.

Таблица 27

Версия протокола	Значение	Примечание
SMB v1	`NT1`	

SMB v2	`SMB2`
	`SMB2_02`
	`SMB2_10`
	`SMB2_22`
	`SMB2_24`
		`SMB2 = SMB2_10`

SMB v3	`SMB3`
	`SMB3_00`
	`SMB3_02`
	`SMB3_10`
	`SMB3_11`
		`SMB3 = SMB3_11`

В зависимости от реализации клиент Samba может принудительно требовать от сервера версию протокола. Обычно версия протокола задается одним из параметров подключения и имеет собственную нотацию. Способы конфигурирования протокола в зависимости от типа клиента, а также допустимые значения приведены в таблице 28.

Таблица 28

Утилита	Конфигурирование	Допустимые значения	Значение по умолчанию

`mount.cifs`	Применение параметра монтирования `vers=`	`1.0`
		`2.0`
		`2.1`
		`3.0`
		`3.02`
		`3.1.1`
		`3.11`
			`3.11`

`smbclient`	Использование параметра `-m`, `--max-protocol` с инструментом командной строки	`NT1`
		`SMB2`
		`SMB3`
			Определяется параметрами в `/etc/samba/smb.conf`

После редактирования конфигурационного файла `/etc/smb.conf` необходимо протестировать его корректность при помощи команды `testparm`, которая проверяет наличие в файле внутренних противоречий и несоответствий.

Примечание. Выполнение `testparm` не подтверждает, что все службы и ресурсы, описанные в конфигурационном файле, доступны и будут корректно работать.

Команда `testparm` имеет следующий синтаксис:

```
testparm [configfile [hostname hostip]]
```

Параметр `configfile` определяет местоположение конфигурационного файла (если это не файл `/etc/smb.conf`). Параметр `hostname hostip` указывает команде `testparm` проверить доступ к службам со стороны узла, определяемого параметром.

Если ошибки не будут обнаружены, на экране появится сообщение вида:

```
it testparm
Load smb config files from /etc/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
Press enter to see a dump of your service definitions
```

При нажатии клавиши **<Enter>** `testparm` протестирует каждый раздел, определенный в конфигурационном файле.

В случае обнаружения ошибок о них будет предоставлена полная информация.

6.9.4. Графическая утилита настройки СЗФС

В состав ОС входит графическая утилита `fly-admin-samba`, которая позволяет настроить пользовательский доступ к ресурсам СЗФС. Установка утилиты выполняется командой:

```
apt install fly-admin-samba
```

Описание использования утилиты приведено в электронной справке.

6.9.5. Запуск сервера

Сервер запускается либо из инициализирующих сценариев, либо из `inetd` в качестве системной службы.

Если сервер запускается из сценариев инициализации, то можно воспользоваться для запуска и остановки работы сервера следующей командой:

```
systemctl {start|stop} smbd
```

Доступ пользователей ОС к ресурсам сервера осуществляется с помощью монтирования СЗФС. Другой возможностью является использование графической утилиты `fly-admin-samba` (см. электронную справку).

Инструмент командной строки `smbclient` позволяет получить информацию о совместно используемых ресурсах или перенести файлы. Например, для запроса списка доступных ресурсов на удаленном сервере `win.netwhart.com` используется команда:

```
smbclient -L -I win.netwhart.com
```

где `-L` — указывает, что требуется вывести список совместно используемых ресурсов;
`-I` — указывает, что указанное далее имя следует рассматривать как имя DNS, а не NetBIOS.

Для пересылки файла необходимо сначала подключиться к серверу путем выполнения команды:

```
smbclient '\\WORKGR1\PUBLIC' -I win.netwhart.com -U tackett
```

где `\\WORKGR1\PUBLIC` — определяет удаленную службу на другом компьютере (обычно это каталог ФС или принтер);
`-U` — позволяет определить имя пользователя для подключения к ресурсу (при этом, если необходимо, СЗФС запросит соответствующий пароль).

После подключения появится приглашение:

```
Smb:\
```

где `\` — текущий рабочий каталог.

Используя инструмент командной строки `smbclient` можно указать команды для передачи файлов и работы с ними. Дополнительно описание параметров инструмента приведено в руководстве `man smbclient`.

6.9.6. Правила конвертации меток целостности

В ОС используется метка целостности, которая может принимать значение 256 и более.

Для штатной работы СЗФС Samba из состава ОС с СЗФС Samba других систем, в которых максимальное значение метки целостности составляет 255, реализована совместимость меток целостности. При передаче из ОС файла с меткой целостности, значение которой составляет 256 или более, в систему с максимальным значением метки целостности равным 255, метка целостности передаваемого файла будет преобразована в максимальное значение 255, т.е. будет выполнено понижение целостности при передаче файла.

Подробное описание метки целостности ОС приведено в документе РУСБ.10015-01 97 01-1.

6.10. Средство создания защищенных каналов

Для создания между компьютерами сети защищенных каналов типа точка-точка или сервер-клиент используется свободная реализация технологии виртуальной частной сети (VPN) с открытым исходным кодом OpenVPN. Данная технология позволяет устанавливать соединения между компьютерами, находящимися за NAT и сетевым экраном, без необходимости изменения их настроек.

ВНИМАНИЕ! OpenVPN не является сертифицированным криптографическим средством защиты информации и не может применяться в целях криптографической защиты информации. Основное назначение OpenVPN в составе ОС — обеспечение целостности заголовка IP-пакетов, содержащего классификационную метку, при передаче по сетям связи.

Для обеспечения безопасности управляющего канала и потока данных OpenVPN использует библиотеку `OpenSSL` (устанавливается автоматически при установке ОС). При этом OpenVPN использует алгоритмы защитного преобразования `OpenSSL` в соответствии с требованиями ГОСТ (пакет библиотеки алгоритмов ГОСТ `libgost-astra`).

Дополнительная информация по применению OpenVPN и библиотеки алгоритмов ГОСТ `libgost-astra` доступна на сайте `wiki.astralinux.ru`.

6.10.1. Установка

Установка программного продукта OpenVPN выполняется либо из графического менеджера пакетов Synaptic, либо из терминала.

Для установки OpenVPN из терминала необходимо:

1) на компьютере, предназначенном на роль сервера OpenVPN, и на клиентских компьютерах установить пакет `openvpn`:

```
apt install openvpn
```

2) на компьютере, предназначенном на роль сервера, для управления службой `openvpn` установить графическую утилиту `fly-admin-openvpn-server` или ин-

струмент командной строки astra-openvpn-server, выполнив соответствующую команду:

```
apt install fly-admin-openvpn-server
apt install astra-openvpn-server
```

Примечания:

1. При установке графической утилиты автоматически будет установлен инструмент командной строки astra-openvpn-server.

2. При установке инструмента командной строки astra-openvpn-server будет автоматически установлен и настроен пакет алгоритмов защитного преобразования ГОСТ libgost-astra.

6.10.2. Управление с помощью инструмента командной строки

6.10.2.1. Параметры инструмента командной строки

Команды, используемые с инструментом командной строки astra-openvpn-server, приведены в таблице 29.

Таблица 29

Параметр	Описание
Информационные команды	
-h, --help	Вывод справки
-v, --version	Вывод версии
--show-ciphers	Вывод списка поддерживаемых ключей
Управление выводом	
-s	Не выводить сообщения и предупреждения. Может быть указана в любом месте. Отменяет вывод комментариев о ходе выполнения, предупреждений, сообщений об ошибках
Управление сервером	
start	Запустить службу openvpn. При выполнении этой команды без указания дополнительных параметров служба будет запущена со стандартной конфигурацией из файла /etc/openvpn/server.conf. Если файл конфигурации, ключи и сертификаты сервера не существуют, то они будут созданы с параметрами по умолчанию. С данной командой дополнительно могут быть заданы параметры сервера, указаны файлы для аутентификации и параметры аутентификации
stop	Остановить службу. После выполнения данной команды другие команды не выполняются
status	Проверить службу. После выполнения данной команды другие команды не выполняются

Продолжение таблицы 29

Параметр	Описание
`rebuild-server-certs`	Остановить службу, удалить все сертификаты сервера и клиентов, повторно сгенерировать все сертификаты сервера и запустить сервер. Имена файлов сертификатов сервера берутся из файла конфигурации сервера. Если файл конфигурации отсутствует, то остальные действия не выполняются. После выполнения данной команды другие команды не выполняются
Параметры сервера	
`server <IP-адрес> <маска>`	IP-адрес и маска создаваемой сети VPN (по умолчанию IP-адрес `10.8.0.0` и маска `255.255.255.0`), например: `astra-openvpn-server server "10.8.0.0 255.255.255.0"`
`port <порт>`	Порт (по умолчанию 1194)
`cipher <метод>`	Метод защитного преобразования данных (по умолчанию `grasshopper-cbc`). Поддерживаются следующие методы защитного преобразования: - `grasshopper-cbc` — алгоритм «Кузнечик» ГОСТ Р 34.13-2015; - `AES-256-GCM` — рекомендован для применения в системах общего назначения; - `AES-256-CBC` — допустим для применения в системах общего назначения; - `AES-128-CBC` — используется для совместимости со старыми системами, к применению не рекомендуется
Указание файлов для аутентификации	
`cert <имя_файла>.crt`	Файл сертификата пользователя
`ca <имя_файла>.crt`	Файл сертификата центра аутентификации
`key <имя_файла>.key`	Личный ключ
`dh <имя_файла>.pem`	Файл Диффи-Хеллмана
`tls-auth <имя_файла>.key`	Файл аутентификации TLS
Параметры аутентификации	
`EASYRSA_REQ_COUNTRY`	Название страны
`EASYRSA_REQ_PROVINCE`	Название области
`EASYRSA_REQ_CITY`	Название города
`EASYRSA_REQ_ORG`	Название организации
`EASYRSA_REQ_EMAIL`	Адрес электронной почты
`EASYRSA_REQ_OU`	Название подразделения организации
`EASYRSA_REQ_CN`	Имя пользователя
Генерация и отзыв ключей клиентов	

Окончание таблицы 29

Параметр	Описание
`client <имя_клиента>`	Создать ключи и сертификаты для указанного клиента
`revoke <имя_клиента>`	Отозвать сертификат указанного клиента
Параметры индивидуальной настройки сервера	
`get <параметр>`	Прочитать значение параметра из файла конфигурации `/etc/openvpn/server.conf`. Если файл конфигурации не существует, то он будет создан с параметрами по умолчанию
`del <параметр>`	Удалить значение параметра из файла конфигурации `/etc/openvpn/server.conf`. Если файл конфигурации не существует, то он будет создан с параметрами по умолчанию, после чего указанный параметра будет удалн
`set <параметр> <значение>`	Записать значение параметра в файл конфигурации `/etc/openvpn/server.conf`. Если файл конфигурации не существует, то он будет создан с параметрами по умолчанию, после чего в файл будет записано указанное значение

Примечания:

1. Если в командной строке заданы информационные команды, то будет выполнена первая из них. Дальнейшее выполнение сценария будет прекращено.
2. Команды управления сервером несовместимы с командами генерации и отзыва ключей для клиентов.
3. Полный список параметров индивидуальной настройки сервера доступен в документации на OpenVPN.

6.10.2.2. Запуск службы

Для запуска службы `openvpn` из терминала ввести команду:

```
astra-openvpn-server start
```

При запуске службы будут созданы следующие стандартные файлы и каталоги:

- файл конфигурации службы `openvpn`:

`/etc/openvpn/server.conf`

- локальный центр аутентификации, размещается в каталоге:

`/etc/openvpn/openvpn-certificates`

- сертификат открытого ключа центра аутентификации:

`/etc/openvpn/keys/ca.crt`

- сертификат открытого ключа:

`/etc/openvpn/keys/server.crt`

- закрытый ключ сервера:

`/etc/openvpn/keys/server.key`

- файл параметров Диффи-Хеллмана для аутентификации пользователей:

`/etc/openvpn/keys/dh2048.pem`

- файл дополнительной аутентификации TLS:

`/etc/openvpn/keys/ta.key`

- дополнительно, при выполнении отзыва сертификатов, будет создан стандартный файл списка отзыва сертификатов:

`/etc/openvpn/keys/crl.pem`

Также при первом запуске службы будут выполнены настройки межсетевого экрана и другие настройки ОС для работы `openvpn` как стандартной системной службы с автоматическим запуском при включении компьютера.

Запуск команды `astra-openvpn-server start` с указанием файлов для аутентификации (см. таблицу 29) позволяет при создании файла конфигурации и запуске службы `openvpn` задать расположение ранее установленных файлов ключей и сертификатов.

ВНИМАНИЕ! Чтобы избежать запроса пароля при автоматическом запуске службы `openvpn` необходимо файлы создавать без применения защитного преобразования.

Пример

Запуск сервера с указанием ранее установленных файлов ключей и сертификатов

```
astra-openvpn-server start cert /root/secrets/server.crt \
    ca /root/secrets/ca.crt key /root/secrets/server.key \
    dh /root/secrets/dh2048.pem tls-auth /root/secrets/ta.key
```

Указание файлов для аутентификации несовместимо с указанием параметров идентификации (см. таблицу 29).

ВНИМАНИЕ! В случае если указан хотя бы один файл для аутентификации, то все файлы будут проверены на существование. При отсутствии одного из файлов сценарий будет завершен с ошибкой без выполнения каких-либо действий. Проверка файлов на корректность не выполняется.

ВНИМАНИЕ! Если заданы файлы для аутентификации, то создание собственного центра аутентификации не выполняется.

6.10.2.3. Генерация сертификатов и ключей

При использовании собственного центра аутентификации создание ключей и сертификатов для клиентов осуществляется на сервере OpenVPN с помощью инструмента командной строки `astra-openvpn-server`. Для создания клиентского комплекта файлов используется команда `client`:

```
astra-openvpn-server client <имя_клиента>
```

При генерации могут быть заданы параметры аутентификации (см. таблицу 29).

Команда генерации ключей клиента несовместима с параметрами сервера и командами управления сервером (см. таблицу 29).

При выполнении данной команды для указанного клиента будет создан новый файл закрытого ключа `<имя_клиента>.key` и файл сертификата открытого ключа `<имя_клиента>.crt`, подписанный центром аутентификации.

Для удобства последующей передачи файлов ключей клиенту, созданные файлы будут скопированы в каталог `/etc/openvpn/clients-keys/<имя_клиента>`. Дополнительно в каталог будут скопированы и другие, необходимые для работы клиента, файлы: файл сертификата центра аутентификации (по умолчанию `ca.crt`) и файл дополнительной аутентификации TLS (`ta.key`).

Дополнительно при создании пользовательских ключей могут быть указаны такие параметры аутентификации, как страна, город, организация и др. (см. таблицу 29). В таблице 29 приведены значения параметров аутентификации, используемые по умолчанию при генерации сертификатов.

ВНИМАНИЕ! Если задан любой из параметров аутентификации, то будет произведена автоматическая генерация сертификатов.

Пример

Задание дополнительных параметров аутентификации при выполнении команды создания сертификатов для клиента:

```
astra-openvpn-server client ivanov \
EASYRSA_REQ_COUNTRY RU \
EASYRSA_REQ_PROVINCE MO \
EASYRSA_REQ_CITY MOSCOW \
EASYRSA_REQ_ORG COMPANY \
EASYRSA_REQ_EMAIL ivanov@company.ru
```

ВНИМАНИЕ! Клиентские ключи генерируются без применения защитных преобразований, чтобы избежать ввода пароля при подключении клиента к серверу.

Параметры аутентификации несовместимы с указанием файлов для аутентификации (см. таблицу 29).

6.10.2.4. Отзыв сертификатов

Отзыв сертификатов применяется для запрета подключений клиента даже в тех случаях, когда в распоряжении клиента имеются копии всех сертификатов и ключей.

Для отзыва сертификата используется команда `revoke` инструмента командной строки `astra-openvpn-server`:

```
astra-openvpn-server revoke <имя_клиента>
```

Команда отзыва ключей клиента несовместима с параметрами сервера и командами управления сервером (см. таблицу 29).

При выполнении данной команды:

- сертификат клиента в базе данных центра аутентификации будет помечен как «отозванный»;
- будет создан (или обновлен ранее созданный) список отозванных сертификатов;
- новый список отозванных сертификатов будет скопирован в каталог `/etc/openvpn/keys`, сервер OpenVPN будет автоматически перезапущен для применения обновлений.

6.10.2.5. Замена сертификатов

Полная замена сертификатов сервера выполняется с помощью инструмента командной строки `astra-openvpn-server`:

```
astra-openvpn-server rebuild-server-certs
```

При выполнении данной команды:

- останавливается служба `openvpn`;
- удаляются все файлы центра аутентификации;
- удаляются все копии сертификатов сервера и клиентов;
- создается новый центр аутентификации;
- создаются новые сертификаты сервера;
- повторно запускается сервер.

Имена файлов сертификатов сервера берутся из файла конфигурации сервера. Если файл конфигурации отсутствует, то никакие действия не выполняются. После выполнения данной команды другие команды не выполняются.

6.10.2.6. Настройка клиента

На компьютер клиента должны быть перенесены файлы ключей и сертификатов, созданные на сервере, либо с помощью отчуждаемого носителя информации, либо путем передачи по защищенному соединению (например, `ssh`).

Для настройки компьютера клиента следует установить программное обеспечение OpenVPN. Установка выполняется либо из графического менеджера пакетов Synaptic, либо из терминала командой:

```
apt install openvpn
```

После установки программного обеспечения OpenVPN следует выполнить следующие действия:

1) создать файл конфигурации клиента. В качестве исходного файла возможно использовать входящий в комплект установки OpenVPN стандартный шаблон файла конфигурации, предоставляемый разработчиками OpenVPN. Шаблон файла конфигурации расположен в `/usr/share/doc/openvpn/examples/sample-config-files/client.conf`. Шаблон файла следует скопировать в каталог `/etc/openvpn/client`, выполнив команду:

```
cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf \
     /etc/openvpn/client
```

2) в скопированном файле конфигурации внести следующие исправления:

а) для параметра `remote` указать в качестве его значения IP-адрес сервера OpenVPN. Если был изменен порт, то также указать данное значение вместо стандартного;

б) в строках:

```
;user nobody
;group nogroup
```

удалить начальные символы «`;`»:

```
user nobody
group nogroup
```

в) для параметров `ca`, `cert` и `key` указать расположение соответствующих файлов сертификатов и ключа для аутентификации, например:

```
ca /etc/openvpn/keys/ca.crt
```

```
cert /etc/openvpn/keys/home-pc.crt
key /etc/openvpn/keys/home-pc.key
```

г) для параметра `tls-auth` указать расположение файла дополнительной аутентификации TLS, например:

```
tls-auth /etc/openvpn/keys/ta.key
```

д) для параметра `cipher` указать метод защитного преобразования данных, используемый службой. Используемый метод защитного преобразования можно узнать на сервере OpenVPN с помощью инструмента командной строки `astra-openvpn-server` (команда `astra-openvpn-server get cipher`), либо с помощью графического инструмента `fly-admin-openvpn-server`. Защитному преобразованию в соответствии с алгоритмами «Кузнечик» по ГОСТ Р 34.12-2015 соответствует значение `grasshopper-cbc`;

е) сохранить исправленный файл.

Для проверки работы клиента OpenVPN из командной строки использовать команду:

```
/usr/sbin/openvpn --config /etc/openvpn/client/client.conf
```

где `client.conf` — конфигурационный файл клиента.

Для запуска клиента OpenVPN в качестве службы выполнить команду:

```
systemctl start openvpn-client@<имя_файла_конфигурации>
```

где <имя_файла_конфигурации> — имя файла конфигурации без расширения, расположенного в каталоге `/etc/openvpn/client`.

6.10.3. Управление службой с помощью графической утилиты

В графической утилите `fly-admin-openvpn-server` («Настройка OpenVPN сервера Fly») доступны:

- вкладка «Настройки» — в ней располагаются элементы управления для настройки сервера OpenVPN. По умолчанию доступны базовые настройки, расширенные настройки становятся доступны после нажатия кнопки **[Показать расширенные настройки]**. Описание настроек приведено в 6.10.3.2;
- вкладка «Клиентские сертификаты» — в ней располагаются элементы управления клиентскими сертификатами. Описание управления сертификатами приведено в 6.10.3.3;
- кнопки **[Запустить]** и **[Остановить]** — служат для управления службой `openvpn`.

6.10.3.1. Управление службой

Для запуска службы `openvpn` с помощью графической утилиты необходимо:

1) запустить `fly-admin-openvpn-server`. При первом запуске будет создан файл конфигурации службы `openvpn` по умолчанию и будут выпущены сертификаты сервера;

2) при необходимости отредактировать файл конфигурации и сертификаты;

3) нажать кнопку **[Запустить]**.

ВНИМАНИЕ! Графическая утилита при ее запуске не производит автоматический запуск службы `openvpn`.

При запуске службы будут созданы следующие стандартные файлы и каталоги:

- файл конфигурации службы `openvpn`:

`/etc/openvpn/server.conf`

- локальный центр аутентификации, размещается в каталоге:

`/etc/openvpn/openvpn-certificates`

- сертификат открытого ключа центра аутентификации:

`/etc/openvpn/keys/ca.crt`

- сертификат открытого ключа:

`/etc/openvpn/keys/server.crt`

- закрытый ключ сервера:

`/etc/openvpn/keys/server.key`

- файл параметров Диффи-Хеллмана для аутентификации пользователей:

`/etc/openvpn/keys/dh2048.pem`

- файл дополнительной аутентификации TLS:

`/etc/openvpn/keys/ta.key`

- дополнительно, при выполнении отзыва сертификатов, будет создан стандартный файл списка отзыва сертификатов:

`/etc/openvpn/keys/crl.pem`

В случае, если на компьютере установлены и настроены библиотеки, поддерживающие метод защитного преобразования по алгоритму ГОСТ Р 34.12-2015 («Кузнечик»), для защиты канала данных будет выбран данный метод. В противном случае будет выбран метод защитного преобразования `AES-256-GCM`.

Также при первом запуске службы будут выполнены настройки межсетевого экрана и другие настройки ОС для работы `openvpn` как стандартной системной службы с автоматическим запуском при включении компьютера.

Для остановки службы `openvpn` используя графическую утилиту необходимо нажать кнопку **[Остановить]**.

6.10.3.2. Настройка службы

Настройка службы выполняется во вкладке «Настройки» графической утилиты.

Базовые настройки включают:

1) «IP-адрес» — позволяет задать IP-адрес создаваемой сети VPN. По умолчанию установлено значение `10.8.0.0`;

2) «Маска» — позволяет задать маску создаваемой сети VPN. По умолчанию установлено значение `255.255.255.0`;

3) «Порт» — сетевой порт сервера, который будут использовать клиенты для подключения. По умолчанию установлено значение `1194`. Поддерживаются номера свободных портов от 1 до 65535;

4) «Метод защитного преобразования» — по умолчанию установлено значение `grasshopper-cbc` («Кузнечик»). Поддерживаются следующие методы:

а) `grasshopper-cbc` — алгоритм «Кузнечик» ГОСТ Р 34.13-2015;

б) `AES-256-GCM` — рекомендован для применения в системах общего назначения;

в) `AES-256-CBC` — допустим для применения в системах общего назначения;

г) `AES-128-CBC` — используется для совместимости со старыми системами, к применению не рекомендуется.

Расширенные настройки позволяют задать расположение ранее предустановленных файлов ключей и сертификатов внешнего центра аутентификации, а также заново выпустить сертификаты локального центра аутентификации.

Для указания расположения ранее предустановленных файлов ключей и сертификатов внешнего центра аутентификации используются следующие поля:

- «Сертификат пользователя» — сертификат открытого ключа;
- «Сертификат ЦС» — сертификат открытого ключа центра аутентификации;
- «Личный ключ» — закрытый ключ сервера;
- «Файл Диффи-Хеллмана» — файл параметров Диффи-Хеллмана;
- «Файл аутентификации TLS» — файл дополнительной аутентификации TLS.

Проверка файлов на корректность не проводится.

Кнопка **[Сбросить сертификаты]** предназначена для удаления всех сертификатов локального центра аутентификации и повторного выпуска сертификатов сервера. После выполнения этого действия сертификаты клиентов станут недействительными, и клиенты потеряют возможность подключения к серверу OpenVPN. При выполнении данного действия:

- останавливается служба `openvpn`;
- удаляются все файлы центра аутентификации;
- удаляются все копии сертификатов сервера и клиентов;
- создается новый центр аутентификации;
- создаются новые сертификаты сервера;
- повторно запускается сервер.

6.10.3.3. Управление сертификатами

Управление сертификатами выполняется во вкладке «Клиентские сертификаты» графической утилиты.

В данной вкладке расположены таблица с данными о клиентских сертификатах и кнопки управления:

1) **[Создать сертификат]** — создание ключа и сертификата пользователя. При нажатии на кнопку будет открыто диалоговое окно с полями:

а) «Имя пользователя» — имя сертификата. Имя сертификата должно быть уникальным, не может быть пустым и не может содержать пробелы;

б) «Страна» — двухбуквенный код страны. Если поле пустое, то по умолчанию будет установлено значение «RU»;

в) «Область» — название области. Если поле пустое, то по умолчанию будет установлено значение «MO»;

г) «Город» — название города. Если поле пустое, то по умолчанию будет установлено значение «Moscow»;

д) «Организация» — название организации. Если поле пустое, то по умолчанию будет установлено значение «none»;

е) «Email» — адрес электронной почты. Если поле пустое, то по умолчанию будет установлено значение «none»;

ж) «Отдел» — название подразделения организации. Если поле пустое, то по умолчанию будет установлено значение «none»;

з) «Имя» — имя пользователя. Если поле пустое, то по умолчанию будет установлено значение «noname»;

При нажатии на кнопку **[Да]** будет создан новый файл закрытого ключа `<имя_клиента>.key` и файл сертификата открытого ключа `<имя_клиента>.crt`, подписанный центром аутентификации.

Для удобства последующей передачи файлов ключей клиенту, созданные файлы будут скопированы в каталог `/etc/openvpn/clients-keys/<имя_клиента>`. Дополнительно в каталог будут скопированы и другие, необходимые для работы клиента, файлы: файл сертификата центра аутентификации (по умолчанию `ca.crt`) и файл дополнительной аутентификации TLS (`ta.key`).

ВНИМАНИЕ! Клиентские ключи генерируются без применения защитных преобразований, чтобы избежать ввода пароля при подключении клиента к серверу;

2) **[Отозвать сертификат]** — отзыв клиентских сертификатов. Отзыв сертификатов применяется для запрета подключений клиента даже в тех случаях, когда в распоряжении клиента имеются копии всех сертификатов и ключей. Для отзыва сертификата выбрать в таблице клиентов строку с отзываемым сертификатом и нажать данную кнопку. При нажатии на данную кнопку будут выполнены следующие действия:

- сертификат клиента в базе данных центра аутентификации будет помечен как «отозванный»;
- будет создан (или обновлен ранее созданный) список отозванных сертификатов;
- новый список отозванных сертификатов будет скопирован в каталог `/etc/openvpn/keys`;

3) **[Открыть папку сертификатов]** — открытие каталога `/etc/openvpn/clients_keys` в файловом менеджере.

6.10.3.4. Настройка клиента

Настройка сетевых подключений клиентских компьютеров осуществляется с помощью графической утилиты `network-manager-openvpn`. Установка утилиты выполняется командой:

```
apt install network-manager-openvpn network-manager-openvpn-gnome
```

Для настройки клиентского подключения следует в области уведомлений панели задач нажать левой кнопкой мыши на значок сетевых соединений и в раскрывшемся меню выбрать «Соединения VPN — Добавить VPN соединение» (или «Соединения VPN — Настроить VPN», если создается не первое соединение). В открывшемся окне из выпадающего списка выбрать «OpenVPN» и нажать **[Создать]**.

В открывшемся окне необходимо:

1) в поле «Шлюз» указать IP-адрес ранее запущенного сервера OpenVPN;
2) в поле «Тип» оставить значение по умолчанию «Сертификат TLS»;
3) в поле «Сертификат CA» указать путь к скопированному файлу сертификата центра аутентификации `ca.crt` (6.10.2.3);
4) в поле «Сертификат пользователя» указать путь к скопированному файлу сертификата открытого ключа пользователя `<имя_клиента>.crt` (6.10.2.3);

5) в поле «Приватный ключ Пользователя» указать путь к файлу закрытого ключа `<имя_клиента>.key` (6.10.2.3);

6) нажать кнопку **[Дополнительно]**, в открывшемся окне перейти во вкладку «Аутентификация TLS» и в секции «Дополнительная аутентификация или шифрование TLS» выполнить настройки:

а) из выпадающего списка «Режим» выбрать «TLS-Auth» (режим «TLS-Crypt» следует выбирать, если необходимо использовать защитное преобразование для соединения);

б) в поле «Файл ключа» указать путь к ранее скопированному на компьютер пользователя файлу дополнительной аутентификации TLS (см. 6.10.3.3);

в) из выпадающего списка «Направление ключа» выбрать «1».

Все остальные настройки можно оставить заданными по умолчанию. После нажатия кнопки **[ОК]** созданное VPN-соединение будет сохранено.

Для включения сохранного соединения нужно повторно нажать левой кнопкой мыши на значок сетевых подключений в области уведомлений панели задач, в раскрывшемся меню выбрать «Соединения VPN» и отметить включаемое соединение.

Для экспорта параметров созданного клиентского соединения с целью их повторного использования на других клиентах выполнить следующие действия:

1) нажать левой кнопкой мыши на значок сетевых соединений в области уведомлений панели задач и в раскрывшемся меню выбрать «Соединения VPN — Настроить VPN»;

2) из появившегося списка соединений выбрать нужное соединение, нажать кнопку **[Изменить]**, затем нажать **[Экспортировать]**;

3) указать файл, в который сохранить параметры соединения.

При создании соединения VPN используя ранее сохраненные параметры соединения необходимо:

1) нажать левой кнопкой мыши на значок сетевых соединений в области уведомлений панели задач и в раскрывшемся меню выбрать «Соединения VPN — Добавить VPN соединение»;

2) в открывшемся окне из выпадающего списка выбрать «Импортировать сохраненную конфигурацию VPN» и нажать **[Создать]**;

3) указать путь к файлу с параметрами соединения.

6.10.4. Диагностика работы службы и клиента

В процессе работы службы и клиента OpenVPN информация о событиях записывается в системный журнал сервера или клиента, соответственно.

Для просмотра системного журнала полностью используется команда:

```
journalctl
```

Для просмотра последних событий и вывода новых событий по мере их появления используется команда:

```
journalctl -f
```

Для вывода только новых сообщений от службы `openvpn` по мере их добавления в журнал используется команда:

```
tail -f /var/log/syslog | grep ovpn-server
```

При каждом подключении клиента в журнал сервера записывается информация о параметрах подключения, в том числе о выбранном методе защитного преобразования передаваемых данных для входящего и исходящего каналов.

Для проверки установленного метода защитного преобразования используется команда:

```
grep  "Data Channel: Cipher" /var/log/syslog
```

6.10.5. Использование инструмента XCA для создания собственного центра аутентификации

6.10.5.1. Установка инструмента XCA

Для безопасного и эффективного управления файлами ключей и сертификатов рекомендуется использовать графический инструмент создания и управления центром аутентификации XCA.

Инструмент XCA применяется для создания центра аутентификации (Certification Authority, CA) и инфраструктуры открытых ключей (Public Key Infrastructure, PKI).

Инструмент XCA входит в состав ОС. Установка выполняется либо из графического менеджера пакетов Synaptic, либо из терминала командой:

```
apt install xca
```

После установки инструмент XCA доступен для запуска из меню «Пуск — Утилиты — Цифровые сертификаты XCA» (при использовании классического меню «Пуск»). По умолчанию

инструмент XCA запускается на языке операционной системы. Выбор языка возможно изменить вручную через меню «Файл — Язык».

После первого запуска инструмента XCA необходимо создать новую БД. Для этого:

1) выбрать в меню пункт «Файл — Новая база данных»;
2) указать название и путь размещения БД;
3) нажать **[Сохранить]**.

Перед созданием БД будет запрошена установка пароля для доступа к БД. При нажатии **[Да]** без установки пароля БД будет создана без пароля.

ВНИМАНИЕ! Утеря БД может привести к компрометации или полной неработоспособности систем, использующих выданные центром сертификаты. Рекомендуется разворачивать центр аутентификации на отдельном физическом компьютере, не подключенном к сети, передачу сертификатов осуществлять с помощью съемных носителей информации и принять все возможные меры для ограничения доступа к БД.

6.10.5.2. Подготовка шаблонов

Перед созданием сертификатов для упрощения дальнейшей работы рекомендуется заполнить и сохранить типовые значения полей, которые будут применяться в дальнейшем при создании сертификатов. Для этой цели в инструменте XCA предусмотрен механизм шаблонов.

Для создания нового шаблона перейти во вкладку «Шаблоны» и нажать кнопку **[Новый шаблон]**. Из появившегося списка выбрать типовой шаблон. Новый шаблон будет создан как копия выбранного предустановленного шаблона. В инструменте XCA предусмотрено три предустановленных шаблона:

- [default] CA — предустановленный шаблон сертификата центра аутентификации (ЦА);
- [default] HTTPS_client — предустановленный шаблон сертификата клиента;
- [default] HTTPS_server — предустановленный шаблон сертификата сервера.

Предустановленные шаблоны ориентированы на службу HTTPS, поэтому рекомендуется создать на их основе свои шаблоны, полностью настроенные на службу OpenVPN. Для всех шаблонов во вкладке «Субъект» следует заполнить следующие поля:

- «Внутреннее имя» — любое имя;
- «countryName» — двухбуквенный код страны;
- «stateOrProvinceName» — двухбуквенный код региона;
- «localityName» — название города;
- «organizationName» — название организации;

- «organizationalUnitName» — название структурной единицы внутри организации;
- «commonName» — общедоступное имя;
- «emailAddress» — адрес электронной почты.

При заполнении информационных полей шаблона не рекомендуется использовать кириллицу. Все поля являются необязательными, однако, в шаблоне, как минимум, обязательно должно быть заполнено либо поле «Внутреннее имя», либо поле «commonName».

Дополнительно необходимо внести следующие изменения в предустановленные шаблоны:

1) для шаблона сертификата ЦА — во вкладке «Расширения» проверить корректность данных:

а) тип сертификата «Центр сертификации»;

б) наличие флага «Critical»;

в) наличие флага «Subject Key Identifier». При необходимости уточнить даты и срок действия сертификата.

После корректировки шаблона сохранить его, нажав кнопку **[Да]**;

2) для шаблона сертификата сервера — во вкладке «Расширения» проверить корректность данных:

а) тип сертификата «Конечный субъект»;

б) наличие флага «Critical»;

в) наличие флага «Subject Key Identifier». При необходимости уточнить даты и срок действия сертификата.

Во вкладке «Область применения ключа»:

а) в левом поле «X509v3 Key Usage» должны быть выбраны пункты «Digital Signature» и «Key Encipherment»;

б) в левом поле «X509v3 Key Usage» снять выбор с пункта «Non Repudiation»;

в) в правом поле «X509v3 Extended Key Usage» должен быть выбран пункт «TLS Web Server Authentication».

Во вкладке «Netscape» в поле «Netscape Cert Type» снять выбор с пункта «SSL Server».

После корректировки шаблона сохранить его, нажав кнопку **[Да]**;

3) для шаблона сертификата клиента — во вкладке «Расширения» проверить корректность данных:

а) тип сертификата «Конечный субъект»;

б) наличие флага «Critical»;

в) наличие флага «Subject Key Identifier». При необходимости уточнить даты и срок действия сертификата.

Во вкладке «Область применения ключа»:

а) в левом поле «X509v3 Key Usage» снять выбор с пунктов «Data Encipherment» и «Key Encipherment»;

б) в левом поле «X509v3 Key Usage» должен быть выбран пункт «Key Agreement»;

в) в правом поле «X509v3 Extended Key Usage» должен быть выбран пункт «TLS Web Client Authentication».

Во вкладке «Netscape» в поле «Netscape Cert Type» снять выбор с пунктов «SSL Client» и «S/MIME».

После корректировки шаблона сохранить его, нажав кнопку **[Да]**.

6.10.5.3. Типовая схема применения инструмента XCA

Типовая упрощенная схема применения инструмента XCA включает в себя следующие действия:

1) создание корневого сертификата ЦА;
2) создание закрытого ключа и сертификата открытого ключа сервера;
3) экспорт для использования сервером:
   а) сертификата ЦА в соответствии с 6.10.5.7;
   б) закрытого ключа сервера в соответствии с 6.10.5.8;
   в) сертификата открытого ключа сервера в соответствии с 6.10.5.8;
   г) файла параметров Диффи-Хеллмана в соответствии с 6.10.5.8;
   д) файла параметров дополнительной аутентификации протокола TLS в соответствии с 6.10.5.8;
4) создание закрытого ключа и сертификата открытого ключа клиента;
5) экспорт для использования клиентом:
   а) сертификата ЦА в соответствии с 6.10.5.7;
   б) закрытого ключа клиента в соответствии с 6.10.5.9;
   в) сертификата открытого ключа клиента в соответствии с 6.10.5.9;
   г) файла параметров дополнительной аутентификации протокола TLS в соответствии с 6.10.5.9;
6) повторная генерация сертификатов по мере истечения их срока действия.

Пункты 4) и 5) перечисления выполняются для каждого нового подключаемого клиента. Пункт 6) повторяется для центра аутентификации, сервера и клиентов по мере истечения срока действия их сертификатов.

Процедура экспорта подразумевает копирование необходимых данных в файлы и перенос соответствующих файлов на компьютеры сервера и клиентов с использованием процедур, предотвращающих несанкционированный доступ к передаваемой информации (сменные носители, защищенные каналы связи и др.).

6.10.5.4. Создание корневого сертификата центра аутентификации

Корневой сертификат может быть получен из внешнего ЦА или создан как самозаверенный собственный корневой сертификат.

Для создания самоподписанного корневого сертификата необходимо запустить инструмент XCA и выполнить следующие действия:

1) перейти во вкладку «Сертификаты», нажать **[Новый сертификат]**;

2) в открывшемся окне будет установлен флаг «Создать самозаверенный сертификат» и в поле «Шаблон для нового сертификата» выбрать предустановленный шаблон «[default] CA». Если был создан собственный шаблон, то выбрать его, затем нажать кнопку **[Применить всё]**;

3) перейти во вкладку «Субъект». Если был применен собственный шаблон, то поля формы будут автоматически заполнены данными из выбранного шаблона (кроме поля «Внутреннее имя», которое должно быть указано индивидуально для каждого сертификата). Заполнить следующие незаполненные поля:

   а) «Внутреннее имя» — указать имя сертификата, например, «rootCA»;

   б) «commonName» — указать то же имя — «rootCA»;

   в) нажать кнопку **[Сгенерировать новый ключ]**.
   Будет предложено создать новый закрытый ключ с заданным именем. Проверить параметры ключа: «Тип Ключа: RSA», «Длинна ключа: 2048 bit». Нажать кнопку **[Создать]**, затем нажать **[Да]**;

4) перейти во вкладку «Расширения»:

   а) убедиться, что в поле «Тип» выбран «Центр Сертификации»;

   б) проверить установку флагов «Critical» и «Subject Key Identifier»;

   в) определить период действия сертификата, заполнив поля «Период действия» и «Срок действия»;

5) перейти во вкладку «Область применения ключа», убедиться, что в левом поле «X509v3 Key Usage» выбраны пункты:

   а) «Certificate Sign»;

   б) «CRL Sign»;

6) перейти во вкладку «Netscape», убедиться, что в поле «Netscape Cert Type» выбраны пункты:

   а) «SSL CA»;

   б) «S/MIME CA»;

   в) «Object signing CA»;

7) после проверок нажать **[Да]** для создания сертификата.

После выполнения данных действий в списке сертификатов появится корневой сертификат, который в дальнейшем будет использовать для подписания других сертификатов.

6.10.5.5. Создание сертификата сервера

Для создания сертификата сервера выполнить следующие действия:

1) перейти во вкладку «Сертификаты», нажать **[Новый сертификат]**;
2) в открывшемся окне во вкладке «Первоисточник»:
   а) установить флаг «Использовать этот сертификат для подписи» (флаг «Создать самозаверенный сертификат» будет снят автоматически) и в соответствующем выпадающем списке выбрать созданный согласно 6.10.5.4 корневой сертификат;
   б) в поле «Шаблон для нового сертификата» выбрать предустановленный шаблон «[default] HTTPS_server». Если был создан собственный шаблон, то выбрать его, затем нажать кнопку **[Применить всё]**;
3) перейти во вкладку «Субъект». Если был применен собственный шаблон, то поля формы будут автоматически заполнены данными из выбранного шаблона (кроме поля «Внутреннее имя», которое должно быть указано индивидуально для каждого сертификата). Заполнить следующие незаполненные поля:
   а) «Внутреннее имя» — указать имя сертификата;
   б) «commonName» — указать то же имя;
   в) нажать кнопку **[Сгенерировать новый ключ]**.
   Будет предложено создать новый закрытый ключ с заданным именем. Проверить параметры ключа: «Тип Ключа: RSA», «Длинна ключа: 2048 bit». Нажать кнопку **[Создать]**, затем нажать **[Да]**;
4) перейти во вкладку «Расширения»:
   а) убедиться, что в поле «Тип» выбран «Конечный субъект»;
   б) проверить установку флагов «Critical» и «Subject Key Identifier»;
   в) определить период действия сертификата, заполнив поля «Период действия» и «Срок действия»;
5) перейти во вкладку «Область применения ключа», убедиться, что:
   а) в левом поле «X509v3 Key Usage» выбраны пункты «Digital Signature» и «Key Encipherment»;
   б) в правом поле «X509v3 Extended Key Usage» выбран пункт «TLS Web Server Authentication»;
6) нажать **[Да]** для создания сертификата.

После создания сертификата сервера он отобразится в общем списке сертификатов. Инструмент XCA представляет список сертификатов в виде дерева, корнем которого является корневой сертификат центра аутентификации.

6.10.5.6. Создание сертификата клиента

Для создания сертификата клиента выполнить следующие действия:

1) перейти во вкладку «Сертификаты», нажать **[Новый сертификат]**;
2) в открывшемся окне во вкладке «Первоисточник»:
   а) установить флаг «Использовать этот сертификат для подписи» (флаг «Создать самозаверенный сертификат» будет снят автоматически) и в соответствующем выпадающем списке выбрать созданный согласно 6.10.5.4 корневой сертификат;
   б) в поле «Шаблон для нового сертификата» выбрать предустановленный шаблон «[default] HTTPS_client». Если был создан собственный шаблон, то выбрать его, затем нажать кнопку **[Применить всё]**;
3) перейти во вкладку «Субъект». Если был применен собственный шаблон, то поля формы будут автоматически заполнены данными из выбранного шаблона (кроме поля «Внутреннее имя», которое должно быть указано индивидуально для каждого сертификата). Заполнить следующие незаполненные поля:
   а) «Внутреннее имя» — указать имя сертификата;
   б) «commonName» — указать то же имя;
   в) нажать кнопку **[Сгенерировать новый ключ]**.
   Будет предложено создать новый закрытый ключ с заданным именем. Проверить параметры ключа: «Тип Ключа: RSA», «Длинна ключа: 2048 bit». Нажать кнопку **[Создать]**, затем нажать **[Да]**;
4) перейти во вкладку «Расширения»:
   а) убедиться, что в поле «Тип» выбран «Конечный субъект»;
   б) проверить установку флагов «Critical» и «Subject Key Identifier»;
   в) определить период действия сертификата, заполнив поля «Период действия» и «Срок действия»;
5) перейти во вкладку «Область применения ключа», убедиться, что:
   а) в левом поле «X509v3 Key Usage» выбран пункт «Key Agreement»;
   б) в правом поле «X509v3 Extended Key Usage» выбран пункт «TLS Web Client Authentication»;
6) нажать **[Да]** для создания сертификата.

После создания сертификата клиента он отобразится в общем списке сертификатов.

6.10.5.7. Экспорт корневого сертификата центра аутентификации

Для работы серверов и клиентов нужен только сертификат ЦА. Закрытый корневой сертификат ЦА не должен передаваться в другие системы, однако, его копии следует хранить в системах резервного копирования и восстановления.

Для экспорта корневого сертификата:

- в основном окне программы перейти во вкладку «Сертификаты»;
- в списке выбрать корневой сертификат и нажать кнопку **[Экспорт]**;
- в открывшейся окне указать имя файла контейнера сертификата, место сохранения и выбрать формат экспорта «PEM (`*.crt`)»;
- нажать кнопку **[Да]**.

6.10.5.8. Экспорт файлов сертификатов и ключей сервера

Для экспорта сертификата сервера необходимо:

1) в основном окне программы перейти во вкладку «Сертификаты»;
2) в списке выбрать сертификат сервера и нажать кнопку **[Экспорт]**;
3) в открывшейся окне указать место сохранения и выбрать формат экспорта «PEM (`*.crt`)»;
4) нажать кнопку **[Да]**.

Для экспорта закрытого ключа сервера необходимо:

1) в основном окне программы перейти во вкладку «Закрытые ключи»;
2) в списке выбрать закрытый ключ сервера и нажать кнопку **[Экспорт]**;
3) в открывшейся окне выбрать формат экспорта «Закрытый ключ PEM (`*.pem`)»;
4) нажать кнопку **[Да]**.

Закрытый ключ сервера экспортируется в открытом виде без применения защитного преобразования данных.

Закрытый ключ сервера должен находиться на сервере и не должен передаваться клиентам.

Для создания файла с параметрами Диффи-Хеллмана необходимо:

1) в основном окне программы выбрать в меню «Дополнительно — Сгенерировать параметры Диффи-Хэллмана»;
2) в открывшейся окне указать значение «2048 (2048 бит)»;
3) нажать кнопку **[Да]**.
   Примечание. Генерация занимает много времени, об активности программы свидетельствует индикатор в правом нижнем углу окна программы;
4) в открывшейся окне указать место для сохранения полученного файла;
5) нажать кнопку **[Да]** для сохранения.

Создание файл дополнительной аутентификации протокола TLS в инструменте XCA не предусмотрено. Данный файл должен быть создан отдельно средствами OpenVPN при помощи команды:

```
openvpn --genkey --secret <имя_файла>
```

6.10.5.9. Экспорт файлов сертификатов и ключей клиента

Для экспорта сертификата клиента необходимо:

1) в основном окне программы перейти во вкладку «Сертификаты»;
2) в списке выбрать сертификат клиента и нажать кнопку **[Экспорт]**;
3) в открывшейся окне указать место сохранения и выбрать формат экспорта «PEM (`*.crt`)»;
4) нажать кнопку **[Да]**.

Для экспорта закрытого ключа клиента необходимо:

1) в основном окне программы перейти во вкладку «Закрытые ключи»;
2) в списке выбрать закрытый ключ клиента и нажать кнопку **[Экспорт]**;
3) в открывшейся окне выбрать формат экспорта «Закрытый ключ PEM (`*.pem`)»;
4) нажать кнопку **[Да]**.

Закрытый ключ клиента экспортируется в открытом виде без применения защитного преобразования данных.

6.10.5.10. Отзыв сертификатов. Списки отзыва сертификатов

Для отзыва сертификата необходимо:

1) в основном окне программы перейти во вкладку «Сертификаты»;
2) найти в списке отзываемый сертификат, нажать правой кнопкой мыши и в раскрывшемся меню выбрать «Отозвать».

Аналогичным способом можно отменить отзыв сертификата, выбрав пункт «Вернуть».

Списки отозванных сертификатов привязываются к корневому сертификату ЦА, подписавшего эти сертификаты.

Для просмотра списка отозванных сертификатов, относящихся к корневому сертификату, необходимо:

1) в основном окне программы перейти во вкладку «Сертификаты»;

2) в списке выбрать корневой сертификат, нажать правой кнопкой мыши и в раскрывшемся меню выбрать «ЦС — Управление отзывами».

Откроется список отозванных сертификатов.

Для создания списка отозванных сертификатов в формате, пригодном для экспорта в другие системы, необходимо:

1) в основном окне программы перейти во вкладку «Сертификаты»;
2) в списке выбрать корневой сертификат, нажать правой кнопкой мыши и в раскрывшемся меню выбрать «ЦС — Сгенерировать CRL»;
3) в открывшемся окне, при необходимости, уточнить параметры списка;
4) нажать кнопку **[Да]**.

Созданные списки отзыва можно просмотреть во вкладке «Списки отзыва сертификатов». Из этой же вкладки списки отозванных сертификатов можно экспортировать, нажав кнопку **[Экспорт]**, формат экспорта «PEM (`*.pem`)».

6.11. Средство удаленного администрирования Ansible

Ansible является программным решением для настройки и централизованного управления конфигурациями удаленных машин, в том числе одновременно группой машин. Для работы Ansible используется существующая инфраструктура SSH.

В Ansible для применения конфигурации на удаленной машине используется режим push mode, который заключается в распространении конфигурации с управляющей машины на удаленную.

6.11.1. Состав

В состав Ansible входят модули, обеспечивающие развертывание, контроль и управление компонентами удаленных машин. Перечень основных модулей приведен в таблице 30.

Таблица 30

Модуль	Описание
shell	Позволяет запускать shell-команды на удаленном узле, например: `ansible -i step-02/hosts -m shell -a 'uname -a' host0.example.org`
copy	Позволяет копировать файл из управляющей машины на удаленный узел: `ansible -i step-02/hosts -m copy -a 'src=<исходный_каталог> dest=<каталог_назначения>' host0.example.org`
setup	Предназначен для сбора фактических данных с узлов: `ansible -i step-02/hosts -m setup host0.example.org`

6.11.2. Установка и настройка Ansible

На управляющей и управляемых машинах должен быть установлен Python.

Дополнительно для работы Ansible необходимы следующие Python-модули на управляющей машине:

- python-yaml;
- paramiko;
- python-jinja2.

Установка модулей осуществляется путем выполнения команды:

```
apt install python-yaml python-jinja2 python-paramiko python-crypto
```

Для установки Ansible выполнить команду:

```
apt install ansible
```

Перечень машин, которыми нужно управлять, задается двумя способами:

- в текстовом файле (по умолчанию используется ini-файл) в каталоге /etc/ansible/hosts;
- с помощью сценария, получающего перечень машин из сторонних программных продуктов, например, от Zabbix.

Кроме списка управляемых машин в ini-файле может указываться дополнительная информация: номера портов для подключения по SSH, способ подключения, пароль для подключения, имя пользователя, объединения групп и т. п.

Примеры:

1. Конфигурационный ini-файл, в квадратных скобках указаны имена групп управляемых машин

```
[dbservers]
nude1.example.ru
nude2.example.ru

[webservers]
srv1.example.ru ansible_ssh_port=8877 ansible_ssh_host=192.168.1.1
srv2.example.ru
srv[3:20].example.ru
```

2. Конфигурационный YAML-файл

```
all:
hosts:
mail.example.ru:
children:
webservers:
hosts:
srv1.example.ru:
jumper:
ansible_port: 8877
ansible_host: 192.168.1.1
srv2.example.ru:
dbservers:
hosts:
nude1.example.ru:
nude2.example.ru:
```

В дополнение к конфигурационному файлу при определении и управлении группами удаленных машин используется переменные параметры. Переменные параметры могут быть объединены в группы. Данные о переменных предпочтительно хранить в отдельных YAML-файлах в соответствующих каталогах:

- `/etc/ansible/group_vars/<имя_группы>` — для переменных группы машин ;
- `/etc/ansible/host_vars/<имя_машины>` — для переменных отдельных машин.

6.11.3. Сценарии Ansible

Ansible позволяет использовать сценарии, предназначенные для выполнения на управляемых машинах. Сценарии пишутся на языке YAML.

Для выполнения сценария используется команда `ansible-playbook` со следующим синтаксисом:

```
ansible-playbook <имя_файла_сценария.yml> ... [другие параметры]
```

Описание основных параметров сценариев приведено в таблице 31.

Таблица 31

Параметр	Описание
`hosts`	Указываются управляемые узлы или группы узлов, к которым нужно применить изменения

Окончание таблицы 31

Параметр	Описание
`tasks`	Описывается состояние, в которое необходимо привести управляемый узел, альтернативой могут быть роли
`gather_facts`	Указывает собирать или нет информацию об узлах перед выполнением задач. Значение по умолчанию — «Да»
`vars`	Указываются переменные, которые будут использованы при выполнении сценария
`connection`	Используется для указания метода соединения с узлами: pure ssh, paramiko, fireball, chroot, jail, local, accelerate
`sudo`	После установления соединения выполнять задачу с привилегиями другого пользователя. Значение по умолчанию — `root`
`sudo_user`	В сочетании с параметром `sudo` можно указать пользователя, с привилегиями которого будет выполнена задача
`vars_prompt`	Перед выполнением сценария Ansible в интерактивном режиме может уточнить указанные в этом разделе параметры
`remote_user` (`user`)	Имя пользователя для авторизации на удаленном узле

7. СРЕДСТВА ОБЕСПЕЧЕНИЯ ОТКАЗОУСТОЙЧИВОСТИ И ВЫСОКОЙ ДОСТУПНОСТИ

7.1. Pacemaker и Corosync

В состав ОС входит набор программного обеспечения Pacemaker и Corosync, используемого для построения кластерных систем высокой доступности. Основные особенности Pacemaker и Corosync:

- обнаружение и восстановление после сбоев узлов и служб;
- независимость от подсистемы хранения — не требуется общее хранилище;
- независимость от типов ресурсов — все что может быть выполнено путем запуска сценария, может быть кластеризовано;
- поддержка кластеров любого размера;
- поддержка кворумных и ресурсозависимых кластеров;
- поддержка избыточной конфигурации;
- автоматическая репликация конфигурации, может быть обновлена с любого узла кластера;
- возможность задания порядка запуска ресурсов независимо от того, на каком узле они находятся;
- поддержка ресурсов, запускаемых на множестве узлов, — клонов;
- поддержка ресурсов с мульти-режимами работы (master/slave, primary/secondary).

С точки зрения кластера все используемые сущности: службы, точки монтирования, тома и разделы — это ресурсы, поэтому в данном руководстве под словом «ресурс» понимается все, что находится под управлением кластера.

7.1.1. Установка

Для установки Pacemaker и Corosync необходимо выполнить следующее:

1) на каждом сервере отказоустойчивого кластера установить пакеты `pacemaker` и `pcs`:

```
sudo apt install pacemaker pcs
```

2) на каждом сервере разрешить автозапуск Corosync. Для этого в конфигурационном файле `/etc/default/corosync` указать параметр:

```
START=yes
```

3) на каждом сервере следует произвести запуск необходимых служб `hacluster`:

```
sudo systemctl start corosync
sudo systemctl start pacemaker
sudo systemctl restart pacemaker
```

7.1.2. Пример настройки кластера

Настройка Pacemaker и Corosync на примере двух серверов с ОС: `server-1` и `server-2`. Оба сервера должны видеть друг друга по имени, для этого должен быть настроен DNS или в файле `/etc/hosts` содержаться соответствующие записи.

Для настройки необходимо выполнить следующие действия:

1) на каждом сервере настроить службу синхронизации времени в соответствии с 6.7;

2) на каждом сервере удалить возможно сохранившуюся предыдущую конфигурацию кластера:

```
sudo pcs cluster destroy
```

3) на каждом сервере установить одинаковый пароль (например, 12345678) для учетной записи администратора кластера `hacluster`, выполнив команду и введя пароль при соответствующих запросах:

```
sudo passwd hacluster
```

4) на первом (главном) сервере настроить авторизацию для обоих серверов, выполнив команду:

```
sudo pcs host auth server-1 server-2 -u hacluster -p 12345678
```

Результат выполнения команды:

```
server-2: Authorized
server-1: Authorized
```

5) создать и запустить кластер, последовательно выполнив на первом сервере команды:

```
sudo pcs cluster setup mycluster server-1 server-2 --force
sudo pcs cluster start --all
```

где `mycluster` — имя создаваемого кластера.
Результат выполнения команд:

```
No addresses specified for host 'server-1', using 'server-1'
No addresses specified for host 'server-2', using 'server-2'
Destroying cluster on hosts: 'server-1', 'server-2'...
server-1: Successfully destroyed cluster
server-2: Successfully destroyed cluster
Requesting remove 'pcsd settings' from 'server-1', 'server-2'
server-1: successful removal of the file 'pcsd settings'
server-2: successful removal of the file 'pcsd settings'
Sending 'corosync authkey', 'pacemaker authkey' to 'server-1', 'server-2'
server-1: successful distribution of the file 'corosync authkey'
```

```
server-1: successful distribution of the file 'pacemaker authkey'
server-2: successful distribution of the file 'corosync authkey'
server-2: successful distribution of the file 'pacemaker authkey'
Synchronizing pcsd SSL certificates on nodes 'server-1', 'server-2'...
server-1: Success
server-2: Success
Sending 'corosync.conf' to 'server-1', 'server-2'
server-1: successful distribution of the file 'corosync.conf'
server-2: successful distribution of the file 'corosync.conf'
Cluster has been successfully set up.

server-1: Starting Cluster...
server-2: Starting Cluster...
```

6) на обоих серверах перезапустить службу `pcsd`:

```
sudo systemctl restart pcsd
```

7) на первом сервере включить автозапуск кластера:

```
sudo pcs cluster enable --all
```

Результат выполнения команды:

```
server-1: Cluster Enabled
server-2: Cluster Enabled
```

8) для текущего кластера, состоящего из двух серверов, задать базовые настройки:
   а) отключить использование механизма `stonith`, отвечающего за изоляцию некорректно работающих серверов от основного кластера, выполнив команду:

   ```
   sudo pcs property set stonith-enabled=false
   ```

   ВНИМАНИЕ! Отключать использование `stonith` рекомендуется только при выполнении тестирования, в эксплуатируемых кластерах для предотвращения потери данных `stonith` должен быть включен. Включить механизм `stonith` можно будет после его настройки;

   б) отключить действия при потере кворума (т. к. кворум возможен в кластере из трех и более серверов), выполнив команду:

   ```
   sudo pcs property set no-quorum-policy=ignore
   ```

   в) для возможности запуска ресурсов на любом из серверов кластера включить симметричный кластер командой:

   ```
   sudo pcs property set symmetric-cluster=true
   ```

Если в кластере требуется ограничить запуск ресурсов на определенном сервере, то необходимо отключить использование симметричного кластера командой:

```
sudo pcs property set symmetric-cluster=false
```

При использовании несимметричного кластера необходимо для каждого ресурса указывать правила и приоритет запуска на серверах. Ресурсы, для которых не определены правила и серверы, не будут запускаться.

Для проверки статуса кластера выполнить команду:

```
sudo pcs status
```

Результат выполнения команды:

```
Cluster name: mycluster
Stack: corosync
Current DC: server-1 (version 2.0.1-9e909a5bdd) - partition with quorum
Last updated: Wed Jul 27 16:08:22 2022
Last change: Wed Jul 27 16:07:41 2022 by root via cibadmin on server-1

2 nodes configured
0 resources configured

Online: [ server-1 server-2 ]

No resources

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

Если вывод команды показывает, что второй сервер недоступен:

```
Online: [ server-1 ]
OFFLINE: [ server-2 ]
```

следует отредактировать конфигурационный файл `/etc/corosync/corosync.conf` на обоих серверах, заменив имена серверов в строках:

```
ring0_addr: server-1
ring0_addr: server-2
```

на их IP-адреса:

```
ring0_addr: <IP-адрес_server-1>
ring0_addr: <IP-адрес_server-2>
```

После редактирования конфигурационного файла следует перезапустить службу `corosync` на обоих серверах, выполнив команду:

```
sudo systemctl restart corosync
```

Для управления кластером Pacemaker используются инструменты командной строки `pcs` и `crm_mon`.

Управление кластером может также осуществляться через веб-интерфейс:

```
https://server-1:2224/
```

7.2. Keepalived

Keepalived используется в качестве управляющего ПО для организации мониторинга и обеспечения высокой доступности узлов и служб.

Демон Keepalived обеспечивает автоматический переход на резервный ресурс в режиме ожидания в случае возникновения ошибки или сбоя основного ресурса.

Для обеспечения автоматического перехода используется протокол VRRP (Virtual Redundancy Routing Protocol). Данный протокол позволяет использовать виртуальный IP-адрес VIP (virtual IP), который является плавающим (расшаренным) между узлами.

7.2.1. Установка

Пакет Keepalived необходимо установить на каждом узле, доступность которых требуется обеспечить, и на каждом резервном узле. Для установки выполнить следующую команду:

```
apt install keepalived
```

7.2.2. Пример настройки

Настройка Keepalived на примере двух серверов с ОС: `server-1` (основной) и `server-2` (резервный). На серверах должен быть настроен режим репликации для обеспечения

горячего резервирования. Также на обоих серверах должно быть два сетевых интерфейса. Одному из сетевых интерфейсов основного сервера присвоить VIP.

На каждом сервере в конфигурационный файл `/etc/sysctl.conf` добавить строку:

```
net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
```

и выполнить для проверки команду:

```
sysctl -p
```

На основном сервере откорректировать конфигурационный файл Keepalived `/etc/keepalived/keepalived.conf`, указав необходимые значения для основных параметров:

- `interface` — интерфейс подключения;
- `state` — статус сервера, для основного указывается значение `MASTER`;
- `virtual_router_id` — идентификатор виртуального маршрутизатора (должен быть одинаковым для обоих серверов);
- `priority` — приоритет основного сервера. Должен быть больше, чем резервного;
- `auth_type` — значение `PASS` задает парольную аутентификацию для серверов;
- `auth_pass` — общий пароль для всех узлов кластера;
- `virtual_ipaddress` — виртуальный IP-адрес.

Пример

Конфигурационный файл `/etc/keepalived/keepalived.conf` основного сервера

```
global_defs {
   notification_email {
       username@domain.ru
   }
   notification_email_from servers@domain.ru
       smtp_server 1.1.1.1
       smtp_connect_timeout 30
       router_id main
}

vrrp_instance server-1 {
   interface eth0
```

```
    state MASTER
    virtual_router_id 200
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass password
         }

    virtual_ipaddress {
        10.1.9.190/32 dev eth0
         }

}
```

Для применения настроек и запуска демона Keepalived выполнить команду:

```
systemctl start keepalived
```

Далее необходимо откорректировать конфигурационный файл Keepalived `/etc/keepalived/keepalived.conf` резервного сервера, указав необходимые значения для основных параметров:

- `interface` — интерфейс подключения;
- `state` — статус сервера, для резервного указывается значение `BACKUP`;
- `virtual_router_id` — идентификатор виртуального маршрутизатора (должен быть одинаковым для обоих серверов);
- `priority` — приоритет резервного сервера. Должен быть меньше, чем основного;
- `auth_type` — значение `PASS` задает парольную аутентификацию для серверов;
- `auth_pass` — общий пароль для всех узлов кластера;
- `virtual_ipaddress` — виртуальный IP-адрес.

Пример

Конфигурационный файл `/etc/keepalived/keepalived.conf` резервного сервера

```
global_defs {
    notification_email {
        username@domain.ru
    }
    notification_email_from servers@domain.ru
```

```
        smtp_server 1.1.1.1
        smtp_connect_timeout 30
        router_id reserve
}

vrrp_instance server-2 {
   interface eth0
   state BACKUP
   virtual_router_id 200
   priority 50
   advert_int 1
   authentication {
       auth_type PASS
       auth_pass password
 }

   virtual_ipaddress {
       10.4.1.190/32 dev eth0
 }
}
```

Для применения настроек и запуска демона Keepalived выполнить команду:

```
systemctl start keepalived
```

7.3. Распределенная файловая система Ceph

Распределенные файловые системы используются в высокоскоростных вычислениях и фокусируются на высокой доступности, производительности и масштабируемости. ОС поддерживает распределенную файловую систему Ceph.

Ceph — распределенная объектная система хранения, предоставляющая файловый и блочный интерфейсы доступа. Ceph представляет собой кластер узлов, выполняющих различные функции, обеспечивая хранение и репликацию данных, а также распределение нагрузки, что гарантирует высокую доступность и надежность. При добавлении или удалении новых узлов кластера массив хранимых данных автоматически балансируется с учетом внесенных изменений.

Схема программных компонентов Ceph представлена на рис. 1.

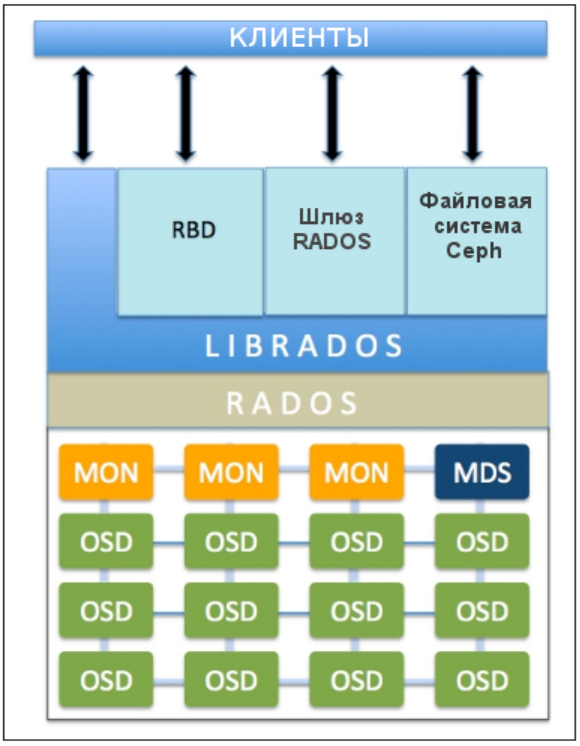

Рис. 1

Основой Ceph является служба RADOS (безотказное автономное распределенное хранилище объектов, Reliable Autonomic Distributed Object Store). В кластере Ceph данные хранятся в виде объектов, а служба RADOS обеспечивает хранение этих объектов независимо от их типа данных. Чтобы обеспечить высокую доступность и надежность системы хранения, служба RADOS осуществляет репликацию данных, обнаружение отказов и восстановление данных, а также миграцию данных и балансировку кластера при добавлении дополнительного устройства хранения или его извлечении.

Служба OSD (служба хранения объектов, Object Storage Daemon) обеспечивает хранение данных и обрабатывает запросы клиентов, обмениваясь данными с другими экземплярами службы OSD. Как правило, один экземпляр службы OSD связан с одним физическим устройством хранения.

Служба MON (монитор, monitor) отслеживает состояние всего кластера путем хранения карты состояния кластера, которая включает в себя карты состояния других программных компонентов Ceph. Для обеспечения высокой доступности и надежности кластера рекомендуется разворачивать три или пять экземпляров службы MON на различных узлах кластера. Главное, чтобы количество экземпляров было нечетным для обеспечения кворума. Если больше половины экземпляров службы MON будут недоступны, операция записи данных в кластер заблокируется для предотвращения рассогласованности данных.

Служба MGR (администратор, manager) отслеживает метрики времени выполнения различных команд и параметры состояния кластера Ceph, включая использование хранилища, текущие метрики производительности и нагрузку на систему. Служба MGR предоставляет

интерфейс взаимодействия для внешних систем управления и мониторинга. Для обеспечения высокой доступности и надежности кластера рекомендуется развернуть несколько экземпляров службы MGR на различных узлах кластера. Как правило, служба MGR разворачивается на тех же узлах кластера, на которых развернуты экземпляры службы MON. При этом в активном режиме функционирует только один экземпляр службы MGR, остальные экземпляры находятся в режиме ожидания.

Библиотека `librados` предоставляет интерфейс доступа к службе RADOS с поддержкой языков программирования PHP, Ruby, Python, C и C++.

Служба RBD (блочное устройство RADOS, RADOS block device) предоставляет пользователю возможность создавать и использовать виртуальные блочные устройства произвольного размера. Программный интерфейс RBD позволяет работать с этими устройствами в режиме чтения/записи и выполнять служебные операции — изменение размера, клонирование, создание и возврат к снимку состояния и т. д.

Шлюз RADOS позволяет использовать Ceph для хранения пользовательских объектов и предоставляет API, совместимый с Amazon S3 RESTful и OpenStack Swift.

Файловая система Ceph (CephFS) — POSIX-совместимая файловая система, использующая Ceph в качестве хранилища. Для того чтобы клиенты могли подключать Ceph как файловую систему, в кластере необходимо развернуть хотя бы один экземпляр службы MDS.

Служба MDS (сервер метаданных, MetaData Server) обеспечивает синхронное состояние файлов в точках монтирования CephFS. Служба MDS отслеживает метаданные файловой иерархии и сохраняет их только для CephFS. Использует активную копию и резервные, причем активная копия в пределах кластера только одна.

7.3.1. Развертывание Ceph

Пример развертывания распределенного хранилища на базе кластера Ceph из трех узлов `node1`, `node2` и `node3`. На узлах кластера будут развернуты службы MON и OSD. Кроме того, на узле `node1` будет запущена служба MGR.

ВНИМАНИЕ! Данная конфигурация предназначена только для ознакомления и тестирования Ceph. При развертывании кластера Ceph на объекте эксплуатации не рекомендуется размещать службы MON и OSD на одном узле.

В описываемой конфигурации в составе каждого из узлов кластера имеются два жестких диска: на дисках `sda` установлена ОС, диски `sdb` будут задействованы для хранения данных.

Как правило, в кластере Ceph используются две сети: одна используется для организации взаимодействия пользователей и служб Ceph, вторая — для репликации данных. В описы-

ваемой конфигурации для этих целей будет использоваться один сегмент сети. Сетевым интерфейсам узлов назначен фиксированный IP-адрес:

- 10.0.0.171 для узла `node1`;
- 10.0.0.172 для узла `node2`;
- 10.0.0.173 для узла `node3`.

В описываемой конфигурации на узлах кластера настроена служба синхронизации времени в соответствии с 6.7.

Для развертывания кластера Ceph необходимо выполнить следующие действия:

1) развернуть первый экземпляр службы MON (см. 7.3.1.1);
2) добавить необходимое количество экземпляров службы MON (см. 7.3.1.2);
3) развернуть необходимое количество экземпляров службы MGR (см. 7.3.1.3);
4) развернуть необходимое количество экземпляров службы OSD (см. 7.3.1.4).

7.3.1.1. Инициализация первого экземпляра службы MON

Для инициализации первого экземпляра службы MON на одном из узлов кластера необходимо выполнить следующие действия:

1) установить пакет ceph:

```
sudo apt install ceph
```

2) сгенерировать идентификатор кластера в формате UUID командой:

```
uuidgen
```

Пример вывода после выполнения команды:

```
f98c5e15-6736-41e9-966d-e38798029719
```

3) создать конфигурационный файл `/etc/ceph/<кластер>.conf` со следующими строками:

```
[global]
fsid = <идентификатор_кластера>
mon initial members = <узел_MON>
mon host = <адрес_узла_MON>
public network = <сеть_управления>
cluster network = <сеть_данных>
auth_allow_insecure_global_id_reclaim = false
osd_pool_default_pg_autoscale_mode = off
```

где <кластер> — условное наименование кластера. По умолчанию используется наименование ceph;

<идентификатор_кластера> — идентификатор в формате UUID;

<узел_MON> — сетевое имя узла, на котором будет развернут первый экземпляр службы MON;

<адрес_узла_MON> — IP-адрес узла, на котором будет развернут первый экземпляр службы MON;

<сеть_управления> — параметры сети, которая используется для взаимодействия пользователей и служб Ceph;

<сеть_данных> — параметры сети, которая используется для репликации данных. В описываемой конфигурации будет использоваться тот же сегмент, что и для сети управления.

В параметре auth_allow_insecure_global_id_reclaim заданное значение false устанавливает запрет на подключение к кластеру тех клиентов, которые не могут безопасным образом восстановить свой идентификатор.

В параметре osd_pool_default_pg_autoscale_mode заданное значение false устанавливает запрет на автоматическое изменение количества групп размещения в пуле.

В описываемой конфигурации файл /etc/ceph/ceph.conf имеет следующие строки:

```
[global]
fsid = f98c5e15-6736-41e9-966d-e38798029719
mon initial members = node1
mon host = 10.0.0.171
public network = 10.0.0.1/24
cluster network = 10.0.0.1/24
auth_allow_insecure_global_id_reclaim = false
osd_pool_default_pg_autoscale_mode = off
```

4) сгенерировать ключ для службы MON:

```
sudo ceph-authtool --create-keyring /tmp/ceph.mon.keyring --gen-key \
     -n mon. --cap mon 'allow *'
```

5) сгенерировать ключ администратора кластера:

```
sudo ceph-authtool --create-keyring /etc/ceph/ceph.client.admin.keyring \
     --gen-key -n client.admin --cap mon 'allow *' --cap osd 'allow *' \
     --cap mds 'allow *' --cap mgr 'allow *'
```

6) добавить ключ администратора кластера в файл с ключом службы MON:

```
sudo ceph-authtool /tmp/ceph.mon.keyring --import-keyring \
     /etc/ceph/ceph.client.admin.keyring
```

7) сгенерировать ключ для службы OSD:

```
sudo ceph-authtool --create-keyring \
     /var/lib/ceph/bootstrap-osd/ceph.keyring --gen-key -n \
     client.bootstrap-osd --cap mon 'profile bootstrap-osd' \
     --cap mgr 'allow r'
```

8) добавить ключ службы OSD в файл с ключом службы MON:

```
sudo ceph-authtool /tmp/ceph.mon.keyring --import-keyring \
     /var/lib/ceph/bootstrap-osd/ceph.keyring
```

9) сформировать карту состояния кластера командой:

```
monmaptool --create --add <узел_MON> <адрес_узла_MON> \
     --fsid <идентификатор_кластера> /tmp/monmap
```

В описываемой конфигурации команда имеет вид:

```
monmaptool --create --add node1 10.0.0.171 \
     --fsid f98c5e15-6736-41e9-966d-e38798029719 /tmp/monmap
```

10) создать рабочий каталог для первого экземпляра службы MON командой:

```
sudo mkdir /var/lib/ceph/mon/<кластер>-<узел_MON>
```

В описываемой конфигурации команда имеет вид:

```
sudo mkdir /var/lib/ceph/mon/ceph-node1
```

11) инициализировать рабочий каталог службы MON, указав карту состояния кластера и ключевой файл:

```
sudo ceph-mon --mkfs -i <узел_MON> ----monmap /tmp/monmap \
     --keyring /tmp/ceph.mon.keyring
```

В описываемой конфигурации команда имеет вид:

```
sudo ceph-mon --mkfs -i node1 --monmap /tmp/monmap --keyring \
     /tmp/ceph.mon.keyring
```

12) системного пользователя ceph установить владельцем рабочего каталога службы MON:

```
sudo chown -R ceph:ceph /var/lib/ceph/mon
```

13) запустить службу MON в качестве системной службы, для этого последовательно выполнить следующие команды:

```
sudo systemctl enable ceph-mon.target
sudo systemctl enable ceph-mon@<узел>
sudo systemctl start ceph-mon@<узел>
```

Пример команд для описываемой конфигурации:

```
sudo systemctl enable ceph-mon.target
sudo systemctl enable ceph-mon@node1
sudo systemctl start ceph-mon@node1
```

14) включить использование второй версии протокола сетевого обмена между экземплярами службы MON:

```
sudo ceph mon enable-msgr2
```

15) вывести информацию о кластере Ceph:

```
sudo ceph -s
```

Пример вывода после выполнения команды:

```
cluster:
id:     f98c5e15-6736-41e9-966d-e38798029719
health: HEALTH_OK

services:
mon: 1 daemons, quorum node1 (age 39s)
mgr: no daemons active
osd: 0 osds: 0 up, 0 in

data:
pools:   0 pools, 0 pgs
objects: 0 objects, 0 B
usage:   0 B used, 0 B / 0 B avail
pgs:
```

Сообщение вида:

```
health: HEALTH_OK
```

указывает на корректность выполненных настроек.

7.3.1.2. Добавление нового экземпляра службы MON

Дополнительные экземпляры службы MON разворачиваются на отдельных узлах кластера Ceph. В описываемой конфигурации экземпляры службы MON будут развернуты на узлах `node2` и `node3`.

Для развертывания нового экземпляра службы MON необходимо выполнить следующие действия:

1) на узле, на котором развернут первый экземпляр службы MON, в конфигурационный файл /etc/ceph/ceph.conf добавить информацию об узлах, на которых будут развернуты дополнительные экземпляры службы MON:

```
mon initial members = <узел_1>, <узел_2> ...  <узел_N>
mon host = <IP-адрес_узла_1>, <IP-адрес_узла_2> ... <IP-адрес_узла_N>
```

Для описываемой конфигурации в файле /etc/ceph/ceph.conf необходимо указать следующую информацию об узлах:

```
...
mon initial members = node1,node2,node3
mon host = 10.0.0.171,10.0.0.172,10.0.0.173
...
```

2) на дополнительных узлах установить пакет ceph-mon:

```
sudo apt install ceph-mon
```

3) на дополнительных узлах получить копии конфигурационного файла /etc/ceph/ceph.conf и ключа администратора кластера /etc/ceph/ceph.client.admin.keyring. Для этого можно воспользоваться следующими командами:

```
ssh <администратор>@<первый_MON> "cat /etc/ceph/ceph.conf" \
    | sudo tee /etc/ceph/ceph.conf
ssh <администратор>@<первый_MON> "sudo -S cat \
     /etc/ceph/ceph.client.admin.keyring" \ | sudo tee \
     /etc/ceph/ceph.client.admin.keyring
```

где <администратор> — локальный администратор узла, на котором развернут первый экземпляр службы MON;

<первый_MON> — IP-адрес узла, на котором развернут первый экземпляр службы MON.

Пример команд для описываемой конфигурации:

```
ssh astra@10.0.0.171 "cat /etc/ceph/ceph.conf" \
    | sudo tee /etc/ceph/ceph.conf
ssh astra@10.0.0.171 "sudo -S cat /etc/ceph/ceph.client.admin.keyring" \
    | sudo tee /etc/ceph/ceph.client.admin.keyring
```

4) на дополнительных узлах получить карту состояния кластера:

```
sudo ceph mon getmap -o /tmp/ceph.map
```

5) на дополнительных узлах получить ключ службы MON:

```
sudo ceph auth get mon. -o /tmp/ceph.mon.keyring
```

6) на дополнительных узлах инициализировать рабочий каталог службы MON, указав карту состояния кластера и ключевой файл:

```
sudo ceph-mon -i <узел> --mkfs --monmap /tmp/ceph.map \
     --keyring /tmp/ceph.mon.keyring
```

Пример команд для описываемой конфигурации:

а) на узле node2:

```
sudo ceph-mon -i node2 --mkfs --monmap /tmp/ceph.map  \
     --keyring /tmp/ceph.mon.keyring
```

б) на узле node3:

```
sudo ceph-mon -i node3 --mkfs --monmap /tmp/ceph.map  \
     --keyring /tmp/ceph.mon.keyring
```

7) на дополнительных узлах для каталога /var/lib/ceph/mon/ceph-<узел> установить владельцем системного пользователя ceph. Пример команд для описываемой конфигурации:

а) на узле node2:

```
sudo chown -R ceph:ceph /var/lib/ceph/mon/ceph-node2
```

б) на узле node3:

```
sudo chown -R ceph:ceph /var/lib/ceph/mon/ceph-node3
```

8)) на дополнительных узлах запустить службу MON в качестве системной службы, для этого последовательно выполнить следующие команды:

```
sudo systemctl enable ceph-mon.target
sudo systemctl enable ceph-mon@<узел>
sudo systemctl start ceph-mon@<узел>
```

Пример команд для описываемой конфигурации:

а) на узле node2:

```
sudo systemctl enable ceph-mon.target
sudo systemctl enable ceph-mon@node2
sudo systemctl start ceph-mon@node2
```

на узле node3:

```
sudo systemctl enable ceph-mon.target
sudo systemctl enable ceph-mon@node3
sudo systemctl start ceph-mon@node3
```

9) на одном из узлов вывести информацию о кластере Ceph:

```
sudo ceph -s
```

В случае успешного развертывания дополнительных экземпляров службы MON в терминале отобразится информация о количестве функционирующих экземпляров. Пример вывода после выполнения команды для описываемой конфигурации:

```
...
services:
mon: 3 daemons, quorum node1,node3,node2 (age 44s)
...
```

7.3.1.3. Добавление экземпляра службы MGR

Как правило, служба MGR разворачивается на тех же узлах кластера, на которых развернуты экземпляры службы MON. В описываемой конфигурации экземпляр службы MGR будет развернут на узле `node1`.

Для развертывания экземпляра службы MGR на узле кластера необходимо выполнить следующие действия:

1) установить пакет `ceph-mgr`:

```
sudo apt install ceph-mgr
```

Для описываемой конфигурации указанная команда не выполняется, так как пакет `ceph-mgr` устанавливается автоматически при установке пакета `ceph`;

2) создать рабочий каталог для экземпляра службы MGR командой:

```
sudo mkdir /var/lib/ceph/mgr/<кластер>-<узел>
```

В описываемой конфигурации команда имеет вид:

```
sudo mkdir /var/lib/ceph/mgr/ceph-node1
```

3) сгенерировать ключ для службы MGR:

```
sudo ceph auth get-or-create mgr.‘hostname -s‘ mon ’allow profile mgr’ \
     osd ’allow *’ mds ’allow *’ \
     -o /var/lib/ceph/mgr/<кластер>-<узел>/keyring
```

В описываемой конфигурации команда имеет вид:

```
sudo ceph auth get-or-create mgr.‘hostname -s‘ mon ’allow profile mgr’ \
     osd ’allow *’ mds ’allow *’ \
     -o /var/lib/ceph/mgr/ceph-node1/keyring
```

4) для рабочего каталога службы MGR установить владельцем системного пользователя ceph:

```
sudo chown -R ceph:ceph /var/lib/ceph/mgr
```

5) запустить экземпляр службы MGR в качестве системной службы, для этого последовательно выполнить следующие команды:

```
sudo systemctl enable ceph-mgr.target
sudo systemctl enable ceph-mgr@<узел>
sudo systemctl start ceph-mgr@<узел>
```

Пример команд для описываемой конфигурации:

```
sudo systemctl enable ceph-mgr.target
sudo systemctl enable ceph-mgr@node1
sudo systemctl start ceph-mgr@node1
```

Запуск экземпляра службы MGR может занять длительное время. Для просмотра текущего статуса службы можно воспользоваться командой:

```
sudo systemctl status ceph-mgr@<узел>
```

6) вывести информацию о кластере Ceph:

```
sudo ceph -s
```

В случае успешного развертывания экземпляра службы MGR в терминале отобразится информация о количестве функционирующих экземпляров. Пример вывода после выполнения команды для описываемой конфигурации:

```
...
services:
...
mgr: node1(active, since 7s)
...
```

7.3.1.4. Добавление экземпляра службы OSD

Экземпляры службы OSD разворачиваются на отдельных узлах кластера Ceph. Как правило, один экземпляр службы OSD связан с одним физическим диском кластера. В описываемой конфигурации экземпляры службы OSD будут развернуты на узлах `node1`, `node2` и `node3`.

ВНИМАНИЕ! Данная конфигурация предназначена только для ознакомления и тестирования Ceph. При развертывании кластера Ceph на объекте эксплуатации не рекомендуется размещать службы MON и OSD на одном узле.

Для развертывания экземпляра службы OSD необходимо выполнить следующие действия:

1) на дополнительных узлах (в описываемой конфигурации это `node2` и `node3` установить пакет `ceph-osd`:

```
sudo apt install ceph-osd
```

2) на дополнительных узлах получить минимально необходимую конфигурацию кластера:

```
ssh <администратор>@<первый_MON> \
   "sudo -S ceph config generate-minimal-conf" \
   | sudo tee /etc/ceph/ceph.conf
```

где <администратор> — локальный администратор узла, на котором развернут первый экземпляр службы MON;
<первый_MON> — IP-адрес узла, на котором развернут первый экземпляр службы MON.

В описываемой конфигурации этот шаг не выполняется, так как на узлах `node2` и `node3` уже имеется конфигурационный файл `/etc/ceph/ceph.conf`;

3) на дополнительных узлах получить копию ключа службы OSD:

```
ssh <администратор>@<первый_MON> "sudo -S cat \
     /var/lib/ceph/bootstrap-osd/ceph.keyring" \
     | sudo tee /var/lib/ceph/bootstrap-osd/ceph.keyring
```

где <администратор> — локальный администратор узла, на котором развернут первый экземпляр службы MON;
<первый_MON> — IP-адрес узла, на котором развернут первый экземпляр службы MON.

В описываемой конфигурации команда имеет вид:

```
ssh astra@10.0.0.171 "sudo -S cat \
     /var/lib/ceph/bootstrap-osd/ceph.keyring" \
     | sudo tee /var/lib/ceph/bootstrap-osd/ceph.keyring
```

4) на всех узлах инициализировать экземпляр службы OSD командой:

```
sudo ceph-volume lvm create --data <диск>
```

В описываемой конфигурации команда имеет вид:

```
sudo ceph-volume lvm create --data /dev/sdb
```

Пример вывода после успешного выполнения команды:

```
...
--> ceph-volume lvm activate successful for osd ID: 0
--> ceph-volume lvm create successful for: /dev/sdb
```

5) на одном из узлов с развернутой службой MON вывести информацию о кластере Ceph:

```
sudo ceph -s
```

В случае успешного развертывания экземпляров службы OSD в терминале отобразится информация о количестве функционирующих экземпляров. Пример вывода после выполнения команды для описываемой конфигурации:

```
...
services:
...
osd: 3 osds: 3 up (since 43s), 3 in (since 57s)
...
```

7.3.2. Использование кластера Ceph

Ceph представляет для клиента различные варианты доступа к данным:

1) файловая система Ceph (CephFS);
2) программный интерфейс RBD;
3) шлюз RADOS.

CephFS предоставляет возможность монтировать один и тот же каталог с данными на чтение и запись множеству клиентов.

Для того чтобы клиенты могли использовать кластер Ceph как файловую систему, необходимо выполнить следующие действия:

1) инициализировать файловую систему (см. 7.3.2.1);
2) развернуть хотя бы один экземпляр службы MDS (см. 7.3.2.2);
3) настроить разделяемый ресурс (см. 7.3.2.3);
4) настроить подключение клиента к разделяемому ресурсу (см. 7.3.2.4).

7.3.2.1. Инициализация CephFS

Чтобы инициализировать файловую систему, на узле с развернутой службой MON необходимо выполнить команду:

```
sudo ceph fs volume create <наименование_фс>
```

В описываемой конфигурации команда имеет вид:

```
sudo ceph fs volume create testcephfs
```

Примечание. В процессе выполнения команды будут автоматически созданы два пула ресурсов хранения: один для размещения метаданных и второй для размещения пользовательских данных.

Чтобы вывести информацию о файловых системах, имеющихся в кластере, на узле с развернутой службой MON необходимо выполнить команду:

```
sudo ceph fs ls
```

Пример вывода после выполнения команды:

```
name: testcephfs, metadata pool: cephfs.testcephfs.meta, data pools:
[cephfs.testcephfs.data]
```

7.3.2.2. Добавление экземпляра службы MDS

В описываемой конфигурации экземпляр службы MDS будет развернут на узле `node1`.

Для развертывания экземпляра службы MDS на узле кластера необходимо выполнить следующие действия:

1) создать рабочий каталог для экземпляра службы MDS командой:

```
sudo mkdir -p /var/lib/ceph/mds/<кластер>-<узел>
```

В описываемой конфигурации команда имеет вид:

```
sudo mkdir -p /var/lib/ceph/mds/ceph-node1
```

2) сгенерировать ключ для службы MDS и разместить его в рабочем каталоге экземпляра службы MDS:

```
sudo ceph auth get-or-create mds.<узел> mon 'profile mds' \
     mgr 'profile mds' mds 'allow *' osd 'allow *' \
     -o /var/lib/ceph/mds/<кластер>-<узел>/keyring
```

В описываемой конфигурации команда имеет вид:

```
sudo ceph auth get-or-create mds.node1 mon 'profile mds' \
     mgr 'profile mds' mds 'allow *' osd 'allow *' \
     -o /var/lib/ceph/mds/ceph-node1/keyring
```

3) для рабочего каталога службы MDS установить владельцем системного пользователя `ceph`:

```
sudo chown -R ceph:ceph /var/lib/ceph/mds
```

4) запустить экземпляр службы MDS в качестве системной службы, для этого последовательно выполнить следующие команды:

```
sudo systemctl enable ceph-mds.target
sudo systemctl enable ceph-mds@<узел>
sudo systemctl start ceph-mds@<узел>
```

Пример команд для описываемой конфигурации:

```
sudo systemctl enable ceph-mds.target
sudo systemctl enable ceph-mds@node1
sudo systemctl start ceph-mds@node1
```

5) вывести информацию о кластере Ceph:

```
sudo ceph -s
```

В случае успешного развертывания экземпляра службы MDS в терминале отобразится информация о количестве функционирующих экземпляров.
Пример вывода после выполнения команды для описываемой конфигурации:

```
...
services:
...
mds: 1/1 daemons up
...
```

Примечание. Пока в кластере Ceph не будет создана файловая система служба MDS находится в неактивном режиме. В связи с этим информация об экземплярах службы MDS не отображается в выводе после выполнения команды:

```
sudo ceph -s
```

7.3.2.3. Подготовка разделяемого ресурса

Чтобы создать разделяемый ресурс, к которому будут подключаться клиенты, на узле с развернутой службой MON необходимо выполнить команду:

```
sudo ceph fs subvolume create <наименование_фс> <ресурс>
```

В описываемой конфигурации команда имеет вид:

```
sudo ceph fs subvolume create testcephfs data1
```

Чтобы настроить доступ к разделяемому ресурсу, на узле с развернутой службой MON необходимо выполнить команду:

```
sudo ceph fs subvolume authorize <наименование_фс> <ресурс> \
     <имя_пользователя> --access-level=<права_доступа>
```

В описываемой конфигурации команда имеет вид:

```
sudo ceph fs subvolume authorize testcephfs data1 user1 --access-level=rw
```

Примечание. Создавать учетные записи предварительно не требуется. В процессе выполнения команды будет автоматически создана учетная запись пользователя, которой будут присвоены указанные права доступа к разделяемому ресурсу.

7.3.2.4. Настройка подключения клиента к разделяемому ресурсу

Для подключения к разделяемому ресурсу необходимы следующие сведения:

- полный путь к разделяемому ресурсу;
- ключ пользователя, которому предоставлен доступ к разделяемому ресурсу.

Чтобы получить указанные сведения, на узле с развернутой службой MON необходимо выполнить команду:

```
sudo ceph auth ls | grep <имя_пользователя> -A2
```

Для описываемой конфигурации команда имеет вид:

```
sudo ceph auth ls | grep user1 -A2
```

Пример вывода после выполнения команды:

```
client.user1
key: AQD9D2hm+0psJxAArn5iVMhDewigF/E6r+dlCg==
caps: [mds] allow rw
path=/volumes/_nogroup/data1/1de2e9d3-fed9-47bc-824a-c6e5ab5512e3
```

Настройка подключения разделяемого ресурса на узле кластера

Для подключения разделяемого ресурса на узле кластера Ceph необходимо выполнить следующие действия:

1) получить ключ пользователя, которому предоставлен доступ к разделяемому ресурсу командой:

```
ssh <администратор>@<первый_MON> "sudo -S ceph auth get-or-create \
    client.user1" | sudo tee /etc/ceph/ceph.client.user1.keyring
```

где <администратор> — локальный администратор узла, на котором развернут первый экземпляр службы MON;

<первый_MON> — IP-адрес узла, на котором развернут первый экземпляр службы MON.

Для описываемой конфигурации команда имеет вид:

```
ssh astra@10.0.0.171 "sudo -S ceph auth get-or-create client.user1" \
    | sudo tee /etc/ceph/ceph.client.user1.keyring
```

2) подключить разделяемый ресурс командой:

```
sudo mount.ceph <имя_пользователя>@.<наименование_фс>=<ресурс> \
    <локальный_каталог>
```

где <наименование_фс> — наименование инициализированной файловой системы (см. 7.3.2.1);

ВНИМАНИЕ! Обязательно должен присутствовать символ точки (« . ») перед наименованием файловой системы;

<имя_пользователя> — имя пользователя, которому предоставлен доступ к разделяемому ресурсу;

<ресурс> — полный путь к разделяемому ресурсу;

<локальный_каталог> — локальный каталог, в который необходимо смонтировать разделяемый ресурс.

Для описываемой конфигурации команда имеет вид:

```
sudo mount.ceph user1@.testcephfs=\
    /volumes/_nogroup/data1/4025b53e-8df1-49b1-adee-365e0eeafc6d /mnt/
```

Настройка подключения разделяемого ресурса на внешнем компьютере с использованием инструментов Ceph

Для подключения разделяемого ресурса на компьютере, не входящем в кластер Ceph, необходимо выполнить следующие действия:

1) установить пакет ceph-common:

```
sudo apt install ceph-common
```

2) получить минимально необходимую конфигурацию кластера:

```
ssh <администратор>@<первый_MON> "sudo -S ceph config \
     generate-minimal-conf" | sudo tee /etc/ceph/ceph.conf
```

где <администратор> — локальный администратор узла, на котором развернут первый экземпляр службы MON;

<первый_MON> — IP-адрес узла, на котором развернут первый экземпляр службы MON.

Для описываемой конфигурации команда имеет вид:

```
ssh astra@10.0.0.171 "sudo -S ceph config generate-minimal-conf" \
     | sudo tee /etc/ceph/ceph.conf
```

3) получить ключ пользователя, которому предоставлен доступ к разделяемому ресурсу командой:

```
ssh <администратор>@<первый_MON> "sudo -S ceph auth get-or-create \
     client.user1" | sudo tee /etc/ceph/ceph.client.user1.keyring
```

где <администратор> — локальный администратор узла, на котором развернут первый экземпляр службы MON;

<первый_MON> — IP-адрес узла, на котором развернут первый экземпляр службы MON.

Для описываемой конфигурации команда имеет вид:

```
ssh astra@10.0.0.171 "sudo -S ceph auth get-or-create client.user1" \
     | sudo tee /etc/ceph/ceph.client.user1.keyring
```

4) подключить разделяемый ресурс командой:

```
sudo mount.ceph <имя_пользователя>@.<наименование_фс>=<ресурс> \
     <локальный_каталог>
```

где <наименование_фс> — наименование инициализированной файловой системы (см. 7.3.2.1);

ВНИМАНИЕ! Обязательно должен присутствовать символ точки (« . ») перед наименованием файловой системы;

<имя_пользователя> — имя пользователя, которому предоставлен доступ к разделяемому ресурсу;

<ресурс> — полный путь к разделяемому ресурсу;

<локальный_каталог> — локальный каталог, в который необходимо смонтировать разделяемый ресурс.

Для описываемой конфигурации команда имеет вид:

```
sudo mount.ceph user1@.testcephfs=\
    /volumes/_nogroup/data1/4025b53e-8df1-49b1-adee-365e0eeafc6d /mnt/
```

Настройка подключения разделяемого ресурса на внешнем компьютере без использования инструментов Ceph

Если на компьютере, не входящем в кластер Ceph, нет возможности установить пакет ceph-common, то подключить разделяемый ресурс можно следующей командой:

```
sudo mount -t ceph <узлы_MON>:<ресурс> <локальный_каталог> \
     -o name=<имя_пользователя>,secret=<ключ>
```

где <узлы_MON> — IP-адреса узлов кластера, на которых развернуты экземпляры службы MON;

<ресурс> — полный путь к разделяемому ресурсу;

<локальный_каталог> — локальный каталог, в который необходимо смонтировать разделяемый ресурс;

<имя_пользователя> — имя пользователя, которому предоставлен доступ к разделяемому ресурсу;

<ключ> — ключ пользователя, которому предоставлен доступ к разделяемому ресурсу.

Для описываемой конфигурации команда имеет вид:

```
sudo mount -t ceph 10.0.0.171,10.0.0.172,10.0.0.173:\
     /volumes/_nogroup/data1/1de2e9d3-fed9-47bc-824a-c6e5ab5512e3 \
     /mnt/ -o name=user1,secret='AQD9D2hm+0psJxAArn5iVMhDewigF/E6r+dlCg=='
```

Сообщение об ошибке вида:

```
2024-06-25T08:21:48.660+0300 76ae9321bfc0 -1 auth: unable to find a keyring on
/etc/ceph/ceph.client.user1.keyring, /etc/ceph/ceph.keyring,/etc/ceph/keyring,
/etc/ceph/keyring.bin: (2) No such file or directory
```

можно игнорировать, такое сообщение появляется в случае, если в ОС установлено программное обеспечение Ceph. В таком случае производится попытка подключить разделяемый ресурс в первую очередь с использованием инструментов Ceph. Затем, в случае неудачи, производится попытка подключить разделяемый ресурс с использованием аутентификационных параметров, указанных в команде в качестве аргументов.

7.4. Средство эффективного масштабирования HAProxy

Для эффективного масштабирования используется программное средство HAProxy. HAProxy обеспечивает высокую доступность, отказоустойчивость и распределение нагрузки для TCP- и HTTP-приложений посредством распределения входящих запросов на несколько обслуживающих серверов.

HAProxy предоставляет следующие возможности:

- периодическая проверка доступности обслуживающих серверов, на которые перенаправляются запросы пользователей;
- несколько алгоритмов определения доступности сервера: tcp-check, http-check, mysql-check;
- распределение HTTP/HTTPS/TCP-запросов между доступными серверами;

- возможность закрепления определенных клиентов за конкретными обслуживающими серверам (stick-tables);
- поддержка IPv6 и UNIX sockets, HTTP/1.1 сжатия (deflate, gzip, libslz), SSL, полная поддержка постоянного HTTP-соединения;
- поддержка переменных блоков и Lua-сценариев в конфигурации сервера;
- веб-интерфейс с актуальным состоянием и статистикой работы программы.

7.4.1. Установка

На основном сервере, который будет принимать запросы и распределять их, необходимо установить пакет HAProxy:

```
apt install haproxy
```

7.4.2. Настройка

Настройка выполняется в конфигурационном файле `/etc/haproxy/haproxy.cfg`, включающем следующие разделы:

- `global` — определяет общую конфигурацию для всего HAProxy;
- `defaults` — является обязательным и определяет настройки по умолчанию для остальных разделов;
- `frontend` — используется для описания набора интерфейсов для принятия соединений от клиентов, а также правил распределения нагрузки;
- `backend` — используется для описания набора серверов, к которым будет выполняться подключение переадресованных входящих соединений, а также определения алгоритма распределения нагрузки;
- `listen` — объединенный раздел для описания `frontend` и `backend`. Используется для описания прокси-сервера в одном разделе, как правило, только для TCP-трафика.

В таблице 32 представлены основные примеры значений параметров конфигурационного файла и их описание.

Таблица 32

Раздел	Параметр	Описание
`global`	`log <address> <facility> [max level [min level]]` Например, `log 127.0.0.1 local0 notice`	Добавляет сервер системного журнала. `<facility>` — должен быть одним из 24 стандартных типов регистрации событий: `kern user mail daemon auth syslog lpr news uucp cron auth2 ftp ntp audit alert cron2 local0 local1 local2 local3 local4 local5 local6 local7`

Продолжение таблицы 32

Раздел	Параметр	Описание
	`maxconn <number>` Например, `maxconn 10000`	Устанавливает максимальное число одновременных подключений для каждого процесса `haproxy`
	`nbproc <number>` Например, `nbproc 2`	Задает количество процессов `haproxy`. По умолчанию создается только один процесс `haproxy`
	`daemon`	Устанавливает процессу `haproxy` режим работы «daemon»
	`user`	Пользователь, от имени которого работает процесс `haproxy`
	`group`	Группа, от имени которой работает процесс `haproxy`
	`chroot /var/lib/haproxy`	Устанавливает окружение процесса `haproxy`
`defaults`	`log global`	Включает в регистрацию событий информацию о трафике
	`mode http`	Режим работы HAProxy. Возможны два режима: - `http` — выполняется анализ Layer 7, подходит для распределения http-трафика; - `tcp` — распределение любого трафика
	`option dontlognull`	Отключает регистрацию пустых подключений
	`retries 3`	Количество попыток определить состояние обслуживающего сервера после сбоя подключения
	`option redispatch`	Распределяет запросы после сбоя подключения к одному из обслуживающих серверов
	`option httpclose`	Закрывает пассивные соединения
	`option forwardfor`	Включает X-Forwarded-For для передачи IP-адреса клиента обслуживающему серверу
`frontend`	`frontend http`	Задает имя `frontend`
	`bind *:80`	Задает IP-адрес и порт для прослушивания запросов
`backend`	`backend sitecluster`	Задает имя обслуживающего сервера

Продолжение таблицы 32

Раздел	Параметр	Описание
	`balance (roundrobin/leastconn/ static-rr/uri/source)`	Настройка алгоритма распределения. Поддерживаются следующие алгоритмы: - `Round Robin` — направляет новые подключения к следующему серверу в циклическом списке, который видоизменяется при помощи веса сервера, на основании которого идет распределение запросов. Вес сервера можно изменить «на лету». Параметр включается при помощи команды `balance roundrobin`; - `Least Connected` — направляет новые подключения к серверу с наименьшим числом соединений. Параметр включается при помощи команды `balance leastconn`; - `Static Round Robin` — направляет новые подключения к следующему серверу в циклическом списке, который видоизменяется при помощи веса сервера, на основании которого идет распределение запросов. В отличие от стандартной реализации `Round Robin`, в данном алгоритме нельзя изменить вес сервера «на лету». Изменение веса сервера требует перезагрузки HAProxy. Параметр включается при помощи команды `balance static-rr`; - `Source` — выбирает сервер исходя из хеша, построенного на основе IP-адреса пользователя. Таким образом, пользователь всегда обращается к одному и тому же серверу
	`server srv-1.3.my.com 21.86.21.20:80 cookie site113ha check inter 2000 fall 3 minconn 30 maxconn 70 weight 100`	Описание обслуживающего сервера, где: - `srv-1.3.my.com` — имя сервера; - `21.86.21.20:80` — IP-адрес: порт; - `cookie site113ha` — задание cookie, необходимого для правильного распределения сессий клиентов; - `check inter 2000 fall 3` — проверка доступности сервера каждые 2 с, при наличии трех ошибок считать сервер недоступным; - `minconn 30 maxconn 70` — организация очереди запросов, ограничение не более 70 одновременно обрабатывающихся запросов; - `weight 100` — вес сервера, возможные значения от 1 до 100
	`stats enable`	Включает статистику

Окончание таблицы 32

Раздел	Параметр	Описание
	`fullconn 200`	Задает максимальное значение одновременных подключений
`listen`	`listen stats-srv-3.my.com *:8180`	Описывает IP-адрес и порт доступа к статистике
	`stats uri /stats`	URL доступа к статистике
	`stats realm Haproxy Statistics`	Заголовок (title) страницы статистики
	`stats show-legends`	Отображает в статистике дополнительную информацию о параметрах
	`stats refresh 5s`	Указывает интервал автоматического обновления страницы статистики
	`stats auth test:test`	Устанавливает логин и пароль доступа к странице статистики

Пример

Конфигурационный файл для распределения нагрузки сервера Apache

```
global
     log /dev/log local0
     log /dev/log local1 notice
     maxconn 40000
     chroot /var/lib/haproxy
     stats socket /run/haproxy/admin.sock mode 660 level admin
     stats timeout 30s
     user haproxy
     group haproxy
     daemon        # Размещение сертификатов SSL
     ca-base /etc/ssl/certs
     crt-base /etc/ssl/private    # Алгоритмы защитного преобразования,
     # применяемые для SSL-подключений
     # Подробнее см. по ссылке:
     # https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
     ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:
          ECDH+AES128:DH+AES:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+
          3DES:!aNULL:!MD5:!DSS
     ssl-default-bind-options no-sslv3

defaults
     log global
     mode http
```

```
    option httplog
    option dontlognull
    retries 3
    option redispatch
    maxconn 2000
    timeout connect 5000
    timeout client 50000
    timeout server 50000
    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 408 /etc/haproxy/errors/408.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http

frontend localnodes
    bind *:80
    mode http
    default_backend nodes

backend nodes
    mode http
    balance roundrobin
    server webserver1 192.168.13.150:80 cookie serv1 check
    server webserver2 192.168.13.151:80 cookie serv2 check
```

8. СРЕДСТВА ОРГАНИЗАЦИИ ЕПП

8.1. Архитектура ЕПП

Единое пространство пользователей представляет собой средства организации работы пользователя в сети компьютеров, работающих под управлением ОС. В основу положен доменный принцип построения сети, подразумевающий объединение в одну сеть логически связанных компьютеров, например, принадлежащих одной организации. При этом пользователь получает возможность работы с сетевыми ресурсами сети и взаимодействия с другими пользователями.

Организация ЕПП обеспечивает:

- сквозную аутентификацию в сети;
- централизацию хранения информации об окружении пользователей;
- централизацию хранения настроек системы защиты информации на сервере.

Сетевая аутентификация и централизация хранения информации об окружении пользователя основана на использовании двух основных механизмов: NSS, описание которого приведено в 8.1.1, и PAM, описание которого приведено в 8.1.2.

В качестве источника данных для базовых системных служб на базе механизмов NSS и PAM используется служба каталогов LDAP в соответствии с 8.1.3.

Сквозная доверенная аутентификация реализуется технологией Kerberos в соответствии с 8.1.4.

Централизация хранения информации об окружении пользователей подразумевает и централизованное хранение домашних каталогов пользователей. Для этого используется СЗФС CIFS в соответствии с 6.9.

При создании ЕПП в качестве основной службы рекомендуется использовать службу FreeIPA, описанную в 8.2.

8.1.1. Механизм NSS

Механизм NSS предоставляет всем программам и службам, функционирующим на локальном компьютере, системную информацию через соответствующие программные вызовы. Он обращается к конфигурационному файлу `/etc/nsswitch.conf`, в котором указаны источники данных для каждой из системных служб. Краткое описание системных служб приведено в таблице 33.

Таблица 33

Служба	Источник данных по умолчанию	Описание
passwd	/etc/passwd	Окружение пользователя (домашний каталог, идентификатор пользователя и пр.)
shadow	/etc/shadow	Пароли пользователей
group	/etc/group	Принадлежность пользователей группам
hosts	/etc/hosts	Соответствие имен хостов адресам
services	/etc/services	Характеристики сетевых служб (порт, тип транспортного протокола)

Каждая из базовых системных служб поддерживает ряд библиотечных программных вызовов, таких как getpwent, getspent, getgrent, getservent. При выполнении данных программных вызовов производится поиск в конфигурационном файле /etc/nsswitch.conf источника данных соответствующей службы (например, passwd для получения домашнего каталога пользователя). По умолчанию в качестве источника данных системных служб используются соответствующие конфигурационные файлы в каталоге /etc (источник files). NSS при получении имени источника данных из конфигурационного файла /etc/nsswitch.conf осуществляет поиск программной разделяемой библиотеки в каталоге /lib с именем libnss_<имя_источника_данных>-<версия_библиотеки>.so, где в качестве имени источника данных выступает строка, полученная из /etc/nsswitch.conf. Например, при вызове getpwent, при условии, что в /etc/nsswitch.conf находится строка:

```
passwd : files
```

будет вызвана соответствующая функция из библиотеки /lib/libnss_files.so.

8.1.2. Механизм PAM

Механизм PAM (Pluggable Authentication Modules — подключаемые модули аутентификации) позволяет интегрировать различные низкоуровневые методы аутентификации и предоставить единые механизмы для использования прикладных программ в процессе аутентификации. Механизм состоит из набора разделяемых библиотек и конфигурационных файлов — сценариев процедур аутентификации.

В каталоге /etc/pam.d расположены конфигурационные файлы PAM для соответствующих служб, в т. ч. файл службы login в котором дана информация по проведению аутентификации.

Модули PAM вызываются при выполнении следующих функций:

1) auth — аутентификация;

2) account — получение привилегий доступа;

3) password — управление паролями;

4) session — сопровождение сессий.

Для выполнения каждой функции может быть перечислено несколько модулей PAM, которые будут вызываться последовательно, образуя стек PAM для данной задачи. Каждый вызываемый модуль возвращает в стек результат своей работы: успешный (PAM_SUCCESS), неуспешный (PAM_AUTH_ERR), игнорирующий (PAM_IGNORE) или иной. Для каждого вызова может быть указан набор управляющих флагов в виде соответствия кода возврата и того, как результат работы модуля скажется на обработке всей служебной задачи, например, ignore, ok, die. Для управления аутентификацией используются следующие флаги:

- requisite — немедленное прекращение дальнейшего выполнения служебной задачи с общим неуспешным результатом в случае неуспешного результата выполнения данного модуля;

- required — требование удачного выполнения этого модуля одновременно с выполнением всех остальных, перечисленных в данной служебной задаче;

- sufficient — в случае позитивных результатов выполнения данного модуля и всех предыдущих с флагом required в стеке задачи немедленно прекращается дальнейшее выполнение служебной задачи в целом с общим позитивным результатом. Если же модуль вернул негативный результат, то его значение игнорируется;

- optional — выполнение данного модуля никак не сказывается на результате всей задачи, но играет дополнительную информационную роль.

8.1.3. Служба каталогов LDAP

Служба каталогов LDAP — общее название клиент-серверной технологии доступа к службе каталогов X.500 с помощью протокола LDAP. Служба каталогов X.500 является средством иерархического представления информационных ресурсов, принадлежащих организации, и информации об этих ресурсах. При этом служба каталогов обеспечивает централизованное управление ресурсами и информацией о них, а также позволяет контролировать их использование третьими лицами. Каждый ресурс может принадлежать одному или более классам. Каждый класс показывает, что ресурс является определенным типом сущности и имеет определенный набор свойств. Совокупности классов могут объединяться в схемы, которые описывают типы ресурсов, применяемые в отдельно взятой предметной области.

Информация, хранящаяся в каталоге, называется «информационной базой каталога» (DIB). Пользователь каталога, который может быть как человеком, так и компьютером, получает доступ к каталогу посредством клиента. Клиент от имени пользователя каталога взаимодействует с одним или более серверами. Сервер хранит фрагмент DIB.

DIB содержит два типа информации:

- пользовательская — информация, предоставляемая пользователям и, возможно, изменяемая ими;
- административная и функциональная — информация, используемая для администрирования и/или функционирования каталога.

Множество записей, представленных в DIB, организовано иерархически в структуру дерева, известную как «информационное дерево каталога» (DIT). При этом запись в каталоге LDAP состоит из одного или нескольких атрибутов, обладает уникальным именем (DN — Distinguished Name) и может состоять только из тех атрибутов, которые определены в описании класса записи. В схеме определено, какие атрибуты являются для данного класса обязательными, а какие — необязательными. Каждый атрибут, хранящийся в каталоге LDAP (например, тип данных), имеет определенный синтаксис, который накладывает ограничения на структуру и формат его значений. Сравнение значений не является частью определения синтаксиса, а задается отдельно определяемыми правилами соответствия. Правила соответствия специфицируют аргумент, значение утверждения, которое также имеет определенный синтаксис.

Предполагается, что информация каталога достаточно статична, т.е. чаще читается, чем модифицируется. Примером подобного каталога является специализированная БД, например, телефонная книга, база данных службы DNS.

Службы каталогов LDAP могут быть использованы в качестве источника данных для базовых системных служб на базе механизмов NSS и PAM.

В результате вся служебная информация пользователей сети может располагаться на выделенном сервере в распределенной гетерогенной сетевой среде. Добавление новых сетевых пользователей в этом случае производится централизованно на сервере службы каталогов.

Благодаря предоставлению информации LDAP в иерархической древовидной форме разграничение доступа в рамках службы каталогов LDAP может быть основано на введении доменов. В качестве домена в данном случае будет выступать поддерево службы каталогов LDAP.

8.1.4. Доверенная аутентификация Kerberos

Kerberos является протоколом, обеспечивающим централизованную аутентификацию пользователей и применяющим техническое маскирование данных для противодействия различным видам атак.

Основным компонентом системы Kerberos является центр распределения ключей (KDC). Программы, настроенные на взаимодействие с Kerberos, называются «керберизованны-

ми приложениями». KDC отвечает за аутентификацию в некоторой области Kerberos. В процессе работы система Kerberos выдает билеты (tickets) на использование различных служб.

Сервером Kerberos называется компьютер, на котором выполняется серверная программа Kerberos, или сама программа KDC. Клиент Kerberos — это компьютер или программа, которые получают билет от сервера Kerberos. Обычно действия системы Kerberos инициирует пользователь, отправляющий запрос на получение услуг от некоторого сервера приложения (например, сервера почты). Kerberos предоставляет билеты принципалам, в роли которых выступают пользователи или серверные программы. Для описания принципала применяется идентификатор, состоящий из трех компонентов: основы (`primary`), экземпляра (`instance`) и области (`realm`). Данный идентификатор имеет вид:

```
основа/экземпляр@область
```

Система Kerberos выполняет следующие задачи:

1) обеспечение аутентификации в сети. Для предотвращения НСД к службам сервер должен иметь возможность идентифицировать пользователей. Кроме того, в некоторых средах важно, чтобы клиент мог идентифицировать серверы. Это исключит работу пользователей с фальшивыми серверами, созданными для незаконного сбора конфиденциальной информации;

2) защиту паролей. Открытость паролей, используемых в ряде сетевых служб, создает угрозу безопасности системы, т. к. они могут быть перехвачены и использованы для незаконного доступа к системе. Для решения данной проблемы используется техническое маскирование билетов Kerberos.
Технология Kerberos представляет собой механизм аутентификации пользователей и служб, основным достоинством которой является повышенная защищенность при использовании в сети, которая достигается механизмом защищенного обмена билетами между пользователями, службами и сервером учетных записей Kerberos. При данном механизме пароли пользователей по сети не передаются, что обеспечивает повышенную защищенность от сетевых атак. С помощью механизма открытых и закрытых ключей, а также синхронизации часов клиентских компьютеров с сервером Kerberos обеспечивается уникальность билетов и их защищенность от подделки.
В ОС используется реализация MIT Kerberos;

3) обеспечение однократной регистрации в сети. Система Kerberos дает возможность пользователю работать с сетевыми службами, пройдя лишь единожды аутентификацию на своем компьютере. При этом для обмена с приложениями дополнительно вводить пароль не требуется.
Локальные системы учетных записей пользователей и система ЕПП существуют в ОС параллельно. Различие между ними проводится с помощью разграничения

диапазонов UID (значения UID меньшие, чем 2500, относятся к локальным пользователям, а большие или равные 2500 — к пользователям ЕПП).

ВНИМАНИЕ! Обязательным требованием для функционирования аутентификации по Kerberos является синхронизация времени на клиенте и сервере. Синхронизация может быть обеспечена использованием сервера NTP (см. 6.7).

8.1.5. Централизация хранения атрибутов СЗИ в распределенной сетевой среде

В среде ОС работа пользователя осуществляется с учетом назначенных ему атрибутов, связанных с механизмами СЗИ ОС, например:

- привилегии администрирования, вхождение в группы;
- разрешенные параметры входа (список разрешенных компьютеров домена);
- политики паролей и учетных записей;
- мандатные атрибуты (диапазон доступных уровней и категорий конфиденциальности, разрешенные метки целостности, привилегии);
- параметры регистрации событий (маски регистрируемых успешных и неуспешных событий).

Одни атрибуты предназначены только для использовании в ЕПП, другие являются общими атрибутами СЗИ ОС. Доступ к мандатным атрибутам пользователей осуществляется с использованием программной библиотеки `parsec`. Данная библиотека получает из соответствующего конфигурационного файла информацию об источнике данных для СЗИ, функционирующих в условиях применения мандатного управления доступом и мандатного контроля целостности. По умолчанию используются локальные текстовые файлы.

Концепция ЕПП подразумевает централизованное хранение системной информации о пользователе (в т. ч. и его мандатные атрибуты). В этом случае вся информация хранится в службе каталогов LDAP.

8.2. Служба FreeIPA

Служба FreeIPA предназначена для реализации централизованного управления сетевыми службами, идентификацией и аутентификацией, а также для установки доверительных отношений и обеспечения взаимодействия Linux-систем с доменом Active Directory (AD).

В FreeIPA используется системный демон SSSD (System Security Services Daemon), управляющий доступом к удаленным каталогам и механизмам аутентификации, входящим в состав FreeIPA.

FreeIPA основывается на технологиях LDAP и Kerberos и поддерживает миграцию учетных записей из LDAP и NIS. FreeIPA предоставляет следующий функционал:

- DNS-сервер;
- сервер времени;
- управление доступом на основе политик.

FreeIPA позволяет создавать централизованные системы по управлению идентификацией пользователей, заданию политик доступа и аудита для сетей на основе ОС. В состав FreeIPA входят следующие компоненты:

- сервер 389 Directory Server — используется в качестве сервера LDAP;
- MIT Kerberos 5 — используется для аутентификации и единой точки входа;
- Apache и Python — используются для управления ПО, входящим в состав FreeIPA;
- BIND и DHCP — используются для управления службой DNS в сети.

В соответствии с моделью мандатного доступа служба FreeIPA реализует для зарегистрированных с помощью службы пользователей:

- задание уровней конфиденциальности;
- задание метки целостности;
- задание PARSEC-привилегий.

Управление FreeIPA доступно как через терминал, так и через веб-интерфейс.

8.2.1. Структура

Основу доменной структуры FreeIPA составляет домен IPA, в который может входить множество DNS доменов. Домен IPA воспринимается внешним доменом AD как отдельный лес доменов AD, при этом домен Primary DNS домена IPA выступает в роли корневого домена леса доменов FreeIPA.

Интеграция домена IPA с доменом AD возможна двумя способами:

- синхронизация учетных записей пользователей и их паролей (не рекомендуется);
- создание доверительных отношений между лесами доменов (рекомендуется).

В документе приводится описание только рекомендованного способа интеграции на основе доверительных отношений между доменом AD и доменом IPA.

Для обеспечения отказоустойчивости FreeIPA применяется репликация — создание реплики FreeIPA, при этом рекомендуется использовать две или три (но не более четырех) реплики. Реплики поддерживают работу в режиме «ведущий–ведомый».

8.2.2. Состав

Все необходимые компоненты службы FreeIPA входят в состав пакетов, приведенных в таблице 34.

Таблица 34

Наименование	Описание
`freeipa-admintools`	Пакет администрирования FreeIPA, содержит набор утилит по управлению сервером FreeIPA
`freeipa-client`	Клиентская часть FreeIPA. Пакет должен устанавливаться на все клиентские компьютеры, входящие в домен
`freeipa-server`	Серверная часть FreeIPA. Пакет должен устанавливаться на контроллере домена. При установке данного пакета также устанавливается средство администрирования `ipa` и клиентская часть
`freeipa-server-dns`	Пакет, предназначенный для установки или интеграции с DNS сервером
`freeipa-server-trust-ad`	Пакет для интеграции с Active Directory от Microsoft путем установки доверительных отношений
`fly-admin-freeipa-server`	Графическая утилита управления FreeIPA
`astra-freeipa-server`	Инструмент командной строки управления FreeIPA
`fly-admin-freeipa-client`	Графическая утилита управления клиентом FreeIPA
`astra-freeipa-client`	Инструмент командной строки управления клиентом FreeIPA

Служба FreeIPA состоит из ядра, отвечающего за основной функционал системы, ряда интерфейсов (LDAP, Kerberos, Config, RPC) и модулей расширения, предназначенных для расширения командного интерфейса утилит и настройки необходимых служб и подсистем, что позволяет повышать функциональность FreeIPA.

В FreeIPA возможно использование следующих группы модулей расширения:

- `freeipa-client-*` — расширение, необходимое клиентской части FreeIPA;
- `freeipa-admintools-*` — расширение утилиты администрирования FreeIPA;
- `freeipa-server-*` — расширение, необходимое для организации хранения атрибутов на сервере FreeIPA.

Описание пакетов приведено на справочных страницах `man`, список которых приведен в таблице 35.

Таблица 35

Наименование	Описание
`ipa`	Администрирование домена IPA

Продолжение таблицы 35

Наименование	Описание
default.conf	Образец конфигурационного файла default.conf
ipa-client-install	Настройка клиентской части FreeIPA
ipa-server-install	Настройка серверной части FreeIPA
ipa-server-upgrade	Обновление сервера FreeIPA
ipa-dns-install	Утилита добавления DNS как службы на серверной части FreeIPA
ipa-backup	Резервное копирования мастер-сервера FreeIPA
ipactl	Интерфейс управления серверной частью FreeIPA
ipa-advise	Предоставляет рекомендации по конфигурациям для различных вариантов использования
ipa-cacert-manage	Управление сертификатами CA на FreeIPA
ipa-certupdate	Обновление локальных БД сертификатов FreeIPA вместе с сертификатами от сервера
ipa-client-automount	Настройка автомонтирования и ФС NFS для FreeIPA
ipa-compat-manage	Включение и выключение модуля совместимости схемы
ipa-csreplica-manage	Управление репликой FreeIPA CS
ipa-getcert	Инструмент ipa-getcert выдает запросы службе certmonger от имени вызывающего пользователя
ipa-getkeytab	Получение keytab-файла. Keytab — это файл с одним или несколькими закрытыми ключами для принципала Kerberos. Keytab-файлы используются службами, например, sshd, при аутентификации Kerberos
ipa-join	Подключение хоста к области FreeIPA и получение keytab-файла для размещения службы хоста принципала Kerberos
ipa-kra-install	Установка KRA на серверной части FreeIPA
ipa-ldap-updater	Обновление настроек FreeIPA LDAP
ipa-managed-entries	Включения и выключение модулей схемы управляемых модулей ввода
ipa-nis-manage	Включение и выключение модуля прослушивателя NIS
ipa-otptoken-import	Импорт OTP-токенов из RFC 6030 XML файлов
ipa-replica-conncheck	Проверка сетевого подключения реплики и мастер-сервера перед установкой
ipa-replica-install	Создание реплики FreeIPA
ipa-replica-manage	Управление репликой FreeIPA
ipa-replica-prepare	Создание файла реплики FreeIPA
ipa-restore	Восстановление мастер-сервера FreeIPA
ipa-rmkeytab	Удаление принципала Kerberos из keytab-файла
ipa-server-certinstall	Установка новых SSL-сертификатов сервера

Окончание таблицы 35

Наименование	Описание
`ipa-winsync-migrate`	Полный переход от пользователей AD, созданных `winsync`, к обычным пользователям AD
`ipa-upgradeconfig`	Обновление конфигурации Apache FreeIPA

8.2.3. Предварительная настройка контроллера домена

При развертывании FreeIPA в качестве контроллера домена следует использовать отдельный компьютер с фиксированным IP-адресом, который в дальнейшем не должен изменяться.

ВНИМАНИЕ! Работа FreeIPA осуществляется только при отключенном режиме `AstraMode` веб-сервера Apache2 (описание режима приведено в 11.2). Инструменты установки сервера FreeIPA `astra-freeipa-server` и/или `fly-admin-freeipa-server` автоматически выключают данный режим.

Для штатного функционирования FreeIPA необходимо выполнение следующих условий:

1) в настройках сетевого интерфейса в качестве первичного DNS должен быть указан IP-адрес компьютера;

2) компьютеру должно быть присвоено полное имя, при этом использовать доменное имя второго уровня или ниже (например, `domain.net`, `domain.test.net`). Имя компьютера можно задать с помощью команды:

```
hostnamectl set-hostname <полное_имя_сервера>
```

Пример

```
hostnamectl set-hostname server.domain.net
```

Инструмент `hostname` должен возвращать полное имя компьютера (например, `server.domain.net`);

3) разрешение имен должно быть настроено таким образом, чтобы имя компьютера разрешалось, в первую очередь, как полное имя (разрешение имен для сервера FreeIPA добавляется автоматически при установке, см. 8.2.4).

Пример

Записи в файле `/etc/hosts` для сервера FreeIPA:

```
127.0.0.1       localhost
192.168.1.1     server.domain.net server
```

4) должна быть выполнена синхронизация времени в ОС для аутентификации по Kerberos. Синхронизация может быть настроена с помощью протокола синхронизации времени (см. 8.2.8).

Настройка всех компонентов сервера FreeIPA осуществляется автоматически при установке сервера с помощью `astra-freeipa-server`/`fly-admin-freeipa-server`.

8.2.4. Установка компонентов FreeIPA

Программные компоненты FreeIPA входят в состав ОС и могут быть установлены с помощью стандартной графической утилиты для работы с пакетами Synaptic либо из терминала.

ВНИМАНИЕ! Без установки пакетов расширения совместно с соответствующими основными пакетами невозможно централизованное хранение атрибутов СЗИ в распределенной сетевой среде, что может привести к невозможности входа пользователей в систему.

Для развертывания FreeIPA необходимо:

1) на компьютере, предназначенном на роль контроллера домена, установить следующие программные компоненты:
   а) серверную часть FreeIPA `astra-freeipa-server`, для установки из терминала ввести команду:

   ```
   apt install astra-freeipa-server
   ```

   б) графическую утилиту управления серверной частью FreeIPA `fly-admin-freeipa-server`, для установки из терминала ввести команду:

   ```
   apt install fly-admin-freeipa-server
   ```

   При установке графической утилиты `fly-admin-freeipa-server` автоматически будет установлен инструмент командной строки `astra-freeipa-server`;
2) на клиентских компьютерах установить следующие программные компоненты:
   а) `astra-freeipa-client`, для установки из терминала ввести команду:

   ```
   apt install astra-freeipa-client
   ```

   б) `fly-admin-freeipa-client`, для установки из терминала ввести команду:

   ```
   apt install fly-admin-freeipa-client
   ```

   При установке графической утилиты `fly-admin-freeipa-client` автоматически будет установлен инструмент командной строки `astra-freeipa-client`.

При установке данных компонентов FreeIPA автоматически устанавливается служба синхронизации времени `chrony` (при этом удаляется служба `systemd-timesyncd`), а также все необходимые пакеты в зависимости от назначения компьютера.

ВНИМАНИЕ! Для создания ЕПП FreeIPA, в которое должны быть интегрированы клиенты, поддерживающие режимы мандатного управления доступом и/или мандатного контроля целостности, в роли сервера FreeIPA необходимо использовать компьютер с включенными соответствующими режимами. После установки сервера FreeIPA изменение его режимов работы мандатного управления доступом и мандатного контроля целостности не поддерживается.

8.2.5. Создание контроллера домена и запуск служб FreeIPA

8.2.5.1. С использованием графической утилиты

Для создания контроллера домена и запуска служб FreeIPA с помощью графической утилиты `fly-admin-freeipa-server` (описание утилиты см. в электронной справке) необходимо запустить графическую утилиту командой:

```
fly-admin-freeipa-server
```

В окне программы, при необходимости, указать следующие данные:

- в поле «Домен» — имя домена, определяется автоматически на основе полного имени компьютера;
- в поле «Имя компьютера» — имя компьютера, определяется автоматически;
- в поле «Пароль» — задать пароль администратора домена. Указанный пароль будет использоваться для входа в веб-интерфейс FreeIPA и при работе с инструментом командной строки.

Далее нажать кнопку **[Создать]**.

После успешного конфигурирования необходимых служб и инициализации домена появится ссылка для перехода в веб-интерфейс FreeIPA, в котором можно продолжить настройку. Порядок работы с FreeIPA используя веб-интерфейс приведен в 8.2.14.

8.2.5.2. С использованием инструмента командной строки

Для создания контроллера домена и запуска служб FreeIPA с помощью инструмента командной строки `astra-freeipa-server` выполнить команду:

```
astra-freeipa-server -d <имя_домена> -n <имя_компьютера> -o
```

После выполнения команды будет определен адрес компьютера и будут выведены на экран все исходные данные.

Пример

```
compname= server
domain= domain.net
будет использован ip address = 192.168.32.97 или укажите ip адрес ключем
     -ip
продолжать ? (y\n)
```

Для подтверждения данных ввести y и нажать **<Enter>**. После подтверждения появится запрос на установку пароля администратора домена. Указанный пароль будет использоваться для входа в веб-интерфейс FreeIPA и при работе с инструментом командной строки.

После ввода пароля будет выполнено конфигурирование необходимых служб и инициализации домена, ход выполнения будет отображаться на экране. После успешного завершения инициализации на экран будут выведены сообщения о перезапуске системных служб, а также данные контроллера домена и ссылка для веб-интерфейса FreeIPA, в котором можно продолжить настройку. Порядок работы с FreeIPA используя веб-интерфейс приведен в 8.2.14.

Пример

```
Restarting Directory Service
Restarting krb5kdc Service
Restarting kadmin Service
Restarting named Service
Restarting ipa_memcached Service
Restarting httpd Service
Restarting ipa-custodia Service
Restarting ipa-otpd Service
Restarting ipa-dnskeysyncd Service
Starting chronyd Service
ipa: INFO: The ipactl command was successful
Существует настроенный домен
host = server.domain.net
basedn = dc=domain,dc=net
domain = domain.net
xmlrpc_uri = https://server.domain.net/ipa/xml
WEB: https://server.domain.net
```

После завершения работы мастера требуется убедиться в наличии открытых портов на сервере:

1) TCP Ports:
   - 80, 443: HTTP/HTTPS;

- 389, 636: LDAP/LDAPS;
- 88, 464: kerberos;
- 53: bind;

2) UDP Ports:
   - 88, 464: kerberos;
   - 53: bind;
   - 123: ntp.

Параметры инструмента командной строки `astra-freeipa-server` приведены в таблице 36.

Таблица 36

Параметр	Описание
`-h, --help`	Вывести справку по командам
`-d`	Задать имя домена
`-n`	Задать имя компьютера
`-ip`	Задать IP-адрес веб-интерфейса. Если адрес не задан, то инструмент пытается определить его автоматически
`-y`	Отключить запрос подтверждения после вывода заданных параметров запуска
`-i`	Вывести информацию о существующем домене
`-px`	Получить пароль администратора домена из `stdin`
`-p`	Получить пароль администратора домена из командной строки (небезопасно)
`-s`	Включить установку и запуск поддержки AD SMB
`-c`	Запретить изменять файл `/etc/hosts`
`-o`	Запретить проверку регистрации домена. Применяется при установке в изолированной сети
`-e`	Отключить установку и запуск собственной службы DNS
`-U`	Удалить все настройки
`-l`	Указать сертификат (имя компьютера и домена должны совпадать)
`-lp`	Указать пароль сертификата

8.2.5.3. Конфигурационный файл FreeIPA

Настройки сервера FreeIPA содержатся в конфигурационном файле `/etc/ipa/default.conf`. Формат файла:

```
имя_параметра=значение # Комментарий
```

Описание параметров конфигурационного файла приведено в таблице 37.

Таблица 37

Параметр	Описание
basedn <запись_DN>	Задает базовую запись DN, используемую при выполнении операций LDAP. Запись должна быть в формате DN (например, dc=example,dc=com)
context <контекст>	Задает контекст, в котором выполняется IPA. Работа IPA определяется в зависимости от контекста. Текущие определенные контексты — cli и server (клиент и сервер). Кроме того, значение используется для загрузки файла /etc/ipa/<контекст>.conf для применения контекстной конфигурации. Например, если необходимо всегда выполнять клиентские запросы в подробном режиме, но при этом не использовать подробный режим на сервере, то следует добавить параметр verbose в /etc/ipa/cli.conf
debug <boolean>	При значении True предоставляет подробную информацию. В частности, значение debug устанавливается для глобального уровня log-журнала. Значение по умолчанию False
domain <имя_домена>	Домен сервера FreeIPA, например, example.com
enable_ra <boolean>	Значение True определяет, что будет использоваться удаленная служба центра аутентификации, например, когда служба Dogtag используется в качестве центра аутентификации. Эта настройка применяется исключительно в конфигурации сервера IPA
fallback <boolean>	Значение True определяет, что клиент IPA должен выполнять возврат и обращаться к другим службам в случае сбоя первого подключения
host <имя_хоста>	Задает имя хоста локальной системы
in_server <boolean>	Определяет, будут ли запросы направляться на сервер IPA (True) или обрабатываться локально (False). Внутри IPA они используются подобно контексту. Та же самая IPA-конструкция используется IPA-инструментами командной строки и сервера. Этот параметр указывает конструкции, выполнить ли команду так, как если бы она была на сервере или переслать ее через XML-RPC на удаленный сервер
in_tree <boolean>	Используется при разработке. Параметр указывается при необходимости выполнить код в исходном дереве
interactive <boolean>	Определяет, следует ли запрашивать значения. Значение по умолчанию True
ldap_uri <URI>	Указывает URI сервера IPA LDAP для подключения. Схема URI может быть ldap или ldapi. По умолчанию используется ldapi, например, ldapi://%2fvar%2frun%2fslapd-EXAMPLE-COM.socket

Продолжение таблицы 37

Параметр	Описание

`log_logger_<уровень> <регулярное_выражение, ...>`	Перечень регулярных выражений `regexp`, разделенных запятыми. Логированиям (loggers), соответствующим регулярным выражениям, будет присвоен уровень `<уровень>`.
	Уровни логирования (logger levels) могут быть явно заданы для конкретных логирований в отличие от глобального уровня журналирования (global logging level). Если имя логирования соответствует регулярному выражению, то ему присваивается соответствующий уровень. Этот элемент конфигурации должен начинаться с `log_logger_level_`, а затем должен следовать символический или числовой уровень журнала (log level). Этот элемент конфигурации полезен, если требуется просмотреть вывод журнала только для одного или нескольких выбранных логирований. Обычно логирования привязаны к классам и модулям.
	Пример
	Настроить логирование модуля `ipalib.dn` на уровень для отладки:
	`log_logger_level_debug = ipalib\.dn\..*`
	Настроить логирование `ipa.plugins.dogtag` на уровень 35:
	`log_logger_level_35 = ipalib\.plugins\.dogtag`
	Примечание. Имена логирований (logger names) — список с разделяющей точкой, образующий путь в данном дереве логирования (logger tree). Символ точки также является метасимволом регулярного выражения (соответствует любому символу), поэтому, чтобы избежать точек в именах логирования, обычно требуется перед ними ставить обратную косую черту «\.».

`mode <режим_работы>`	Определяет режим работы сервера. В настоящее время поддерживаемыми значениями являются эксплуатация (`production`) и разработка (`development`). При работе в режиме `production` некоторые самопроверки пропускаются для повышения производительности
`mount_ipa <URI>`	Задает точку монтирования для регистрации сервера разработки. По умолчанию `/ipa/`
`prompt_all <boolean>`	Определяет, должны ли для клиента IPA запрашиваться все параметры, в т. ч. необязательные значения. По умолчанию устанавливается `False`
`ra_plugin <имя>`	Задает имя назначенного для использования CA. Текущими параметрами являются `dogtag` и `selfsign`. Настройка на стороне сервера. Изменять значение не рекомендуется, т.к. назначенный CA настраивается только во время первоначальной установки
`realm <realm>`	Указывает область Kerberos

Продолжение таблицы 37

Параметр	Описание
`session_auth_duration <интервал_времени>`	Задает допустимый интервал для времени кэширования учетных данных проверки подлинности в сеансе. По истечении срока действия учетные данные будут автоматически переопределены. Например, `2 hours`, `1h:30m`, `10 minutes`, `5min, 30sec`
`session_duration_type <тип_вычисления>`	Определяет способ вычисления срока действия сеанса. Возможные значения: - `inactivity_timeout` — срок действия увеличивается на значение `session_auth_duration` каждый раз, когда пользователь обращается к службе; - `from_start` сроком действия сеанса является начало сеанса пользователя плюс значение `session_auth_duration`
`server <имя_сервера>`	Задает имя сервера IPA
`skip_version_check <boolean>`	Пропустить проверки версии API клиента и сервера. Может привести к ошибкам/сбоям, когда новые клиенты обращаются к прежним серверам. Использовать с осторожностью
`startup_timeout <время_ожидания>`	Определяет время ожидания в секундах до начала запуска сервера. Значение по умолчанию 120 секунд
`startup_traceback <boolean>`	Если сервер IPA не запускается при заданном значении `True`, то сервер будет пытаться сгенерировать обратное python-отслеживание, чтобы облегчить определение причины сбоя
`validate_api <boolean>`	Используется внутри исходного пакета IPA для проверки неизменности API. Применяется для предотвращения регрессии. Если установлено значение `True`, то некоторые ошибки игнорируются, чтобы обеспечить загрузку инфраструктуры IPA, достаточной для проверки API, даже если дополнительные компоненты не установлены. Значение по умолчанию `False`
`verbose <boolean>`	При установке значения `True` предоставляет дополнительные сведения — устанавливает глобальный уровень журнала (global log level) на событие `info`

Окончание таблицы 37

Параметр	Описание
wait_for_dns <boolean>	Контролирует синхронность работы IPA команд dnsrecord-{add,mod,del}. Команды DNS будут повторять DNS-запросы указанное количество попыток до тех пор, пока DNS-сервер возвращает ответ up-to-date на запрос об измененных записях. Задержка между повторными попытками одна секунда. Команды DNS будут порождать исключение DNSDataMismatch, если ответ не совпадает с ожидаемым значением, даже после указанного числа попыток. DNS-запросы будут отправлены в очередь для разрешения решателем, который сконфигурирован в файле /etc/resolv.conf на сервере IPA. **ВНИМАНИЕ!** Не включать параметр в режиме production! Это может вызвать проблемы, если решатель (resolver) на сервере IPA использует кэширование сервера, а не локального сервера авторизации или, например, если DNS-ответы будут изменены шлюзом DNS64. Значение по умолчанию disable (выключено), параметр отсутствует
xmlrpc_uri <URI>	Задает URI сервера XML-RPC для клиента. Может использоваться IPA и используется некоторыми внешними средствами, такими как ipa-getcert. Например, https://ipa.example.com/ipa/xml
jsonrpc_uri <URI>	Задает URI сервера JSON для клиента. Используется IPA. Если параметр не задан, он наследуется от xmlrpc_uri. Например, https://ipa.example.com/ipa/json
rpc_protocol <URI>	Задает тип RPC-вызовов IPA makes: jsonrpc или xmlrpc. По умолчанию используется jsonrpc

Более подробное описание конфигурационного файла приведено в руководстве man.

Пример

Конфигурационный файл /etc/ipa/default.conf

```
[global]
host = server.domain.net
basedn = dc=domain,dc=net
realm = DOMAIN.NET
domain = domain.net
xmlrpc_uri = https://server.domain.net/ipa/xml
ldap_uri = ldapi://%2fvar%2frun%2fslapd-DOMAIN-NET.socket
enable_ra = False
ra_plugin = none
mode = production
```

8.2.5.4. Управление службами FreeIPA

Для проверки работы и управления службами FreeIPA используется команда `ipactl`:

1) запуск служб FreeIPA:

```
ipactl start
```

2) отображение текущего состояния всех служб FreeIPA:

```
ipactl status
```

3) перезапуск служб FreeIPA:

```
ipactl restart
```

4) остановка служб FreeIPA:

```
ipactl stop
```

Дополнительно с командой `ipactl` можно использовать параметр `-d` для выполнения команды в режиме отладки:

```
ipactl start -d
```

8.2.6. Ввод компьютера в домен

8.2.6.1. Настройка клиентского компьютера

Для ввода нового компьютера в домен необходимо выполнение условий:

1) клиентский компьютер не должен входить в другой домен;

2) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя.

Пример

Файл `/etc/hosts`:

```
127.0.0.1 localhost
192.168.1.2 client.domain.net client
192.168.1.1 server.domain.net server
```

Инструмент `hostname` должен возвращать полное имя компьютера, например `client.domain.net`.

Разрешение имен также может быть настроено с помощью сервера DNS в соответствии с 6.5;

3) клиентский компьютер и сервер FreeIPA должны видеть друг друга в сети. Для проверки можно использовать команду:

```
ping <ip-адрес>
```

4) должна быть выполнена синхронизация времени в ОС для аутентификации по Kerberos. Синхронизация может быть настроена с помощью протокола синхронизации времени (см. 8.2.8);

5) наличие установленного пакета `astra-freeipa-client`.

Далее необходимо настроить DNS-адрес сервера FreeIPA на клиентском компьютере одним из способов:

1) отредактировав конфигурационный файл `resolv.conf`;

2) отредактировав файл `interfaces`;

3) с помощью утилиты NetworkManager.

ВНИМАНИЕ! В некоторых случаях, если адрес сервера FreeIPA стоит в DNS не первым, клиентский компьютер может не находить домен.

Ввод компьютера в домен можно выполнить с помощью инструмента командной строки или графической утилиты.

8.2.6.2. Ввод компьютера в домен с использованием инструмента командной строки

Для ввода компьютера в домен с использованием инструмента командной строки `astra-freeipa-client` необходимо выполнить команду:

```
sudo astra-freeipa-client -d <контроллер_домена> -u admin -px
```

Для просмотра перечня дополнительных параметров для запуска с командой `astra-freeipa-client` выполнить:

```
astra-freeipa-client --help
```

После ввода компьютера в домен необходимо выполнить перезагрузку.

8.2.6.3. Ввод компьютера в домен с использованием графической утилиты

Для ввода компьютера в домен с использованием графической утилиты `fly-admin-freeipa-client` («Настройка FreeIPA клиент Fly», описание утилиты см. в электронной справке) необходимо запустить графическую утилиту командой:

```
fly-admin-freeipa-client
```

В открывшемся окне, приведенном на рис. 2, следует ввести:

1) в поле «Домен» — имя домена;
2) в поле «Логин» — имя администратора домена;
3) в поле «Пароль» — пароль администратора домена.

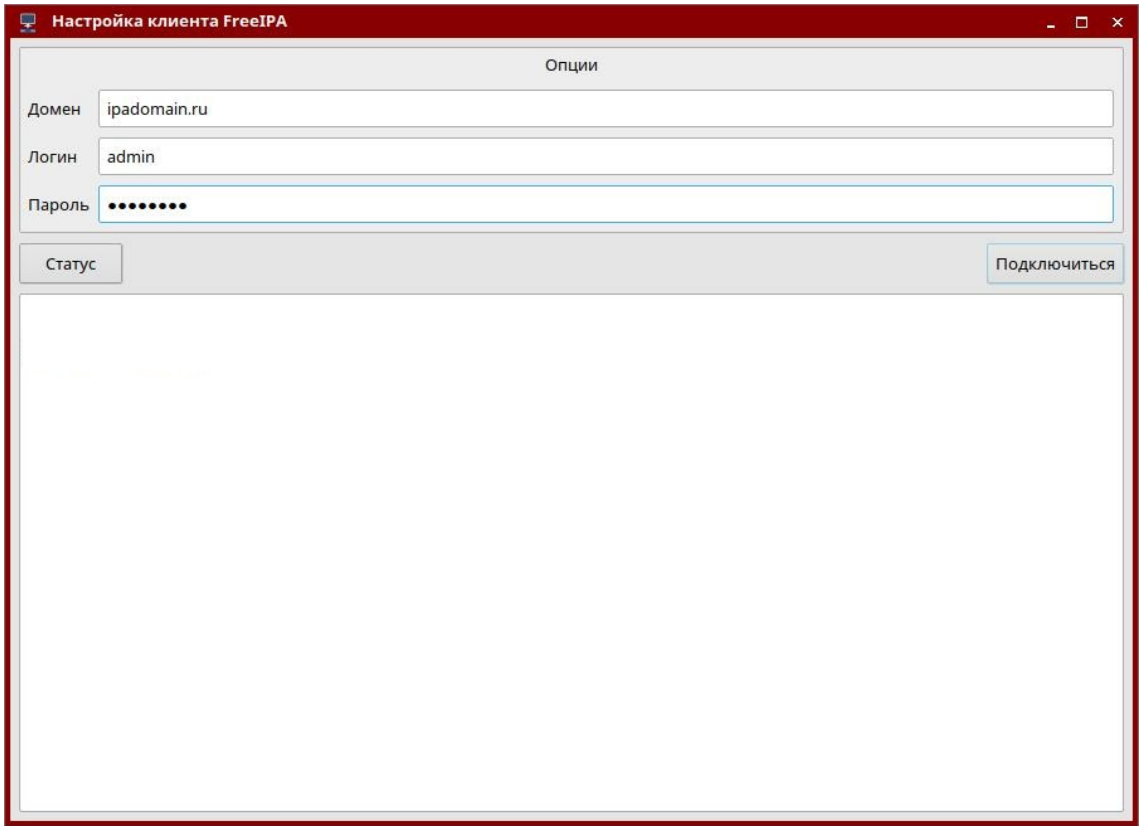

Рис. 2

После ввода данных следует нажать кнопку **[Подключиться]**.

8.2.6.4. Отображение списка доменных учетных записей в окне входа в ОС

По умолчанию список доменных учетных записей не отображается в окне входа в ОС, в том числе если в графической утилите `fly-admin-dm` включено отображение списка пользователей и настроен диапазон отображения, содержащий системные идентификаторы пользователей (`uid`) домена FreeIPA. Описание графической утилиты `fly-admin-dm` см. в электронной справке.

Для включения отображения списка доменных пользователей, дополнительно к настройкам с помощью графической утилиты `fly-admin-dm`, необходимо откорректировать конфигурационный файл `/etc/sssd/sssd.conf`, изменив в секции `[domain]` значение параметра `enumerate` на `TRUE` или добавив параметр, если он отсутствует:

```
[domain]
enumerate = True
```

При включении отображения списка доменных пользователей в окне входа в ОС рекомендуется ограничивать выводимый список путем задания соответствующего диапазона в графической утилите `fly-admin-dm`, т. к. вывод большого списка пользователей может снизить производительность.

8.2.7. Шаблоны конфигурационных файлов

Служба FreeIPA в процессе работы осуществляет конфигурирование сетевых служб (Samba, Kerberos, LDAP и т. п.) с помощью их конфигурационных файлов. Для удобства существуют шаблоны конфигурационных файлов, модифицируемых службой FreeIPA. Шаблоны расположены в каталогах `/usr/share/ipa` и `/usr/share/ipa/advise/legacy/`.

Перечень шаблонов конфигурационных файлов приведен в таблице 38.

Таблица 38

Имя шаблона	Служба	Описание, размещение конфигурационного файла
`*.ldif`	389-BASE	LDAP схемы
`default.conf`	IPA	`/etc/ipa/default.conf`
`ipa-httpd.conf.template`	IPA	`/etc/systemd/system/apache2.service.d/ipa.conf`
`ipa-kdc-proxy.conf.template`	IPA	`/etc/ipa/kdcproxy/ipa-kdc-proxy.conf`
`sssd.conf.template`	SSSD	`/etc/sssd/sssd.conf`
`ldap.conf`	LDAP клиенты	`/etc/ldap/ldap.conf`
`krb5.conf.template`	Kerberos клиенты	`/etc/krb5.conf`
`kdc.conf.template`	Kerberos KDC	`/etc/krb5kdc/kdc.conf`
`certmap.conf.template`	389-BASE	`/etc/dirsrv/config/certmap.conf`
`bind.named.conf.template`	BIND9	`/etc/bind/named.conf`
`custodia.conf.template`	IPA	`/etc/ipa/custodia/custodia.conf`
`smb.conf.template`	Samba	`/etc/samba/smb.conf`
`opendnssec_conf.template`	Opendnssec	`/etc/opendnssec/conf.xml`
`pam.conf.sssd.template`	SSSD	`/etc/pam.d/`
`mldap.conf`	PARSEC	`/etc/parsec/mldap.conf`
`mswitch.conf`	PARSEC	`/etc/parsec/mswitch.conf`
`krb.con.template`	IPA	`/usr/share/ipa/html`
`krbrealm.con.template`	IPA	`/usr/share/ipa/html`
`krb5.ini.template`	IPA	`/usr/share/ipa/html`

8.2.8. Настройка синхронизация времени

При установке и инициализации FreeIPA конфигурация службы синхронизации времени `chronyd` настраивается автоматически для использования российских серверов точного времени ВНИИФТРИ.

При развертывании FreeIPA в сети без доступа к общедоступным серверам точного времени необходимо в конфигурационном файле `/etc/chrony/chrony.conf` указать IP-адрес локального сервера времени.

Затем перезапустить службу синхронизации времени:

```
systemctl restart chronyd
```

ВНИМАНИЕ! При использовании виртуальных машин процедура перезапуска автоматической синхронизации обязательно должна быть выполнена после каждого перезапуска и/или отката виртуальных машин.

8.2.9. Создание резервной копии и восстановление

Поддерживается создание резервных копий двух типов: полная резервная копия всей системы и резервная копия только данных. Установка пароля на резервные копии не поддерживается.

Резервные копии хранятся в каталоге `/var/lib/ipa/backup`. Для полного резервного копирования и резервного копирования данных используются, соответственно, обозначения `ipa-full-YEAR-MM-DD-HH-MM-SS` и `ipa-data-YEAR-MM-DD-HH-MM-SS`, где `YEAR-MM-DD-HH-MM-SS` — год, месяц, день, час, минуты и секунды в часовом поясе GMT создания резервной копии, например, 2018-03-05-10-30-22.

В каталоге `/var/lib/ipa/backup` размещается файл, в котором приведена информация о резервных копиях: тип, система, даты резервного копирования, версия FreeIPA, версия резервного копирования и др.

ВНИМАНИЕ! Резервную копию невозможно восстановить на другом компьютере или на другой версии FreeIPA.

Резервное копирование выполняется с помощью команды `ipa-backup`. Дополнительно с командой возможно использовать параметры, приведенные в таблице 39.

Таблица 39

Параметр	Описание
`--data`	Выполнить резервное копирование только данных. По умолчанию выполняется резервное копирование всех файлов FreeIPA и данных
`--logs`	Включить в резервную копию файлы журнала службы FreeIPA
`--online`	Выполнить резервное копирование без остановки сервера. Используется с параметром `--data`
`-v`, `--verbose`	Вывести сведения об отладке
`-d`, `--debug`	Вывести детальные сведения об отладке. Используется с параметром `--verbose`
`-q`, `--quiet`	Вывести только сведения о ошибках
`--log-file=FILE`	Выполнять регистрацию событий в файл `FILE`

8.2.10. Создание резервного сервера FreeIPA (настройка репликации)

Новый сервер FreeIPA возможно настроить на выполнение роли резервного сервера (реплики). Созданный резервный сервер будет являться точной копией исходного сервера FreeIPA и приравниваться к мастер-серверу. Изменения, внесенные в любой мастер-сервер, автоматически реплицируются на другие мастер-сервера.

Для добавления резервного сервера в домен FreeIPA необходимо выполнить следующие действия:

1) резервному серверу назначить фиксированный IP-адрес, который впоследствии не должен изменяться, и зарегистрировать резервный сервер в качестве клиента в домене FreeIPA в соответствии с 8.2.6;

2) на резервном сервере установить программный компонент `astra-freeipa-server` в соответствии с 8.2.4;

3) на резервном сервере запустить службу SSH, выполнив команду: `sudo systemctl enable --now ssh`

4) на основном сервере домена с использованием инструмента `astra-freeipa-server-crt` выпустить сертификат для резервного сервера с последующим переносом сертификата в домашний каталог администратора резервного сервера:

```
astra-freeipa-server-crt --host <реплика> --export --push \
     <администратор>@<IP-адрес> --pin <пароль> --48
```

где <реплика> — полное доменное имя резервного сервера;

<администратор> — имя администратора резервного сервера;

<IP-адрес> — IP-адрес резервного сервера;

<пароль> — пароль к создаваемому контейнеру закрытого ключа и сертификата;

--48 — указание создать сертификат для FreeIPA версии 4.8.x (по умолчанию будут создаваться сертификаты для FreeIPA версии 4.6.x) .

Во время выпуска сертификата на все вопросы ответить «y» («Да»), и затем вести пароль администратора резервного сервера;

5) на резервном сервере из домашнего каталога администратора, в который ранее был скопирован контейнер закрытого ключа и сертификата, выполнить команду:

```
astra-freeipa-replica -a <реплика>.p12 --pin <пароль>
```

где <реплика> — полное доменное имя резервного сервера (в таком формате задается имя файла контейнера закрытого ключа и сертификата);

<пароль> — пароль к созданному контейнеру закрытого ключа и сертификата.

В ходе выполнения команды необходимо вести пароль администратора домена, а затем на все вопросы ответить «y» («Да»).

В случае успешной активации резервный сервер должен появиться на топологической схеме в веб-интерфейсе FreeIPA («IPA-сервер — Топология — Topology Graph», см. рис. 3).

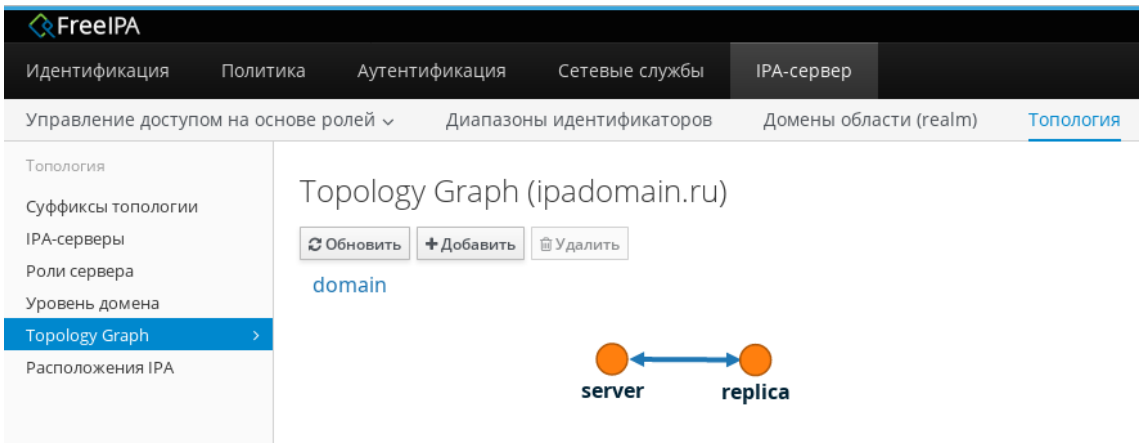

Рис. 3

8.2.11. Доверительные отношения между доменами

8.2.11.1. Общие сведения

Перед настройкой доверительных отношений контроллер домена AD должен быть настроен и работоспособен, а службы FreeIPA запущены в соответствии с 8.2.5.

ВНИМАНИЕ! Не удастся установить доверительные отношения с доменом AD, если имя области сервера FreeIPA не совпадает с его доменным именем.

Для создания доверительных отношений сервера FreeIPA с доменом AD служит пакет `freeipa-server-trust-ad`. Установка службы доверительных отношений выполняется с помощью инструмента командной строки `ipa-adtrust-install`.

В случае необходимости переустановки ранее удаленных объектов или поврежденных файлов конфигурации команду `ipa-adtrust-install` можно запустить несколько раз. Таким образом могут быть созданы новая конфигурация Samba (файл `smb.conf`) и конфигурация, на которой базируется регистрация. Некоторые элементы, например конфигурация локального диапазона, не могут быть изменены в результате повторного запуска команды `ipa-adtrust-install`, т. к. в данном случае изменения могут затронуть и другие объекты.

При выполнении команды `ipa-adtrust-install` для разрешения обмена информацией между доменами FreeIPA и AD необходимо удостовериться, что открыты следующие порты:

- 135/tcp EPMAP
- 138/tcp NetBIOS-DGM
- 139/tcp NetBIOS-SSN
- 445/tcp Microsoft-DS
- 1024/tcp
- 3268/tcp Microsoft-GC
- 138/udp NetBIOS-DGM
- 139/udp NetBIOS-SSN
- 389/udp LDAP

Дополнительно с командой `ipa-adtrust-install` возможно использовать параметры, приведенные в таблице 40.

Таблица 40

Параметр	Описание
`-d, --debug`	Вывести детальные сведения об отладке
`--netbios-name=NETBIOS_NAME`	Задать имя NetBIOS для домена FreeIPA. Если не указано, то оно определяется на основе ведущего компонента DNS-имени домена. Если запустить команду `ipa-adtrust-install` во второй раз с другим именем NetBIOS, то это имя изменится. **ВНИМАНИЕ!** Изменение имени NetBIOS может нарушить существующие доверительные отношения с другими доменами

Продолжение таблицы 40

Параметр	Описание
`--add-sids`	Добавить SID для существующих пользователей и групп в качестве активных на заключительных шагах запуска команды `ipa-adtrust-install`. Если в среде существует множество действующих пользователей и групп и несколько резервных серверов, то выполнение данного действия может привести к высокой скорости репликации трафика и снижению производительности всех серверов FreeIPA в среде. Чтобы избежать этого рекомендуется генерацию SID запускать после выполнения команды `ipa-adtrust-install`, для этого загрузить отредактированную версию `ipa-sidgen-task-run.ldif` с помощью команды `ldapmodify` на сервере домена AD
`--add-agents`	Добавить мастер-сервер FreeIPA в список для предоставления информации о пользователях доверенных лесов. Мастер-сервер FreeIPA может предоставлять эту информацию клиентам SSSD. Мастер-серверы FreeIPA не добавляются в список автоматически, т. к. для этого требуется перезапуск службы LDAP на каждом из них. Компьютер, на котором выполнена команда `ipa-adtrust-install`, добавляется автоматически. **ВНИМАНИЕ!** Мастер-серверы FreeIPA, на которых команда `ipa-adtrust-install` не была запущена, могут работать с информацией о пользователях доверенных лесов только если они активированы путем выполнения команды `ipa-adtrust-install` на любом другом мастер-сервере FreeIPA
`-U`, `--unattended`	Удалить без подтверждения. Ввод данных пользователем не будет запрашиваться
`--rid-base=RID_BASE`	Задать первое значение RID локального домена. Первый Posix ID локального домена будет присвоен данному RID, второй будет присвоен RID+1 и т. д.
`--secondary-rid-base=SECONDARY_RID_BASE`	Задать начальное значение вторичного RID диапазона, которое используется только в том случае, если пользователь и группа используют один и тот же Posix ID
`-A`, `--admin-name=ADMIN_NAME`	Задать имя пользователя с правами администратора для данного сервера FreeIPA. По умолчанию `admin`

Окончание таблицы 40

Параметр	Описание
`-a, --admin-password=password`	Задать пароль для пользователя с правами администратора для данного сервера FreeIPA. Будет запрашиваться в интерактивном режиме если параметр `-U` не указан. Учетные данные администратора будут использованы для получения билета Kerberos перед настройкой поддержки доверительных отношений перекрестной области, а также в дальнейшем, чтобы убедиться, что билет содержит MS-PAC сведения, необходимые для фактического добавления доверительных отношений с доменом AD при помощи команды `ipa trust-add -type=ad`

8.2.11.2. Предварительная настройка

Серверы домена AD и домена FreeIPA должны находиться в одной сети и на обоих серверах должна успешно выполняться команда:

```
ping <IP-адрес>
```

где `<IP-адрес>` — IP-адрес сервера домена AD при выполнении команды на сервере домена FreeIPA или IP-адрес сервера домена FreeIPA при выполнении команды на сервере домена AD.

8.2.11.3. Инициализация доверительных отношений

Для инициализации доверительных отношений необходимо на сервере домена FreeIPA выполнить следующие действия:

1) получить полномочия администратора домена и проверить работоспособность служб FreeIPA, выполнив команды:

```
kinit <администратор_домена_FreeIPA>
id <администратор_домена_FreeIPA>
getent passwd <администратор_домена_FreeIPA>
```

В результате выполнения команд не должны быть выявлены ошибки;

2) запустить службу доверительных отношений FreeIPA командой:

```
sudo ipa-adtrust-install
```

На все вопросы ответить «Да» («y») и затем вести пароль администратора домена FreeIPA. Проверить правильность автоматического определения имени домена и ответить «Да» («y»);

3) настроить и проверить перенаправление DNS. Добавление зоны перенаправления осуществляется командой:

```
ipa dnsforwardzone-add <домен_AD> --forwarder=WIN_IP ?forward-policy=only
```

Проверка успешного выполнения команды выполняется путем:

а) проверки доступности сервер домена AD:

```
ping -c 3 <сервер_домена_AD>.<домен_AD>
```

б) проверки доступности службы FreeIPA:

```
dig SRV _ldap._tcp.<домен_FreeIPA>
```

в) проверки доступности службы домена AD:

```
dig SRV _ldap._tcp.<домен_AD>
```

4) сохранить конфигурацию Samba, выполнив команду:

```
cp /etc/samba/smb.conf /etc/samba/smb.conf && sudo testparm \
    | sudo tee /etc/samba/smb.conf > /dev/null
```

5) проверить работоспособность службы Samba командой:

```
smbclient -k -L <сервер_домена_FreeIPA>.<домен_FreeIPA>
```

6) установить доверительные отношения между доменами:

а) одностороннее доверительное отношение — одностороннее доверие к домену AD, при котором область FreeIPA доверяет лесу доменов AD, используя механизм доверительных отношений между деревьями доменов AD, но дерево доменов AD не доверяет области FreeIPA. Пользователи дерева доменов AD получают доступ к ресурсам области FreeIPA. Устанавливается командой:

```
ipa trust-add --type=ad <домен_AD> --admin \
     <администратор_домена_AD> --password
```

б) двустороннее доверительное отношение устанавливается командой:

```
ipa trust-add --type=ad <домен_AD> --admin \
     <администратор_домена_AD> --password --two-way=true
```

в) внешнее доверительное отношение — отношение доверия между доменами AD, находящимися в разных лесах доменов AD. Установление доверительных отношений между лесами доменов всегда требует установления доверительных отношений между корневыми доменами этих лесов, однако, внешнее доверительное отношение может быть установлено между любыми доменами в лесу. Применяется для установления доверительных отношений с конкретными доменами и не переходит границы доверенного домена. Устанавливается командой:

```
ipa trust-add --type=ad <домен_AD> --admin \
     <администратор_домена_AD> --password --two-way=true --external
```

7) после установления доверительных отношений следует выполнить команду для получения списка доверенных доменов:

```
ipa trust-fetch-domains <домен_AD>
```

Домен должен быть найден при выполнении команды:

```
ipa trustdomain-find <домен_AD>
```

8) для работы пользователей домена AD в домене FreeIPA следует зарегистрировать данных пользователей, добавив соответствующие группы и пользователей в них:

```
ipa group-add --desc='ad domain external map' ad_admins_external \
     --external
ipa group-add --desc='ad domain users' ad_admins
ipa group-add-member ad_admins_external --external \
     '<домен_AD>\Domain Admins'
ipa group-add-member ad_admins --groups ad_admins_external
```

На запросы «member_user» и «member_group» нажать клавишу **<Enter>**;

9) для предоставления пользователям прав доступа к разделяемым ресурсам требуется указать их идентификаторы безопасности. Для получение идентификатора безопасности пользователей домена AD на сервере AD из оболочки CMD (но не из оболочки PowerShell) выполнить команду:

```
c:\> wmic useraccount get name,sid
```

Для получение идентификатора безопасности пользователей домена FreeIPA на сервере FreeIPA выполнить команду:

```
ipa group-show ad_admins_external --raw
```

Для добавления разделяемого каталога `/share_dir`, который будет доступен для пользователей домена AD под именем `share_name` выполнить:

```
sudo mkdir /share_dir
sudo net conf setparm 'share_name' 'comment' 'Trust test share'
sudo net conf setparm 'share_name' 'read only' 'no'
sudo net conf setparm 'share_name' 'valid users' "$d_admins_sid"
sudo net conf setparm 'share_name' 'path' '/share_dir'
```

Проверить, что ресурс добавлен, выполнив команду:

```
smbclient -k -L <сервер_домена_FreeIPA>.<домен_FreeIPA>
```

После добавления каталога проверить доступность ресурса с сервера AD через веб-браузер.

8.2.11.4. Проверка установки доверительных отношений

При успешной установке доверительных отношений пользователи домена AD должны получить возможность входа в систему с использованием своего имени и пароля:

- через терминал;
- через графический интерфейс;
- через SSH (если установлена соответствующая сетевая служба).

Также пользователям AD предоставляется возможность доступа к разделяемым ресурсам.

ВНИМАНИЕ! Для входа необходимо использовать полное имя пользователя с указанием домена, к которому пользователь относится, например, `Administrator@windomain.ad`, при этом имя домена пишется строчными буквами, а имя пользователя с сохранением строчных и заглавных букв.

Проверка настройки DNS на сервере домена AD выполняется из командной строки. Для просмотра записей выполнить команду:

```
c:\>nslookup.exe
```

В выводе выполнения команды будут приведены записи о работе служб и служб домена:

1) записи, отвечающие за работу служб Kerberos через UDP и LDAP через TCP:

```
> set type=SRV
> _kerberos._udp.<домен_FreeIPA>.
_kerberos._udp.<домен_FreeIPA>.       SRV service location:
priority                = 0
weight                  = 100
port                    = 88
svr hostname   = <сервер_домена_FreeIPA>.<домен_FreeIPA>.
> _ldap._tcp.<домен_FreeIPA>.
_ldap._tcp.<домен_FreeIPA>       SRV service location:
priority                = 0
weight                  = 100
port                    = 389
svr hostname   = <сервер_домена_FreeIPA>.<домен_FreeIPA>.
```

2) записи, отвечающие за имя Kerberos realm домена FreeIPA:

```
> set type=TXT
_kerberos.<домен_FreeIPA>.
_kerberos.<домен_FreeIPA>.        Text =
        "<домен_FreeIPA>"
```

3) после выполнения команды `ipa-adtrust-install` должны появиться записи, отвечающие за работу служб MS DC Kerberos через UDP и LDAP через TCP:

```
> set type=SRV
> _kerberos._udp.dc._msdcs.<домен_FreeIPA>.
_kerberos._udp.dc._msdcs.<домен_FreeIPA>.        SRV service location:
priority                = 0
weight                  = 100
port                    = 88
svr hostname = <сервер_домена_FreeIPA>.<домен_FreeIPA>.
> _ldap._tcp.dc._msdcs.<домен_FreeIPA>.
_ldap._tcp.dc._msdcs.<домен_FreeIPA>.        SRV service location:
priority                = 0
weight                  = 100
port                    = 389
svr hostname = <сервер_домена_FreeIPA>.<домен_FreeIPA>.
```

Проверка наличия записей для работы служб AD на DNS-сервере AD выполняется из командной строки. Для просмотра записей выполнить команду:

```
c:\>nslookup.exe
```

Записи, отвечающие за работу служб Kerberos через UDP и LDAP через TCP:

```
> set type=SRV
> _kerberos._udp.dc._msdcs.<домен_AD>.
_kerberos._udp.dc._msdcs.<домен_AD>.  SRV service location:
priority = 0
weight = 100
port = 88
svr hostname = <сервер_домена_AD>.<домен_AD>.
> _ldap._tcp.dc._msdcs.<домен_AD>.
_ldap._tcp.dc._msdcs.<домен_AD>.  SRV service location:
priority = 0
weight = 100
port = 389
svr hostname = <сервер_домена_AD>.<домен_AD>.
```

Проверка настройки DNS на сервере домена FreeIPA и наличия записей для работы служб FreeIPA на DNS-сервере FreeIPA выполняется из командной строки.

Для просмотра записи, отвечающей за работу службы Kerberos через UDP, выполнить команду:

```
dig +short -t SRV _kerberos._udp.<домен_FreeIPA>.
```

Запись выводится в следующем виде:

```
0 100 88 <сервер_домена_FreeIPA>.<домен_FreeIPA>.
```

Для просмотра записи, отвечающей за работу службы LDAP через TCP, выполнить команду:

```
dig +short -t SRV _ldap._tcp.<домен_FreeIPA>.
```

Запись выводится в следующем виде:

```
0 100 389 <сервер_домена_FreeIPA>.<домен_FreeIPA>.
```

Для просмотра записи, отвечающей за имя Kerberos realm домена FreeIPA, выполнить команду:

```
dig +short -t TXT _kerberos.<домен_FreeIPA>.
```

Запись выводится в следующем виде:

```
"<домен_FreeIPA>"
```

После выполнения команды `ipa-adtrust-install` должны появиться записи, отвечающие за работу служб MS DC Kerberos через UDP и LDAP через TCP:

```
# dig +short -t SRV _kerberos._udp.dc._msdcs.<домен_FreeIPA>.
0 100 88 <сервер_домена_FreeIPA>.<домен_FreeIPA>.

# dig +short -t SRV _ldap._tcp.dc._msdcs.<домен_FreeIPA>.
0 100 389 <сервер_домена_FreeIPA>.<домен_FreeIPA>.
```

Проверка наличия записей для работы служб AD на DNS-сервере FreeIPA выполняется из командной строки.

Записи, отвечающие за работу служб Kerberos через UDP и LDAP через TCP:

```
# dig +short -t SRV _kerberos._udp.dc._msdcs.<домен_AD>.
0 100 88 <сервер_домена_AD>.<домен_AD>.
```

```
# dig +short -t SRV _ldap._tcp.dc._msdcs.<домен_AD>.
0 100 389 <сервер_домена_AD>.<домен_AD>.
```

Если запись `_kerberos._udp.dc._msdcs.source-<домен_AD>.` недоступна, то необходимо проверить `_kerberos._tcp.dc._msdcs.source-<домен_AD>.`

8.2.12. Создание самоподписанного сертификата

8.2.12.1. Создание сертификата с помощью инструмента XCA

Установка и настройка инструмента XCA выполняется в соответствии с 6.10.5.1.

Для создания цепочки сертификатов необходимо запустить инструмент XCA и выполнить следующие действия:

1) создать корневой сертификат:
   а) во вкладке «Закрытые ключи» нажать кнопку **[Новый ключ]**. В открывшемся окне в поле «Внутреннее имя» указать имя «rootKey» и нажать **[Создать]**;
   б) перейти во вкладку «Сертификаты», нажать **[Новый сертификат]**;
   в) в открывшемся окне «Создать сертификат x509» перейти во вкладку «Субъект»:
      - в поле «Внутреннее имя» указать имя сертификата «rootCA»;
      - в поле «commonName» указать то же имя — «rootCA»;
      - в блоке «Закрытый ключ» выбрать ранее созданный ключ «rootKey»;
   г) в окне «Создать сертификат x509» перейти во вкладку «Расширения»:
      - в поле «Тип» выбрать «Центр Сертификации»;
      - определить период действия сертификата, указав в блоке «Выбор периода» значение «10»;
      - нажать кнопку **[Применить]**, затем нажать **[Да]**.
2) создать сертификат для сервера:
   а) в основном окне программы перейти во вкладку «Закрытые ключи» и нажать кнопку «Новый ключ»;
   б) в открывшемся окне в поле «Внутреннее имя» указать имя «serverKey» и нажать **[Создать]**;
   в) перейти во вкладку «Сертификаты», нажать **[Новый сертификат]**;
   г) в открывшемся окне «Создать сертификат x509» во вкладке «Первоисточник»:
      - в блоке «Подписание» установить флаг «Использовать этот сертификат для подписи» и выбрать значение «rootCA» (имя корневого сертификата);
      - в поле «Алгоритм подписи» указать «SHA 256»;

д) в окне «Создать сертификат x509» перейти во вкладку «Субъект»:

- в поле «Внутреннее имя» указать FQDN сервера, для которого формируется сертификат, например, `dc01.example.ru`;
- в поле «commonName» также указать FQDN сервера, для которого формируется сертификат;
- в блоке «Закрытый ключ» выбрать ранее созданный ключ «serverKey»;

е) в окне «Создать сертификат x509» перейти во вкладку «Расширения»:

- в поле «Тип» выбрать «Конечный субъект»;
- определить период действия сертификата, указав в блоке «Выбор периода» значение «10»;
- нажать кнопку **[Применить]**, затем нажать **[Да]**.

3) экспортировать сертификат сервера:

а) в основном окне программы перейти во вкладку «Сертификаты»;

б) выбрать требуемый сертификат сервера и нажать кнопку **[Экспорт]**;

в) в открывшемся окне указать имя файла контейнера сертификата и его расположение;

г) в блоке «Формат для экспорта» выбрать формат «PKCS12» и нажать кнопку **[Да]**;

д) задать пароль на экспортируемый контейнер и нажать кнопку **[Да]**.

На контроллере домена FreeIPA для указания контейнера с сертификатом выполнить команду `astra-freeipa-server` с параметрами `-l` и `-lp`:

```
astra-freeipa-server -l <путь_к_контейнеру> -lp <пароль_к_контейнеру>
```

Просмотреть перечень дополнительных параметров для запуска с командой `astra-freeipa-server` можно выполнив:

```
astra-freeipa-server --help
```

8.2.12.2. Создание сертификата с помощью инструмента командной строки

Инструмент командной строки `astra-freeipa-server-crt` автоматизирует выпуск сертификатов для серверов (реплик) FreeIPA и предназначен для автоматизации работы в системах, в которых не применяется DogTag, являющийся штатной системой управления сертификатами FreeIPA.

Установка инструмента командной строки `astra-freeipa-server-crt` выполняется автоматически при установке графической утилиты `fly-admin-freeipa-server` или инструмента командной строки `astra-freeipa-server` в соответствии с 8.2.4.

При инициализации домена FreeIPA в соответствии с 8.2.5 в каталоге `/etc/ssl/freeipa` первого контроллера домена автоматически создаются файлы, перечень которых приведен в таблице 41.

Таблица 41

Наименование, размещение файла	Описание
`/etc/ssl/freeipa/ca.key`	Закрытый ключ центра аутентификации
`/etc/ssl/freeipa/ca.crt`	Сертификат закрытого ключа центра аутентификации
`/etc/ssl/freeipa/server.key`	Закрытый ключ сервера
`/etc/ssl/freeipa/server.crt`	Сертификат закрытого ключа сервера

При первом запуске инструмента командной строки `astra-freeipa-server-crt` будет создан новый закрытый ключ сервера, который будет размещен в файле `/etc/ssl/freeipa/<имя_компьютера>.<имя_домена>.key`. Созданный закрытый ключ будет использоваться для выпуска и перевыпуска всех сертификатов.

ВНИМАНИЕ! Замена закрытых ключей посредством инструмента командной строки `astra-freeipa-server-crt` не поддерживается.

Кроме того, при запуске инструмента командной строки `astra-freeipa-server-crt` без указания параметров будет создан новый сертификат сервера. Выпущенный сертификат будет размещен в файле `/etc/ssl/freeipa/<имя_компьютера>.<имя_домена>-<дата_время>.crt`.

ВНИМАНИЕ! По умолчанию будут создаваться сертификаты для FreeIPA версии 4.6.x. Поэтому при запуске инструмента командной строки `astra-freeipa-server-crt` всегда необходимо указывать параметр `--48` (создавать сертификаты для FreeIPA версии 4.8.x).

Параметры инструмента командной строки `astra-freeipa-server` приведены в таблице 42.

Таблица 42

Параметр	Описание
`-h, --help`	Вывести справку по инструменту командной строки
`--certdir DIR`	Задать имя каталога (`DIR`) для поиска ключа и сертификата центра аутентификации и для размещения создаваемых сертификатов. Если каталог не существует — он будет создан. Значение по умолчанию `/etc/ssl/freeipa`
`--host FQDN`	Указать полное доменное имя (`FQDN`) сервера, для которого выпускается сертификат. Если имя не задано — используется `hostname` текущего сервера
`--cacrt FILE`	Указать имя файла (`FILE`) с существующим сертификатом центра аутентификации. Значение по умолчанию `/etc/ssl/freeipa/ca.crt`

Окончание таблицы 42

Параметр	Описание
`--cakey FILE`	Указать имя файла (`FILE`) с существующим закрытым ключом центра аутентификации. Значение по умолчанию `/etc/ssl/freeipa/ca.key`
`--sekey FILE`	Указать имя файла (`FILE`) с закрытым ключом сервера. Если файл не существует — будет создан новый закрытый ключ. Если имя не задано — ключ будет размещен в файле с именем `FQDN.key` в каталоге для размещения сертификатов
`--sekey_parm ALG`	Указать через двоеточие алгоритм и длину закрытого ключа сервера (`ALG`). Значение по умолчанию `rsa:2048`
`--secrt_days NUM`	Указать в днях срок действия (`NUM`) выпускаемого сертификата. Значение по умолчанию `365 дней`
`--export`	Экспортировать сертификат в контейнер формата `pkcs12` для установки нового сервера (новой реплики) FreeIPA. Экспорт будет выполнен в файл с именем `FQDN-<дата_время>.p12` в каталоге для размещения сертификатов. Не требуется для обновления сертификата уже установленного сервера
`--pin PIN`	Пароль (`PIN`) для экспорта сертификата. Чтобы задать пустой пароль, указать пробел в кавычках `--pin " "`. Если пароль не задан — он будет запрошен в процессе выполнения команды
`--push ADMIN`	Скопировать через `ssh/scp` созданные файлы на сервер, указанный в параметре `--host`, и зарегистрировать их. В параметре `ADMIN` можно задать не только имя пользователя, но и адрес целевого сервера, например `admin@192.168.32.11`. Все действия будут выполняться от имени `ADMIN`. Все файлы будут копироваться в домашний каталог этого пользователя. Если выполнялся экспорт сертификата для нового сервера (новой реплики), то сертификат будет скопирован в файл с именем `FQDN.p12`. Если создавался новый сертификат для существующего сервера, то: - копия этого сертификата будет скопирована в файл с именем `FQDN.crt`; - будет сделана попытка зарегистрировать его в БД сертификатов `/etc/apache2/nssdb`. **ВНИМАНИЕ!** После регистрации в БД сертификатов нового сертификата службы FreeIPA должны быть перезапущены вручную
`--46`	Создать сертификаты для FreeIPA версии 4.6.x. Данный параметр используется по умолчанию
`--48`	Создать сертификаты для FreeIPA версии 4.8.x
`-y`	Выполнить действия без запроса подтверждения

Пример использования инструмента командной строки `astra-freeipa-server-crt` для создания реплики в домене FreeIPA представлен в 8.2.10.

В случае необходимости выпуска новых сертификатов, например при истечении срока действия, можно воспользоваться следующей командой:

```
astra-freeipa-server-ctr --host <имя_компьютера>.<имя_домена> \
     --push <имя_локального_администратора>
```

8.2.13. Настройка веб-сервера Apache2 для работы в домене FreeIPA

Настройка работы веб-сервера Apache2 в домене FreeIPA осуществляется:

- для выполнения аутентификации с использованием Kerberos;
- для обеспечения сквозной аутентификации приложений.

Для развертывания веб-сервера Apache2 требуется установить пакет `apache2` на компьютере, предназначенном для выполнения роли веб-сервера. Установка выполняется командой:

```
sudo apt install apache2
```

Для обеспечения совместной работы веб-сервера Apache2 в домене FreeIPA требуется:

1) настроенный домен FreeIPA, например `ipadomain0.ru`, с настроенной службой разрешения имен (DNS);
2) отдельный компьютер для размещения веб-сервера Apache2;
   а) веб-сервер должен быть введен в домен FreeIPA в соответствии с 8.2.6;
   б) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось как полное имя веб-сервера (FQDN), например `web.ipadomain0.ru`;
   в) веб-серверу должен быть назначен постоянный IP-адрес.

8.2.13.1. Настройка аутентификации Kerberos

Для настройки аутентификации Kerberos требуется дополнительно установить модуль аутентификации Kerberos `libapache2-mod-auth-gssapi`. Установка выполняется командой:

```
sudo apt install libapache2-mod-auth-gssapi
```

Если в ОС установлен в соответствии с 11.3 модуль аутентификации через PAM веб-сервера Apache2 `authnz_pam`, отключить его при помощи команды:

```
a2dismod authnz_pam
```

и активировать модуль веб-сервера Apache2 `auth_gssapi` при помощи команды:

```
a2enmod auth_gssapi
```

Далее веб-сервер необходимо зарегистрировать как доменную службу — либо с помощью веб-интерфейса администратора FreeIPA, либо получив билет Kerberos администратора домена и выполнив команду `ipa service-add` (команду следует выполнить либо на контроллере домена, либо на веб-сервере):

```
kinit admin
ipa service-add HTTP/web.ipadomain0.ru
```

где `web.ipadomain0.ru` — полное доменное имя компьютера, на котором будет развернута служба.

Затем на веб-сервере выгрузить таблицу ключей для зарегистрированной службы:

```
sudo kinit admin
sudo ipa-getkeytab -p HTTP/web.ipadomain0.ru@IPADOMAIN0.RU
-k /etc/ipa/apache2.keytab
```

Параметр `-k` команды `ipa-getkeytab` задает имя файла, в который будет сохранена таблица ключей (`/etc/ipa/apache2.keytab`).

Для выгруженного файла с таблицей ключей задать права доступа, выполнив команды:

```
chown www-data /etc/ipa/apache2.keytab
chmod 600 /etc/ipa/apache2.keytab
```

Примечания:

1. Для получения ключей не требуется механизм `sudo` — достаточно билета Kerberos. Механизм `sudo` используется для записи таблицы ключей в каталог `/etc/ipa/`. При этом билет Kerberos также должен быть получен с помощью механизма `sudo`, так так полученный от имени обычного пользователя билет будет недействителен.
2. Команду получения таблицы ключей `ipa-getkeytab` можно выполнить на контроллере домена — в этом случае полученную таблицу ключей необходимо защитить от несанкционированного доступа и скопировать в соответствующий каталог на веб-сервере.

Далее требуется на веб-сервере создать конфигурационный файл аутентификации для областей, требующих аутентификации, например файл `/etc/apache2/conf-available/kerberos-auth.conf` со следующими строками:

```
<Directory /var/www>
# тип аутентификации
AuthType GSSAPI
# Подсказка с информацией о ресурсе (выводится при запросе пароля)
AuthName "Astra Kerberos protected area"
GssapiCredStore keytab:<путь_к_таблице_ключей>
# Включить 3 нижних параметра, если нужно кешировать сессии
#GssapiUseSessions On
#Session On
#SessionCookieName myapp_web_gssapi_session path=/my_url;httponly;secure;
Require valid-user
</Directory>
```

Созданный конфигурационный файл аутентификации необходимо указать в конфигурационных файлах виртуальных web-сайтов, размещаемых в каталоге `/etc/apache2/sites-available/`, с помощью директивы `Include`. Например, для конфигурационного файла `000-default.conf` виртуального web-сайта, устанавливаемого по умолчанию:

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

Include conf-available/kerberos-auth.conf
</VirtualHost>
```

Аналогично конфигурационный файл аутентификации `kerberos-auth.conf` можно включить в конфигурационный файл сайта `/etc/apache2/sites-enabled/default-ssl.conf`.

Для использования аутентификации Kerberos необходимо, чтобы веб-браузер пользователя поддерживал метод аутентификации `negotiate`.

Для включения аутентификации `negotiate` в веб-браузере Mozilla Firefox необходимо:

1) в адресной строке веб-браузера ввести:

```
about:config
```

2) для параметра network.negotiate-auth.trusted-uris задать маски доменов, для которых будет использоваться аутентификация negotiate. В общем случае в качестве значения можно указать http://, https://;

3) если необходимо обеспечить сквозную аутентификацию из сценариев при работе с другими службами, например с сервером СУБД, в веб-браузере Mozilla Firefox для параметра network.negotiate-auth.delegation-uris следует задать маски доменов, которым можно передавать данные для сквозной аутентификации. При этом в запускаемых сценариях следует выставить переменную окружения KRB5CCNAME. Например, для сценариев на языке PHP:

```
putenv("KRB5CCNAME=".$_SERVER[?KRB5CCNAME?]);
```

8.2.13.2. Настройка защищенных соединений SSL с использованием сертификатов

При установке веб-сервера Apache2 для защищенных соединений SSL по умолчанию используется предустановленный закрытый ключ /etc/ssl/private/ssl-cert-snakeoil.key и соответствующий ему сертификат /etc/ssl/certs/ssl-cert-snakeoil.pem. Данные ключ и сертификат следует заменить на ключ и сертификат, выданные центром аутентификации согласно 8.2.12.1.

Созданные центром аутентификации сертификат и ключ, например apache.crt и apache.key, следует сохранить в каталоге /etc/ipa/.

Расположение сертификатов необходимо указать в конфигурационных файлах web-сайтов, поддерживающих соединения SSL. Например, в конфигурационном файле etc/apache2/sites-enabled/default-ssl.conf web-сайта, устанавливаемого по умолчанию:

```
SSLCertificateFile /etc/ipa/apache.crt
SSLCertificateKeyFile /etc/ipa/apache.key
```

Для начала работы с использованием SSL необходимо:

1) загрузить модуль работы по протоколу SSL, выполнив команду:

```
sudo a2enmod ssl
```

2) включить web-сайт, для которого настраивается работа по протоколу SSL. Например, для включения устанавливаемого по умолчанию web-сайта default-ssl выполнить команду:

```
sudo a2ensite default-ssl
```

3) обновить конфигурацию веб-сервера, выполнив команду:

```
sudo systemctl reload apache2
```

8.2.13.3. Настройка каталогов для работы с конфиденциальными данными

При необходимости возможно настроить каталоги для работы с конфиденциальными данными. Для этого следует:

1) на веб-сервере назначить мандатные атрибуты каталогам с виртуальными серверами:

```
sudo pdpl-file 3:0:-1:CCNR /var/www/
sudo pdpl-file 3:0:-1:CCNR /var/www/html/
```

2) перезапустить веб-сервер:

```
sudo systemctl restart apache2
```

8.2.14. Веб-интерфейс FreeIPA

Использование веб-интерфейса возможно после запуска FreeIPA согласно 8.2.5.1 или 8.2.5.2. Для входа в веб-интерфейс ввести в адресной строке браузера ссылку, предоставленную при запуске FreeIPA. В случае если при первом входе в веб-интерфейс появится сообщение о том, что соединение не защищено, следует добавить данный адрес в исключения.

Для входа в веб-интерфейс используется имя учетной записи `admin` и пароль, заданный при запуске FreeIPA (см. 8.2.5.1 и 8.2.5.2).

8.2.14.1. Установка мандатных атрибутов (user mac)

Для установки мандатных атрибутов пользователя необходимо:

1) выбрать пользователя и перейти во вкладку «Параметры»;
2) используя раскрывающиеся списки «Min MAC», «Max MAC» и «Уровень целостности» задать мандатные атрибуты;
3) для установки мандатных атрибутов нажать **[Сохранить]**.

Поле «Мандатный атрибут» должно принять заданное значение в соответствии с рис. 4.

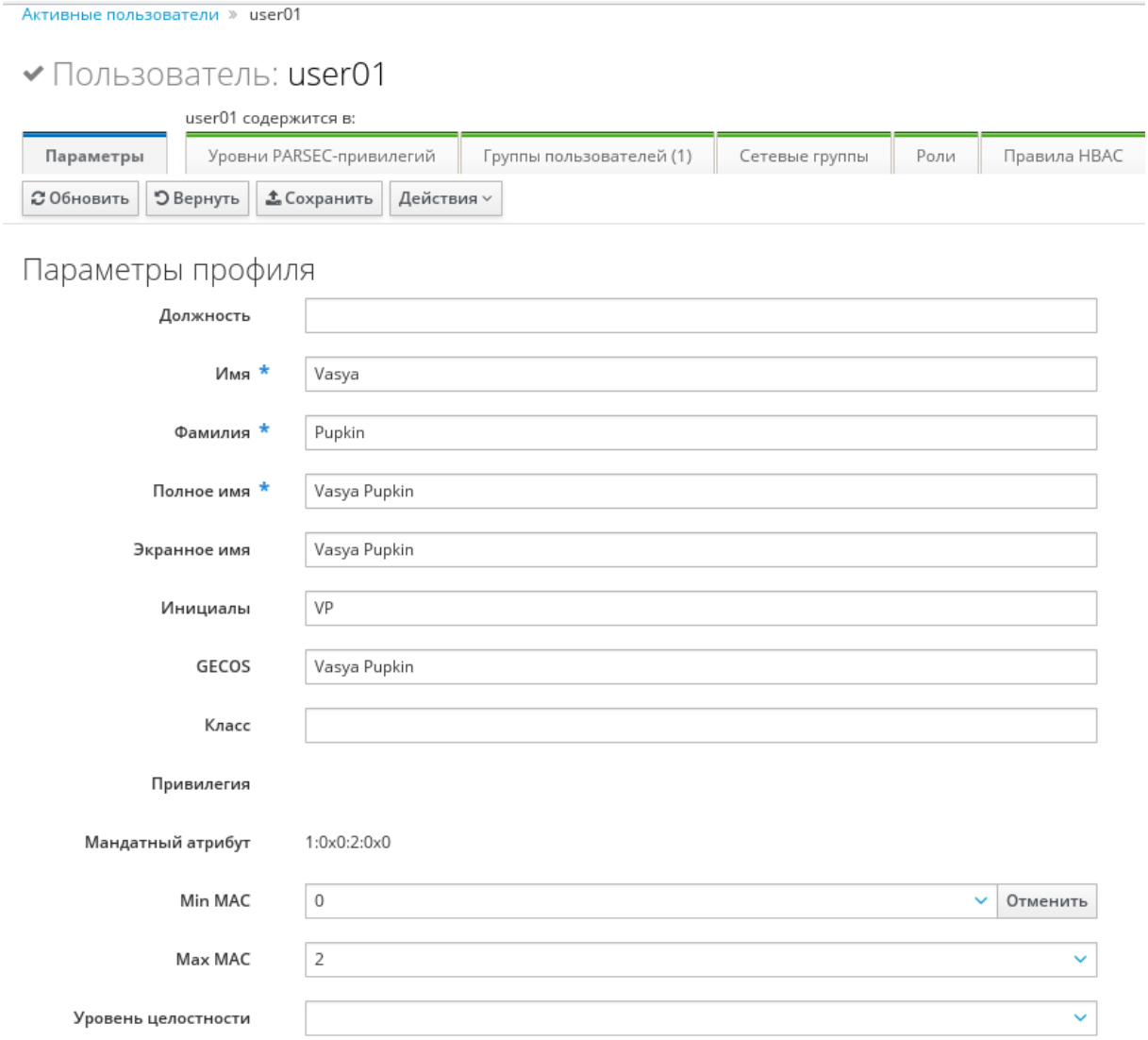

Рис. 4

8.2.14.2. Установка привилегий PARSEC (parsec cap)

Для установки привилегий PARSEC необходимо:

1) выбрать пользователя и перейти во вкладку «Уровни PARSEC-привилегий»;
2) нажать **[Добавить]**;
3) в открывшемся окне в блоке «Доступен» отметить требуемые привилегии;
4) переместить отмеченные привилегии в блок «Ожидаемый», нажав кнопку **[>]**, затем нажать **[Добавить]** (см. рис. 5).

Поле «Мандатный атрибут» должно принять заданное значение.

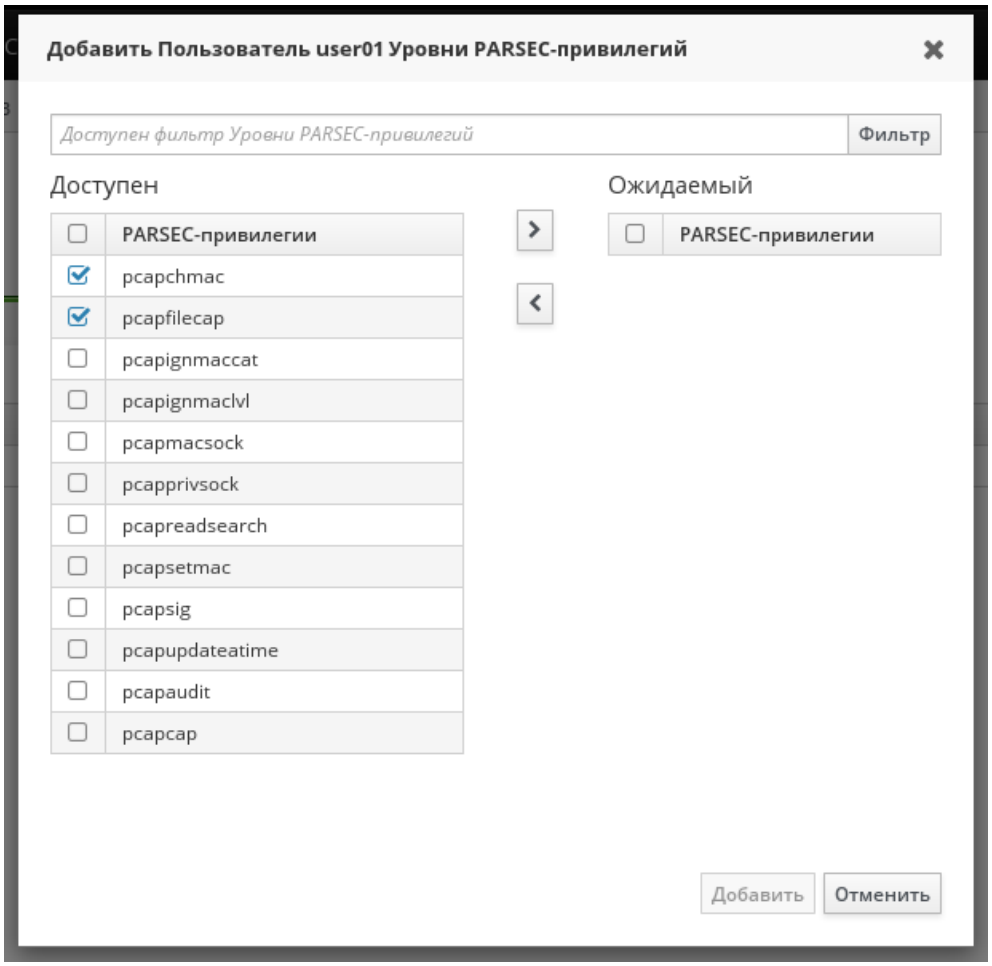

Рис. 5

Описание PARSEC-привилегий приведено в документе РУСБ.10015-01 97 01-1.

8.2.15. Удаление контроллера домена

Для удаления контроллера домена с помощью инструмента командной строки `astra-freeipa-server` выполнить команду:

```
astra-freeipa-server -U
```

8.3. Samba

В состав ОС входит пакет программ Samba, предназначенный для решения задач совместимости со средой Microsoft Active Directory.

Samba позволяет ОС выступать как в роли контроллера домена AD, так и в роли клиента домена.

Возможности Samba:

- служба аутентификации на базе Kerberos;
- LDAP-совместимая служба каталогов с поддержкой репликации;
- поддержка групповых политик;
- поддержка доверительных отношений;
- DNS-сервер на базе BIND или собственной реализации.

В состав ОС входят консольные и графические средства, позволяющие инициализировать AD домен или подключиться к уже существующему.

Актуальные инструкции для разных сценариев применения приведены на официальном сайте `wiki.astralinux.ru`.

8.3.1. Настройка контроллера домена

В состав ОС входит инструмент командной строки `astra-sambadc`, включающий сценарии автоматизированной настройки и построения нового контроллера домена или включения в существующий домен в роли контроллера домена.

Для установки инструмента выполнить команду:

```
apt install astra-sambadc
```

При выполнении команды также будут установлены необходимые для работы домена AD пакеты `samba`, `winbind` и `ntp`.

Для создания нового домена в дополнение к инструменту `astra-sambadc` и автоматически устанавливаемым пакетам следует установить пакет `krb5-kdc`:

```
apt install krb5-kdc
```

Для создания нового домена используется команда:

```
astra-sambadc -d <имя_домена> -px
```

Данные, необходимые для создания домена и не указанные при выполнении команды, будут запрошены в интерактивном режиме.

Дополнительная информация по использованию команды доступна при выполнении команды с параметром `-h`:

```
astra-sambadc -h
```

Для настройки и построения нового контроллера домена или включения в существующий домен в роли контроллера домена в графическом режиме используется утилита `fly-admin-ad-server`.

Для установки графической утилиты выполнить команду:

```
apt install fly-admin-ad-server
```

Описание графической утилиты приведено в электронной справке.

8.3.2. Настройка участников домена

В состав ОС входит инструмент командной строки `astra-winbind`, включающий сценарии автоматизированной настройки компьютера для ввода в существующий домен.

ВНИМАНИЕ! Перед вводом компьютера в домен необходимо настроить на этом компьютере службу разрешения имен (DNS) так, чтобы в качестве сервера DNS использовался сервер DNS домена. Если этого не сделать, то контроллер домена не будет обнаружен.

Для ввода компьютера в домен используется команда:

```
astra-winbind -dc <имя_домена> -u <имя_администратора_домена> -px
```

Данные, необходимые для ввода в домен и не указанные при выполнении команды, будут запрошены в интерактивном режиме.

Дополнительная информация по использованию команды доступна при выполнении команды с параметром `-h`:

```
astra-winbind -h
```

Для ввода компьютера в существующий домен в графическом режиме используется утилита `fly-admin-ad-client`. Описание графической утилиты приведено в электронной справке.

Для проверки успешности присоединения к домену можно использовать команду:

```
net ads testjoin -k
```

8.4. Настройка сетевых служб

Ряд сетевых служб, таких как СУБД, электронная почта, обработка гипертекстовых документов (web), система печати и др. для работы в ЕПП должны быть соответствующим образом настроены. Как правило, настройка заключается в обеспечении возможности использования этими службами сквозной аутентификации по Kerberos и получения необходимой информации из БД LDAP.

Примечание. При выполнении настройки сетевых служб потребуется использование учетной записи привилегированного пользователя через механизм `sudo`. При снятии блокировки на интерактивный вход в систему для суперпользователя `root` не рекомендуется осуществлять переключение в режим суперпользователя командой `su`. Необходимо использовать команду:

```
# su -
```

ВНИМАНИЕ! Для обеспечения нормальной работы пользователя с сетевыми службами в ЕПП должна быть явно задана его классификационная метка (диапазон уровней конфиденциальности и категорий конфиденциальности) с помощью соответствующих утилит, даже если ему не доступны уровни и категории выше 0.

Описание настройки следующих сетевых служб приведены в соответствующих подразделах:

- система обмена сообщениями электронной почты описана в 16.4;
- защищенный комплекс программ гипертекстовой обработки данных описан в 11.4;
- защищенный комплекс программ печати и маркировки документов описан в 14.3.2.

Описание настройки СУБД приведено в документе РУСБ.10015-01 97 01-3.

9. ВИРТУАЛИЗАЦИЯ СРЕДЫ ИСПОЛНЕНИЯ

ОС поддерживает технологию виртуализации. Данная технология позволяет запускать множество виртуальных машин (ВМ), называемых гостевыми, на одной физической машине, называемой хостовой машиной. При этом гостевые операционные системы, установленные на каждой из гостевых машин, могут отличаться друг от друга и от операционной системы хостовой машины и являются полностью изолированными. Монитор виртуальных машин (гипервизор) обеспечивает параллельную работу гостевых операционных систем, их изоляцию, защиту, управление ресурсами и другие необходимые функции. Основными средствами, необходимыми для создания среды виртуализации, являются:

- сервер виртуализации libvirt;
- программа эмуляции аппаратного обеспечения QEMU.

Описание защиты среды виртуализации приведено в РУСБ.10015-01 97 01-1.

9.1. Сервер виртуализации libvirt

Пакет сервера виртуализации состоит из службы сервера виртуализации `libvirtd`, предоставляющей возможность удаленного управления по сети с использованием различных протоколов и способов аутентификации, клиентской библиотеки `libvirt0`, командной оболочки `virsh` и ряда других утилит командной строки. Графический интерфейс управления виртуализацией обеспечивается пакетом `virt-manager`.

ВНИМАНИЕ! Все конфигурационные файлы или файлы сервера виртуализации libvirt, содержащие ключевую информацию Kerberos или PKI, не должны быть доступны пользователям.

Сервер виртуализации использует следующие каталоги хостовой файловой системы (ФС):

1) `/etc/libvirt/` — каталог конфигурации сервера виртуализации libvirt:
   а) `qemu/` — каталог конфигурационных XML-файлов виртуальных машин QEMU:
   - `network/` — каталог конфигурационных XML-файлов виртуальных сетей;
   - `*.xml` — конфигурационные XML-файлы виртуальных машин QEMU;

   б) `storage/` — каталог конфигурационных файлов пулов файлов-образов;

   в) `libvirt.conf` — клиентский конфигурационный файл сервера виртуализации libvirt;

   г) `libvirtd.conf` — конфигурационный файл службы сервера виртуализации `libvirtd` (см. 9.2);

   д) `qemu.conf` — конфигурационный файл QEMU (см. 9.6);
2) `/var/lib/libvirt/` — рабочий каталог сервера виртуализации libvirt:
   а) `images/` — каталог файлов-образов по умолчанию;

б) network/ — рабочий каталог виртуальных сетей;

в) qemu/ — рабочий каталог запущенных виртуальных машин QEMU:

- save/ — каталог сохраненных состояний виртуальных машин;

- snapshot — каталог снимков виртуальных машин;

г) runimages/ — каталог расположения копий файлов-образов виртуальных машин, запущенных в режиме запрета модификации файлов-образов;

3) /var/run/libvirt/ — каталог текущего рабочего состояния сервера виртуализации libvirt:

а) network/ — рабочий каталог запущенных виртуальных сетей;

б) qemu/ — каталог текущих конфигурационных xml-файлов запущенных виртуальных машин QEMU;

в) libvirt-sock — Unix-сокет для локальных соединений со службой сервера виртуализации libvirtd;

г) libvirt-sock-ro — Unix-сокет, доступный только для чтения, для локальных соединений со службой сервера виртуализации libvirtd.

9.2. Служба сервера виртуализации libvirtd

Служба сервера виртуализации libvirtd предоставляет возможность удаленного управления сервером виртуализации по сети с использованием различных протоколов и способов аутентификации. При этом поддерживается возможность решения всех задач по созданию и учету виртуальных машин, настройке их конфигурации и непосредственно запуска.

Доступ к службе сервера виртуализации возможен как с помощью локальных Unix-сокетов, так и по сети с помощью консольных или графических инструментов управления виртуальными машинами.

Основным конфигурационным файлом службы сервера виртуализации является /etc/libvirt/libvirtd.conf. Он содержит описание необходимых для работы службы настроек и параметров. Файл разбит на секции, описывающие параметры функционирования службы сервера виртуализации: интерфейсы взаимодействия и права доступа к ним, способы и параметры аутентификации, политику разграничения доступа, состав выводимой в журнал информации и т. п. Основные параметры, указываемые в файле, приведены в таблице 43.

Таблица 43 – Параметры конфигурационного файла /etc/libvirt/libvirtd.conf

Параметр	Описание
listen_tls	Принимать TLS-соединения с использованием сертификатов

Продолжение таблицы 43

Параметр	Описание
listen_tcp	Принимать TCP-соединения **ВНИМАНИЕ!** Для применения данной настройки дополнительно необходимо указать значение -l для параметра libvirtd_opts в конфигурационном файле /etc/default/libvirtd
listen_addr	Адрес сетевого интерфейса для приема соединений
tls_port	Порт для сетевых соединений TLS
tcp_port	Порт для сетевых соединений TCP
auth_tcp	Используемая для TCP-соединений аутентификация. Параметр должен содержать значение "sasl" (см. 9.3).
admin_group	Группа администраторов сервера виртуализации (значение по умолчанию "libvirt-admin"). Участие в данной группе позволяет администрировать ВМ. Если параметр закомментирован, то доступ к ВМ, в т.ч. для администрирования, будут иметь все пользователи
admvm_group	Группа администраторов виртуальных машин (значение по умолчанию "libvirt-admvm"). Участие в данной группе позволяет получать доступ к ВМ или выполнять ее администрирование в соответствии с ACL. При применении драйвера доступа "polkit" не рекомендуется изменение значения данного параметра
devel_group	Группа разработчиков виртуальных машин (значение по умолчанию "libvirt-dev"). При применении драйвера доступа "polkit" не рекомендуется измененять значение данного параметра
access_drivers	Применяемый драйвер доступа к серверу виртуализации. Параметр должен содержать значение ["parsec"] для реализации мандатного и дискреционного управления доступом и значение ["polkit "parsec"] для реализации мандатного, дискреционного и ролевого управления доступом
integrity_control	Применение механизма контроля целостности на основании подсчета контрольных сумм конфигураций («отпечатка конфигурации»). Для включения механизма необходимо установить значение 1. Включение рекомендуется осуществлять после завершения всех настроек системы виртуализации

Продолжение таблицы 43

Параметр	Описание
integrity_image_control	Применение механизма контроля целостности к файлам образов ВМ. Для включения механизма необходимо задать значение 1. Применяется только при включенном механизме контроля целостности на основании подсчета контрольных сумм конфигураций (integrity_control = 1)
hash_type	Алгоритм вычисления контрольной суммы (хеша) образов ВМ
ilev_vm	Категория целостности, присваиваемая ВМ. Значение по умолчанию 63
memory_integrity_check_period_s	Применение механизма контроля целостности к установленным на контроль областям памяти ВМ. Для включения механизма необходимо задать значение периода проверки (в секундах). При выявлении нарушения целостности областей памяти ВМ выполняется регистрация фактов нарушения целостности объектов контроля
memory_integrity_check_shutdown_domain	Применение режима принудительного выключения ВМ в случае нарушения целостности установленных на контроль областей памяти ВМ. Для включения режима необходимо задать значение 1. При выявлении нарушения целостности областей памяти ВМ помимо регистрации фактов нарушения будет выполняться принудительное выключение ВМ
file_integrity_check_period_s	Применение механизма контроля целостности к установленным на контроль файлам гостевой операционной системы в процессе ее функционирования. Для включения механизма необходимо задать значение периода проверки (в секундах). При выявлении нарушения целостности файлов гостевой операционной системы выполняется регистрация фактов нарушения целостности объектов контроля
file_integrity_check_shutdown_domain	Применение режима принудительного выключения ВМ в случае нарушения целостности установленных на контроль файлов гостевой операционной системы. Для включения режима необходимо задать значение 1. При выявлении нарушения целостности файлов гостевой операционной системы помимо регистрации фактов нарушения будет выполняться принудительное выключение ВМ

Окончание таблицы 43

Параметр	Описание
`file_integrity_on_startup_VM`	Применение механизма контроля целостности файлов гостевой операционной системы при запуске ВМ. Для включения механизма необходимо задать значение `1`. При выявлении нарушения целостности файлов гостевой операционной системы обеспечивается блокировка запуска ВМ и регистрация фактов нарушения целостности объектов контроля. Использование механизма возможно при условии применения ОС в качестве гостевой операционной системы

Примечание. Конфигурационные параметры TLS для доступа к серверу виртуализации libvirt рассматриваются в 9.7.

9.3. Конфигурационные файлы сервера виртуализации

При использовании механизмов SALS для доступа к серверу виртуализации libvirt или к рабочим столам виртуальных машин через систему VNC или по протоколу SPICE необходимо наличие соответствующих конфигурационных файлов с параметрами SASL в каталоге `/etc/sasl2`. Для сервера виртуализации требуется файл `libvirt.conf`, для QEMU (VNC и SPICE) — `qemu.conf`.

Описание основных параметров конфигурационного файла SASL `/etc/sasl2/libvirt.conf` приведено в таблице 44.

Таблица 44

Параметр	Описание
`mech_list`	Список механизмов SASL
`keytab`	Путь к файлу ключевой информации Kerberos. Параметр необходим при использовании в ЕПП. Должен содержать корректные значения для файлов, содержащих ключевую информацию для VNC и SPICE
`sasldb_path`	Путь к базе данных SASL. При использовании в ЕПП не применяется и должен быть закомментирован.

ВНИМАНИЕ! Файлы ключевой информации Kerberos для VNC и SPICE должны быть доступны на чтение пользователям, запускающим виртуальные машины, и группе `libvirt-qemu`.

Для VNC и SPICE могут быть заданы другие пути расположения конфигурационного файла SASL. Описание конфигурационного файла `qemu.conf` приведено в 9.6.

9.4. Консольный интерфейс virsh

В состав пакетов сервера виртуализации libvirt входит консольный интерфейс управления виртуальными машинами `virsh`, позволяющий в консоли с помощью командной оболочки производить действия по управлению конфигурацией виртуальных машин.

Командная оболочка содержит набор команд по управлению виртуальными машинами, файлами-образами носителей, виртуальными интерфейсами и сетями и позволяет править конфигурационные файлы виртуальных машин.

Более подробно возможности консольного интерфейса управления виртуальными машинами `virsh` описаны в соответствующем руководстве `man`.

9.5. Графическая утилита virt-manager

Графическая утилита управления виртуальными машинами `virt-manager` предоставляет доступ к возможностям сервера виртуализации libvirt из графического интерфейса пользователя. Внешний вид окна утилиты приведен на рис. 6.

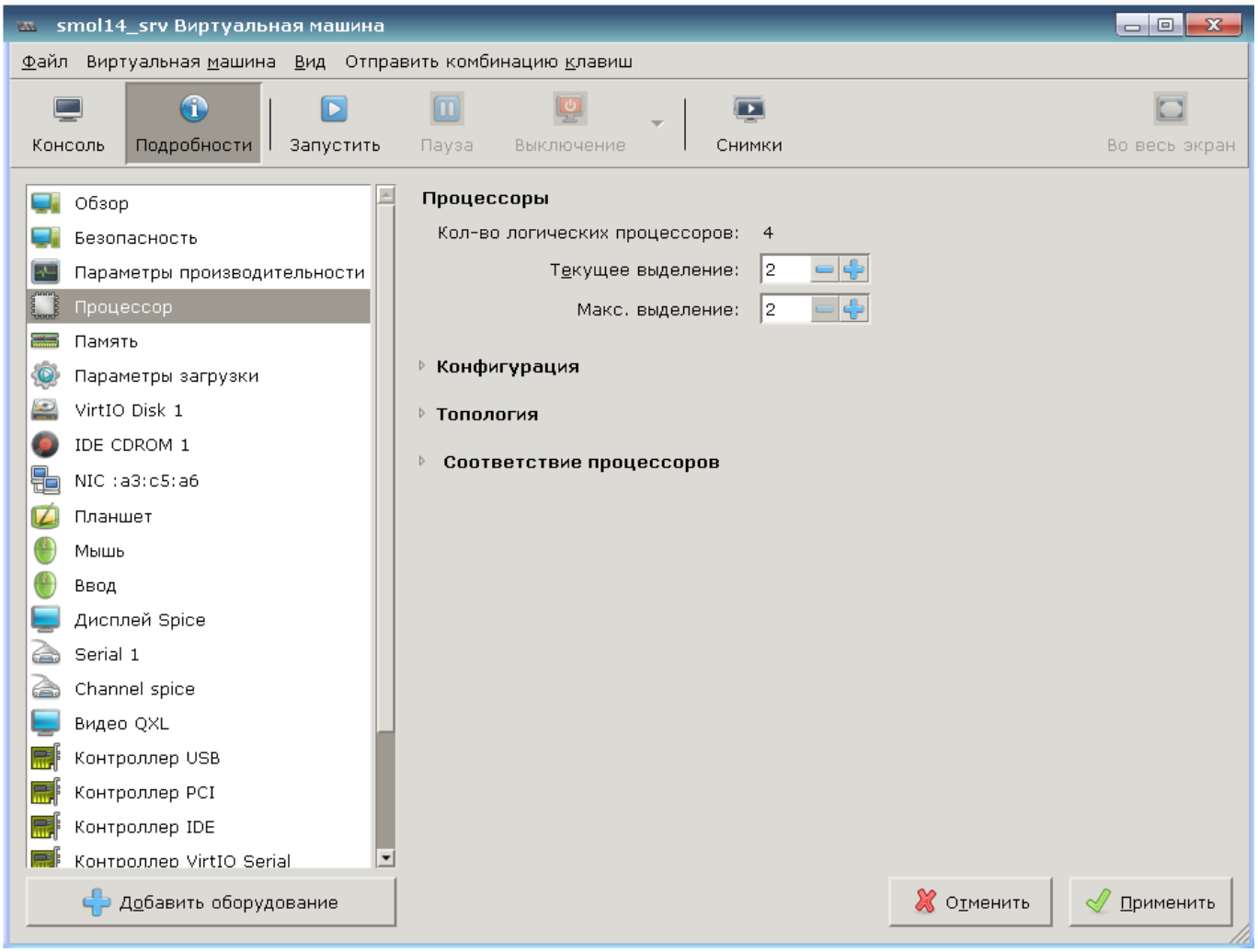

Рис. 6

Утилита позволяет выполнять действия по созданию виртуальных машин, управлению их конфигурацией и файлами-образов дисковых носителей. Также обеспечивает удаленный доступ к рабочему столу выбранной виртуальной машины по протоколам VNC и SPICE.

9.6. Средства эмуляции аппаратного обеспечения на основе QEMU

Средства эмуляции аппаратного обеспечения на основе QEMU реализуют программно-аппаратное окружение запускаемой виртуальной машины, включая заданную конфигурацию аппаратной платформы и набор эмулируемых устройств, доступных гостевой операционной системе. В случае совпадения гостевой аппаратной платформы и аппаратной платформы хостовой машины используются возможности аппаратной поддержки виртуализации средствами KVM (Kernel-based Virtual Machine) для хостовых операционных систем семейства Linux.

Компонент состоит из пакетов, представляющих программу эмуляции аппаратного обеспечения QEMU для различных аппаратных платформ и необходимый набор утилит командной строки.

QEMU Guest Agent (гостевой агент QEMU) обеспечивает возможность взаимодействия с гостевой ОС. Для отправки и получения команд гостевой агент использует последовательное соединение virtio. Он позволяет зафиксировать файловую систему до выполнения снимка, при этом в снимке не будет большой части записанных данных. Фиксация файловой системы возможна только с драйверами хранилищ Ceph и qcow2. Для использования агента необходимо установить пакет `qemu-guest-agent` на гостевой ОС.

ВНИМАНИЕ! Применение гостевого агента QEMU доступно только для виртуальных машин, запущенных из нулевого мандатного контекста.

Запущенная средствами QEMU/KVM виртуальная машина представляет собой отдельный процесс хостовой операционной системы.

Основным конфигурационным файлом QEMU является `/etc/libvirt/qemu.conf`. Он содержит описание параметров, необходимых для запуска и функционирования виртуальных машин (например, интерфейсов взаимодействия с рабочим столом виртуальных машин, способов и параметров аутентификации, политики управления безопасностью и изоляцией виртуальных машин), а также значения по умолчанию некоторых параметров конфигурации виртуальных машин. Описание основных параметров конфигурационного файла `/etc/libvirt/qemu.conf` приведено в таблице 45.

Таблица 45

Параметр	Описание
`vnc_listen`	Адрес сетевого интерфейса для приема соединений VNC
`vnc_tls`	Использовать TLS для приема соединений VNC

Окончание таблицы 45

Параметр	Описание
vnc_tls_x509_cert_dir	Путь к сертификатам TLS при работе VNC
vnc_password	Пароль для соединений VNC
vnc_sasl	Использовать SASL для приема соединений VNC
vnc_sasl_dir	Каталог конфигурационного файла qemu.conf, содержащий SASL-параметры для приема соединений VNC (см. 9.3)
spice_listen	Адрес сетевого интерфейса для приема соединений по протоколу SPICE
spice_tls	Использовать TLS для приема соединений по протоколу SPICE
spice_tls_x509_cert_dir	Путь к сертификатам TLS при работе по протоколу SPICE
spice_password	Пароль для соединений по протоколу SPICE
spice_sasl	Использовать SASL для приема соединений по протоколу SPICE
spice_sasl_dir	Каталог конфигурационного файла qemu.conf, содержащий SASL-параметры для приема соединений по протоколу SPICE
security_driver	Применяемый драйвер безопасности. Параметр должен содержать значение "parsec"
run_images_dir	Каталог расположения копий файлов-образов виртуальных машин, запущенных в режиме запрета модификации файлов-образов

Примечание. Конфигурационные параметры TLS для доступа к рабочим столам виртуальных машин посредством VNC рассматриваются в 9.8.

ВНИМАНИЕ! При использовании в виртуальной машине SPICE-графики, в гостевой ОС должен быть установлен QXL-драйвер. В ОС драйвер устанавливается с пакетом xserver-xorg-video-qxl.

9.7. Идентификация и аутентификация при доступе к серверу виртуализации libvirt

Сервер виртуализации может использовать для идентификации и аутентификации клиентов следующие механизмы:

- локальная peer-cred аутентификация;
- удаленная SSH-аутентификация (строка соединения qemu+ssh://...);
- удаленная SASL-аутентификация, в том числе с поддержкой Kerberos (строка соединения qemu+tcp://...);
- удаленная TLS-аутентификация с использованием сертификатов (строка соединения qemu+tls://...).

ВНИМАНИЕ! В целях обеспечения удаленного доступа пользователей с использованием сетей связи общего пользования к средствам виртуализации из состава ОС должны приме-

няться средства криптографической защиты информации, прошедшие процедуру оценки соответствия согласно законодательству Российской Федерации.

Параметры для различных способов аутентификации задаются в конфигурационном файле `/etc/libvirt/libvirtd.conf`: параметры локальных UNIX сокетов (секция «UNIX socket access control»), разрешение приема сетевых соединений tcp и tls (параметры `listen_tls` и `listen_tcp`) и порты для их приема (параметры `tls_port` и `tcp_port`), расположение необходимых файлов при использовании сертификатов x509 (секция «TSL x509 certificate configuration»), варианты аутентификации (параметры `auth_unix_ro`, `auth_unix_rw`, `auth_tcp`, `auth_tls`).

По умолчанию используется следующее расположение файлов сертификатов на сервере для аутентификации к серверу виртуализации libvirt:

- `/etc/pki/CA/cacert.pem` — корневой сертификат;
- `/etc/pki/CA/crl.pem` — файл отозванных сертификатов;
- `/etc/pki/libvirt/servercert.pem` — сертификат открытого ключа сервера виртуализации libvirt;
- `/etc/pki/libvirt/private/serverkey.pem` — закрытый ключ сервера виртуализации libvirt.

Примечание. Файлы ключей сервера виртуализации libvirt должны быть доступны на чтение группе `libvirt-qemu`.

По умолчанию используется следующее расположение файлов сертификатов на клиенте для аутентификации к серверу виртуализации libvirt («~» — домашний каталог пользователя):

- `/etc/pki/CA/cacert.pem` — корневой сертификат;
- `~/.pki/libvirt/clientcert.pem` — сертификат открытого ключа клиента;
- `~/.pki/libvirt/clientkey.pem` — закрытый ключ клиента.

В случае SASL-аутентификации используется конфигурационный файл `/etc/sasl2/libvirt.conf`, в котором задаются параметры аутентификации SASL (например, применяемые механизмы). Имя службы сервера виртуализации libvirt при использовании SASL-аутентификации регистрируется как `libvirt/<имя сервера>@<домен>`.

ВНИМАНИЕ! При указании механизма SASL `gssapi` следует в конфигурационном файле `/etc/default/libvirtd` указать с помощью соответствующей переменной окружения расположение файла ключей Kerberos сервера виртуализации, например:

```
export KRB5_KTNAME=/etc/libvirt/libvirt.keytab.
```

Для организации двусторонней аутентификации пользователя по ключам при удаленном подключении к серверу виртуализации необходимо на узле, с которого будет производиться

подключение, сгенерировать SSH-ключ и скопировать его публичную часть на сервер. Для этого с правами пользователя, от имени которого будет создаваться подключение, требуется выполнить следующие действия:

1) создать ключ командой:

```
ssh-keygen -t rsa
```

2) скопировать созданный ключ на хост командой (при соответствующем запросе ввести пароль для аутентификации):

```
ssh-copy-id user@host
```

где `host` — IP-адрес сервера с libvirt;
`user` — пользователь, заведенный на сервере. В результате появится возможность работы с домашними каталогами пользователя `user` на сервере с libvirt.

Для защиты закрытого ключа (аутентифицирующего пользователя) от утечки, он может быть преобразован на парольной фразе (passphrase), которая задается при создании ключа. Пользователю необходимо вводить пароль для преобразования ключа один раз в начале сессии.

При следующем подключении запрос о подлинности сервера не задается, если сервер не был подменен (ключ сервера не изменился). Если сервер оказывается подменен (изменен адрес, на который разрешается имя `host`), выдается предупреждение и соединение не устанавливается.

Для осуществления подключения по SSH запустить `virt-manager`, в меню выбрать «Файл — Добавить соединение», отметить флаг «Connect to remote host over SSH» в соответствии с рис. 7.

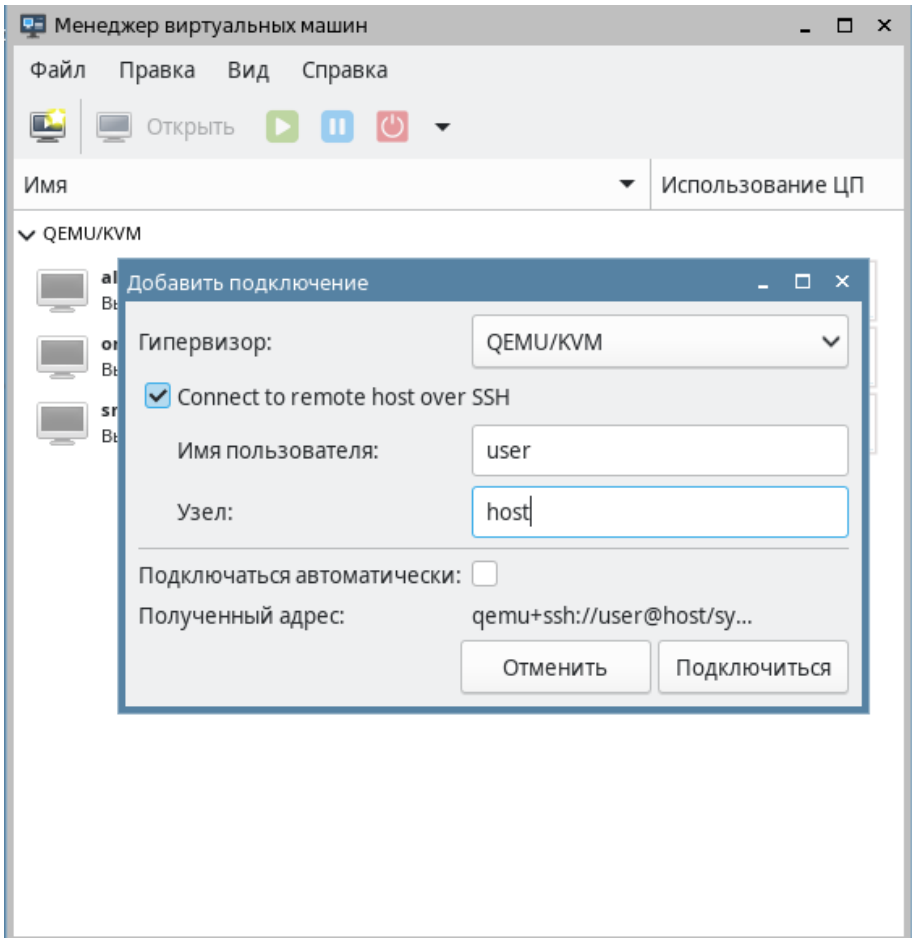

Рис. 7

9.8. Идентификация и аутентификация при доступе к рабочему столу виртуальных машин

Параметры аутентификации при доступе к рабочему столу виртуальной машины задаются в конфигурационном файле `/etc/libvirt/qemu.conf` раздельно для VNC и SPICE. В данном конфигурационном файле указываются необходимые параметры аутентификации и пути к конфигурационным файлам SASL, например `/etc/sasl2/qemu.conf`. Имя служб VNC и SPICE при использовании SASL-аутентификации регистрируется как `vnc/<имя сервера>@<домен>` и `spice/<имя сервера>@<домен>`, соответственно.

По умолчанию используется следующее расположение файлов сертификатов на сервере для аутентификации к виртуальной машине посредством VNC:

- `/etc/pki/libvirt-vnc/ca-cert.pem` — корневой сертификат;
- `/etc/pki/libvirt-vnc/server-cert.pem` — сертификат открытого ключа сервера VNC QEMU;
- `/etc/pki/libvirt-vnc/server-key.pem` — закрытый ключ сервера VNC QEMU.

Примечание. Файлы ключей сервера VNC QEMU должны быть доступны на чтение группе `libvirt-qemu`.

По умолчанию используется следующее расположение файлов сертификатов на клиенте для аутентификации к серверу VNC QEMU («~» — домашний каталог пользователя):

- `/etc/pki/CA/cacert.pem` — корневой сертификат;
- `~/.pki/libvirt-vnc/clientcert.pem` — сертификат открытого ключа клиента;
- `~/.pki/libvirt-vnc/private/clientkey.pem` — закрытый ключ клиента.

10. КОНТЕЙНЕРИЗАЦИЯ

В ОС реализован механизм контейнеризации, обеспечивающий режим виртуализации и изоляции ресурсов на уровне ядра ОС. Использование данного механизма позволяет запускать приложение и необходимый ему минимум системных библиотек в полностью стандартизованном контейнере, соединяющемся с хостовой ОС при помощи определенных интерфейсов.

Контейнеры используют ядро хостовой ОС и, в отличие от полной виртуализации, не требуют эмуляции аппаратного обеспечения. Приложения, запущенные внутри разных контейнеров, изолированы и не могут влиять друг на друга.

10.1. Контейнеризация с использованием Docker

В состав ОС входит программное обеспечение Docker для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации.

10.1.1. Установка Docker

Установка Docker возможна либо через графический менеджер пакетов Synaptic, либо через терминал с помощью команды:

```
sudo apt install docker.io
```

После установки возможно добавить пользователя в группу `docker`, что позволит работать с Docker без использования `sudo`.

Для включения пользователя в группу `docker` выполнить команду:

```
sudo usermod -aG docker <имя_пользователя>
```

Текущего пользователя можно включить в группу командой:

```
sudo usermod -aG docker $USER
```

Для применения действия необходимо выйти из текущей сессии пользователя и зайти повторно.

10.1.2. Работа с Docker

Полный список команд для работы с Docker доступен на странице помощи:

```
docker help
```

Информацию о параметрах конкретной команды можно получить на странице помощи или в справочной странице `man`.

Пример

```
docker attach --help
man docker-attach
```

ВНИМАНИЕ! Описание работы с образами и контейнерами Docker приведено для привилегированного режима. Данный режим не рекомендуется к применению в связи с потенциальной небезопасностью использования контейнеров в привилегированном режиме. Рекомендуется работать с Docker в непривилегированном (rootless) режиме в соответствии с описанием 10.1.3.

10.1.2.1. Создание образа Docker

Образ — это шаблон контейнера, включающий в себя:

1) базовую файловую систему;
2) слои — изменения в файловой системе, расположенные друг над другом в том порядке, в котором эти изменения были произведены;
3) параметры выполнения, используемые при запуске контейнера из данного образа.

Примечание. Из одного образа возможно запускать несколько контейнеров.

Каждый слой образа представляет собой инструкцию, выполняемую в базовой файловой системе при создании образа. В процессе работы контейнера изменения файловой системы образуют новый слой контейнера, а слои образа остаются неизменными.

Слои могут быть последовательно записаны в текстовом документе, который называется докерфайлом (Dockerfile).

Образ возможно создать тремя способами:

- из `chroot`-окружения;
- с помощью докерфайла;
- на основе контейнера.

Создание образа из chroot-окружения

Для создания собственных образов Docker из `chroot`-окружения необходимо установить пакет `debootstrap`. Это можно сделать либо с помощью графического менеджера пакетов Synaptic, либо из терминала, выполнив команду:

```
sudo apt install debootstrap
```

Для создания образа Docker необходимо:

1) собрать `chroot`-окружение;
2) настроить `chroot`-окружение;
3) конвертировать `chroot`-окружение в образ Docker.

Сборка `chroot`-окружения выполняется инструментом командной строки `debootstrap` от имени администратора.

Загрузка пакетов для сборки `chroot`-окружения может быть выполнена из репозитория, доступного по сети.

Пример

```
sudo debootstrap --verbose \
     --components=main,contrib,non-free 1.8_x86-64 /var/docker-chroot \
     http://dl.astralinux.ru/astra/stable/1.8_x86-64/repository-main
```

где `1.8_x86-64` — код дистрибутива;
`/var/docker-chroot` — каталог сборки окружения;
`http://dl.astralinux.ru/...` — расположение репозитория в сети.

Загрузка пакетов для сборки `chroot`-окружения также может быть выполнена из репозитория в локальной ФС.

При сборке `chroot`-окружения для удобства дальнейшей работы можно сразу установить пакеты `ncurses-term`, `mc`, `locales`, `nano`, `gawk`, `lsb-release`, `acl`, `perl-modules`.

Пример

```
sudo debootstrap --verbose --include \
     ncurses-term,mc,locales,nano,gawk,lsb-release,acl,perl-modules-5.28 \
     1.8_x86-64 /var/docker-chroot file:///srv/repo
```

где 1.8_x86-64 — код дистрибутива;
/var/docker-chroot — каталог сборки окружения;
file:///srv/repo — каталог локального репозитория.

Примечание. При включенном МРД и/или МКЦ рекомендуется размещать каталог сборки chroot-окружения в /var. Данный каталог имеет метку безопасности 3:63:-1:ccnr, что позволяет создавать в нем файловые объекты с любыми метками безопасности. Для работы пользователей в непривилегированном режиме администратором системы должен быть создан доступный пользователю каталог с необходимой меткой безопасности.

Настройка chroot-окружения выполняется от имени администратора в следующей последовательности:

1) при необходимости настроить для chroot-окружения разрешение имен в файле /etc/resolv.conf и список репозиториев в /etc/apt/sources.list (например, скопировать одноименные файлы из корневой ФС в каталог для chroot-окружения);

2) перейти в chroot-окружение командой sudo chroot и обновить пакеты окружения:

```
sudo chroot /var/docker-chroot
apt update
apt dist-upgrade
exit
```

Для создания образа Docker следует добавить настроенное chroot-окружение в архив с помощью инструмента командной строки tar, запущенного от имени администратора, и конвертировать полученный архив в образ командой:

```
docker import <параметры>
```

Пример

Создать образ wiki/astralinux:se из chroot-окружения:

```
sudo tar -C /var/docker-chroot -cpf - . | sudo docker import \
    - wiki/astralinux:se --change "ENV PATH \
    /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin" \
    --change 'CMD ["/bin/bash"]'
```

где `-C /var/docker-chroot` — задать каталог в качестве рабочего каталога для архивирования;
`-cpf - .` — создать новый архив из рабочего каталога с сохранением разрешений, установленных на входящие в него каталоги и файлы, и передать архив в `stdin`;
`docker import -` — импортировать данные для создания образа из `stdout`;
`--change "ENV PATH ..."` — задать переменную окружения `PATH`;
`--change 'CMD ["/bin/bash"]'` — задать команду, которая будет автоматически выполнена в контейнере при запуске контейнера из данного образа.

Если все операции выполнены успешно, то созданный образ будет отображаться в списке образов, доступном по команде:

```
sudo docker images
```

Для запуска контейнера из созданного образа используется команда:

```
sudo docker run -it --rm <имя_образа>
```

где `-i` — запустить контейнер в интерактивном режиме;
`-t` — выделить терминал для контейнера;
`--rm` — удалить контейнер после выхода из него.

Создание образа с использованием докерфайла

Докерфайл представляет собой инструкции для создания образа Docker. Используя докерфайл и контекст (совокупность каталогов и файлов в указанном месте) возможно создавать новые образы на основе существующих с помощью команды:

```
docker build
```

При этом полное содержимое контекста рекурсивно пересылается службе `dockerd` и может быть скопировано в создаваемый образ командами, указанными в докерфайле. Поэтому не рекомендуется использовать в качестве контекста корневой каталог файловой системы ОС. Местоположение контекста может быть задано как путь к каталогу в файловой системе либо как ссылка на репозиторий в сети.

По умолчанию используется докерфайл с именем `Dockerfile`, расположенный в каталоге контекста сборки. Произвольное расположение докерфайла задается параметром `-f`:

```
docker build -f <путь_к_докерфайлу> .
```

Перед выполнением инструкций в докерфайле проводится их проверка на корректность. Если в инструкциях содержится ошибка (например, неправильный синтаксис), при попытке собрать образ будет выведено сообщение об ошибке.

Пример

Создать образ с использованием докерфайла, содержащего несуществующую инструкцию `RUNCMD`:

```
docker build -t test/myimg .
```

В терминале будет выведено сообщение об ошибке:

```
Sending build context to Docker daemon 2.048 kB
Error response from daemon: Unknown instruction: RUNCMD
```

При создании нового образа инструкции выполняются последовательно и результат выполнения каждой инструкции записывается в отдельный слой образа.

Пример

Для сборки нового образ на основе существующего образа `wiki/astralinux:se` следует:

1) создать корневой каталог контекста сборки:

```
mkdir build-smolensk
```

2) создать в контексте сборки файл `data-to-import` содержащий произвольный текст:

```
echo "Это импортированные данные" > build-smolensk/data-to-import
```

3) в файл `build-smolensk/Dockerfile` внести следующий текст:

```
# указание из какого образа выполнять сборку
FROM wiki/astralinux:se
# скопировать файл data-to-import из контекста сборки в образ
COPY /data-to-import /srv
# создать в образе пустой файл /srv/created-file
RUN touch /srv/created-file
# вывести на печать содержимое скопированного файла
RUN cat /srv/data-to-import
# вывести на печать рабочий каталог
RUN echo Current work directory is $(pwd)
```

4) выполнить сборку образа с именем `test`:

```
docker build -t test build-smolensk/
```

Вывод в терминале будет иметь следующий вид:

```
Sending build context to Docker daemon   5.12kB
Step 1/5 : FROM wiki/astralinux:se
---> 60d0611fe56a
Step 2/5 : COPY /data-to-import /srv
---> 7a75a002d29f
Step 3/5 : RUN touch /srv/created-file
---> Running in 709bb54af8c3
Removing intermediate container 709bb54af8c3
---> b5fd28178901
Step 4/5 : RUN cat /srv/data-to-import
---> Running in 4c69f455cf2f
Это импортированные данные
Removing intermediate container 4c69f455cf2f
---> c8f8c7c3797a
Step 5/5 : RUN echo Current work directory is $(pwd)
---> Running in 27db5fcaaba5
Current work directory is /
Removing intermediate container 27db5fcaaba5
---> 14446097a09e
Successfully built 14446097a09e
Successfully tagged test:latest
```

5) проверить, что образ `test` присутствует в списке образов:

```
docker images
```

6) если образ был успешно создан, запустить контейнер из образа:

```
docker run --rm -it test
```

7) проверить содержимое контейнера, выполнив в контейнере команду:

```
ls -l /srv
```

Результат выполнения команды должен быть следующего вида:

```
total 4
-rw-r--r-- 1 root root  0 Jan 20 10:12 created-file
-rw-r--r-- 1 root root 51 Jan 20 10:11 data-to-import
```

затем выполнить команду:

```
cat /srv/data-to-import
```

Результат выполнения команды должен быть следующего вида:

```
Это импортированные данные
```

Из вывода в терминале видно, что в контейнере присутствует файл `data-to-import`, скопированный из контекста сборки в образ, и пустой файл `created-file`, созданный в образе при сборке.

Подробное описание команды `docker build` и работы с докерфайлами приведено в `man docker-build` и `man dockerfile`, соответственно.

Создание образа из контейнера

При наличии сохраненного или активного контейнера (описание работы с контейнерами приведено в 10.1.2.3) данный контейнер возможно конвертировать в образ Docker следующей командой:

```
docker container commit <параметры> <имя_контейнера> <имя_образа>
```

Пример

Создать образ `test-image` из контейнера `test`:

```
docker container commit test test-image
```

Все изменения в контейнере относительно образа, из которого тот был запущен, а также команды, переданные в качестве параметров при создании нового образа, сформируют новый слой создаваемого образа.

Параметры данной команды описаны в `man docker-container-commit`.

10.1.2.2. Копирование образа

Образ, хранящийся на локальной машине, может быть скопирован (например на другую машину).

Пример

Для того чтобы скопировать образ `wiki/astralinux:se` на другую машину, необходимо:

1) выгрузить образ в архив:

```
docker save -o astralinux-se.bz2 wiki/astralinux:se
```

где `-o` — задает имя файла, в который будет выведен образ. Если этот параметр не указан, образ будет выведен в `stdout`;

2) скопировать полученный файл `astralinux-se.bz2` на целевую машину;

3) на целевой машине загрузить файл в локальный реестр образов:

```
docker load -i astralinux-se.bz2
```

где `-i` — указывает имя файла, из которого будет загружен образ. Если этот параметр не указан, образ будет загружен из `stdin`.

Эту процедуру возможно выполнить одной командой с копированием созданного архива через SSH (для этого на целевой машине должен быть настроен SSH):

```
docker save wiki/astralinux:se | bzip2 | ssh user@host \
     'bunzip2 | docker load'
```

где `docker save wiki/astralinux:se` — выгрузить образ `wiki/astralinux:se` в `stdout`;
`bzip2` — программа сжатия данных;
`ssh user@host 'bunzip2 | docker load'` — подключиться через SSH к машине с именем `host` от имени пользователя `user` и запустить команды загрузки образа из стандартного ввода (`stdin`). Пользователь `user` на целевой машине должен иметь право работать с Docker без использования `sudo`.

10.1.2.3. Создание и работа с контейнерами

Для того, чтобы создать новый контейнер с заданным именем из образа, используется следующая команда:

```
docker run <параметры> <имя_образа>
```

Примеры:

1. Создать контейнер с именем `run-smolensk` из образа `smolensk`:

```
docker run --name run-smolensk --rm -it smolensk
```

где `run-smolensk` — имя контейнера. Если параметр не указан, присваивается случайное имя;

`--rm` — уничтожить контейнер после завершения его работы. Если параметр не указан, контейнер будет локально сохранен;

`-i` — запустить контейнер в интерактивном режиме. Если параметр не указан, контейнер запустится в фоновом режиме;

`-t` — создать терминал;

`smolensk` — имя образа, из которого создается контейнер.

2. Для создания нескольких контейнеров с произвольными именами из образа `smolensk` следует:

1) запустить контейнер из образа `smolensk`:

```
docker run smolensk
```

2) выполнить команду повторно;

3) вывести список контейнеров:

```
docker container ls -a
```

В выводе будут отображены два контейнера со случайными именами, созданные из образа `smolensk`:

```
CONTAINER ID IMAGE ... NAMES
b894e0b0b22d smolensk ... admiring_murdock
825a33f9c18c smolensk ... amazing_morse
```

Для запуска сохраненного контейнера следует использовать команду:

```
docker start <имя_контейнера>
```

Пример

Запустить контейнер `amazing_morse` в интерактивном режиме:

```
docker start -ai amazing_morse
```

К контейнеру, работающему в фоновом режиме, можно подключиться командой:

```
docker attach <имя_контейнера>
```

Пример

Подключиться к контейнеру `amazing_morse`, работающему в фоновом режиме:

```
docker attach amazing_morse
```

Для просмотра списков контейнеров выполнить команду:

```
sudo docker container list
```

10.1.2.4. Запуск контейнеров на выделенном уровне МКЦ

С целью изоляции и ограничения среды исполнения потенциально опасного или вредоносного кода в ОС реализована возможность запуска контейнеров на низком уровне МКЦ. Описание функции приведено в документе РУСБ.10015-01 97 01-1.

10.1.2.5. Монтирование файловых ресурсов хостовой машины в контейнер

Docker поддерживает следующие типы монтирования файловых ресурсов хостовой машины в контейнер:

1) `bind` — монтирование файла или каталога, расположенного на хостовой машине, в контейнер;

2) `mount` — монтирование управляемых Docker изолированных томов для хранения данных;

3) `tmpfs` — монтирование временного файлового хранилища (`tmpfs`) в контейнер. Это позволяет контейнеру размещать временные ресурсы в памяти хостовой машины.

Параметры монтирования задаются при создании контейнеров и сохраняются в течение их работы.

ВНИМАНИЕ! Чтобы предотвратить нежелательные изменения в конфигурации хостовой машины, следует исключить монтирование файловых ресурсов, влияющих на конфигурацию хостовой машины, либо ограничить права доступа контейнера к файловым ресурсам правом на чтение.

Параметры монтирования могут быть заданы с использованием одного из двух флагов, определяющих формат, в котором будут заданы параметры и их значения:

1) с использованием флага `-v` — параметры монтирования задаются набором значений, разделенных двоеточием. Набор параметров зависит от типа монтирования. Например, тип монтирования `bind` будет иметь следующий вид:

```
docker run -v <монтируемый_ресурс>:<точка_монтирования>:
     <дополнительные_параметры> <имя_образа>
```

2) с использованием флага `--mount` — параметры монтирования задаются в виде `<параметр>=<значение>` и отделяются друг от друга запятыми:

```
docker run --mount type=<тип_монтирования>,source=<монтируемый_ресурс>,
     target=<точка_монтирования>,<дополнительные_параметры> <имя_образа>
```

bind

Тип монтирования `bind` монтирует каталог, расположенный на хостовой машине, в ФС контейнера. Содержимое каталога на хостовой машине и в точке монтирования в контейнере полностью идентично, а изменения в одном каталоге повторяются в другом.

ВНИМАНИЕ! Данный метод монтирования является устаревшим.

С использованием флага `-v` параметры типа монтирования `bind` задаются следующим образом:

```
docker run -v <монтируемый_ресурс>:<точка_монтирования>:
     <дополнительные_параметры> <имя_образа>
```

С использованием флага `--mount` параметры типа монтирования `bind` задаются следующим образом:

```
docker run --rm -it --mount type=<тип_монтирования>,
     source=<монтируемый_ресурс>,target=<точка_монтирования>,
     <дополнительные_параметры> <имя_образа>
```

Примеры:

1. Смонтировать рабочий каталог хостовой машины в каталог `/app` контейнера с параметром `readonly`, используя флаг `--mount`:

```
docker run --rm -it --mount \
     type=bind,source="$(pwd)",target=/app,readonly smolensk
```

2. Смонтировать рабочий каталог хостовой машины в каталог `/app` контейнера с параметром `read-only`, используя флаг `-v`:

```
docker run --rm -it -v $(pwd):/app:ro smolensk
```

Тип монтирования `bind` позволяет настраивать распространение монтирования (bind propagation). В контексте контейнеризации распространение монтирования определяет, как события монтирования (монтирование и размонтирование ресурсов) в контейнере могут повлиять на ресурсы хостовой машины и/или других контейнеров, а события монтирования на хостовой машине — на ресурсы одного или нескольких контейнеров.

Для настройки распространения монтирования используется параметр `bind-propagation`. Параметр принимает следующие значения:

- `shared` — если при создании контейнера в него был смонтирован каталог с этим параметром, например каталог `/myfiles` на хостовой машине в `/contfiles` в контейнере, то другие ресурсы, смонтированные внутри `/contfiles`, также будут доступны в `/myfiles`. Аналогично ресурсы, смонтированные внутри `/myfiles`, будут доступны в `/contfiles`. Если при создании нескольких контейнеров в каждый из них был смонтирован с этим параметром один и тот же каталог, то смонтированные внутри него ресурсы также будут доступны в каждом из данных контейнеров;
ВНИМАНИЕ! В режиме `shared` изменения в одной точке монтирования распространяются на все остальные точки монтирования, что может привести к нежелательным изменениям файловых объектов других контейнеров, и, как следствие, нарушению их работы;
- `rshared` — то же, что `shared`, но применяется рекурсивно;
- `slave` — если при создании контейнера в него был смонтирован каталог с этим параметром, например каталог `/myfiles` на хостовой машине в `/contfiles` в контейнере, то другие ресурсы, смонтированные внутри `/myfiles`, также будут доступны в каталоге `/contfiles`, но при этом ресурсы, смонтированные внутри `/contfiles`, не будут доступны на хостовой машине;
- `rslave` — то же, что `slave`, но применяется рекурсивно;
- `private` — если при создании контейнера в него был смонтирован каталог с этим параметром, например каталог `/myfiles` на хостовой машине в `/contfiles` в контейнере, то другие ресурсы, смонтированные внутри `/contfiles`, не будут доступны на хостовой машине, а ресурсы, смонтированные внутри `/myfiles`, не будут доступны в контейнере;
- `rprivate` — то же, что `private`, но применяется рекурсивно. Используется по умолчанию.

Примеры:

1. Смонтировать подкаталог `/target` рабочего каталога хостовой машины в каталог `/app` контейнера с типом распространения монтирования `rslave`, используя флаг `--mount`:

```
docker run -d -it --mount \
     type=bind,source="$(pwd)"/target,target=/app,readonly,\
     bind-propagation=rslave smolensk
```

2. Смонтировать подкаталог `/target` рабочего каталога хостовой машины в каталог `/app` контейнера с типом распространения монтирования `shared`, используя флаг `-v`:

```
docker run -d -it -v "$(pwd)"/target:/app:ro,shared smolensk
```

mount

Том Docker представляет собой файловую систему, расположенную на хостовой машине вне контейнера и находящуюся под управлением Docker. Тома хранятся в каталоге Docker на хостовой машине, например `/var/lib/docker/volumes/`. Тома существуют независимо от жизненного цикла контейнера и могут быть многократно использованы разными контейнерами. Управление томами описано в `man docker-volume`.

Для создания тома используется следующая команда:

```
docker volume create <имя_тома>
```

Пример

Создать том с именем `my-vol`:

```
docker volume create my-vol
```

С флагом `-v` параметры монтирования `mount` задаются следующим образом:

```
docker run -v <имя_тома>:<точка_монтирования> <имя_образа>
```

С использованием флага `--mount` команда будет иметь следующий вид:

```
docker run --mount src=<имя_тома>,dst=<точка_монтирования> <имя_образа>
```

Примеры:

1. Смонтировать том `my-vol` в каталог `/app` контейнера с использованием флага `-v`:

```
docker run --rm -it -v my-vol:/app smolensk
```

2. Смонтировать том `my-vol` в каталог `/app` контейнера с использованием флага `--mount`:

```
docker run --rm -it --mount src=my-vol,dst=/app smolensk
```

tmpfs

Тип монтирования `tmpfs` монтирует временное файловое хранилище (`tmpfs`) в ФС контейнера, что позволяет контейнеру хранить временные файловые ресурсы в памяти хостовой машины. Доступ к этим файловым ресурсам имеет только тот контейнер, в котором они были созданы. При остановке контейнера временные файловые ресурсы будут полностью удалены из ФС контейнера и памяти хостовой машины.

С использованием флага `--mount` параметры монтирования `tmpfs` задаются следующим образом:

```
docker run --mount type=tmpfs,destination=<точка_монтирования> <имя_образа>
```

С использованием флага `--tmpfs` команда будет иметь следующий вид:

```
docker run --tmpfs <точка_монтирования> <имя_образа>
```

Примеры:

1. Запустить контейнер из образа `smolensk` с монтированием `tmpfs` в каталог контейнера `/app`, используя флаг `--mount`:

```
docker run --rm -it --mount type=tmpfs,destination=/app smolensk
```

2. Запустить контейнер из образа `smolensk` с монтированием `tmpfs` в каталог контейнера `/app`, используя флаг `--tmpfs`:

```
docker run --rm -it --tmpfs /app smolensk
```

Примечания:

1. Синтаксис `--tmpfs` не поддерживает использование параметров монтирования.
2. Монтирование `tmpfs` не поддерживает флаг `-v`.

10.1.3. Работа с Docker в непривилегированном режиме

Работа с образами и контейнерами Docker в непривилегированном (rootless) режиме подразумевает работу от имени пользователя без использования механизма `sudo`. В непривилегированном режиме служба контейнеризации и контейнеры не получают прав суперпользователя в хостовой ОС, при этом для приложения в контейнере служба контейнеризации работает как суперпользователь.

Для работы с Docker в непривилегированном режиме используется инструмент командной строки `rootlessenv`, который настраивает окружение для работы в данном режиме. При работе в непривилегированном режиме инструмент `rootlessenv` должен использоваться вместо механизма `sudo` в командах при настройке и работе с образами и контейнерами Docker (см. 10.1.2).

Для настройки работы в режиме rootless необходимо выполнить следующие шаги:

1) установить Docker в соответствии с 10.1.1;

2) установить пакет `rootless-helper-astra` для использования непривилегированного режима :

```
sudo apt install rootless-helper-astra
```

3) запустить службы rootless Docker для пользователя, который будет использовать образы и контейнеры Docker в непривилегированном режиме:

```
sudo systemctl start rootless-docker@<имя_пользователя>
```

4) при необходимости настроить автозапуск служб rootless Docker выполнить:

```
sudo systemctl enable rootless-docker@<имя_пользователя>
```

Запуск и настройка автозапуска служб могут быть выполнены для нескольких пользователей, для этого необходимо выполнить соответствующие команды отдельно для каждого пользователя.

Чтобы запустить контейнер от имени текущего пользователя с использованием `rootlessenv`, следует выполнить:

```
rootlessenv docker run --rm -ti <имя_образа>
```

Для запуска контейнера от имени произвольного пользователя с `rootlessenv` выполнить:

```
sudo -u <имя_пользователя> rootlessenv docker run --rm -ti <имя_образа>
```

Работа с образами и контейнерами, созданными в режиме rootless, возможна только в режиме rootless.

Для просмотра списка контейнеров, созданных в режиме rootless, выполнить команду:

```
rootlessenv docker container list
```

Работа с `rootless-helper-astra` и `rootlessenv` более подробно описана в `man rootless-helper-astra` и `man rootlessenv`, соответственно.

Описание работы с образами и контейнерами Docker в непривилегированном режиме с ненулевыми метками безопасности приведено в документе РУСБ.10015-01 97 01-1.

10.2. Контейнеризация с использованием Podman

Программное обеспечение Podman для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации аналогично программному обеспечению Docker, но предоставляет дополнительные возможности по управлению группами контейнеров и может работать без использования учетной записи `root`. Podman использует контейнеры стандарта Open Container Initiative (OCI), что обеспечивает совместимость с образами Docker.

10.2.1. Установка Podman

Podman представлен одноименным пакетом `podman`. Пакет может быть установлен с помощью графического менеджера пакетов Synaptic или из командной строки с помощью команды:

```
sudo apt install podman
```

При работе с включенным мандатным управлением доступом после установки пакета необходимо перезапустить пользовательскую сессию или перезагрузить ОС.

10.2.2. Стандартные команды

Podman поддерживает все команды Docker, кроме `docker swarm`, а также имеет ряд собственных команд. Полный список команд для работы с Podman доступен на странице помощи:

```
podman --help
```

Информация о параметрах конкретной команды доступна на странице помощи или на справочной странице `man`.

Пример

```
podman pod create --help
man podman-pod-create
```

ВНИМАНИЕ! По умолчанию Podman использует среду выполнения контейнеров `crun`. В ОС работа контейнеров поддерживается только при использовании среды `runc`. В связи с этим для запуска контейнеров следует использовать команды Podman с параметрами `--runtime=runc` и `--cgroup-manager=cgroupfs`.

10.2.3. Работа с Podman

10.2.3.1. Включение отладки

Для включения отладки используется параметр `--log-level` с указанием требуемого уровня отладки. Отладочная информация выводится в стандартный поток сообщений об ошибках `stderr`.

Пример

Выполнение команды с включенным уровнем отладки `debug`:

```
podman --log-level debug ps -a
```

10.2.3.2. Запуск контейнера из образа

Для запуска контейнера из загруженного образа используется команда:

```
podman run <параметры> <имя_образа>
```

Пример

Запустить контейнер из образа `astralinux` в интерактивном режиме с терминалом и уничтожить его после завершения работы:

```
podman run -it --rm astralinux /bin/bash
```

Для запуска контейнера в фоновом режиме используется команда:

```
podman run -d <имя_образа>
```

При запуске контейнера к его файловой системе может быть примонтирован каталог из файловой системы хостовой машины. Для этого следует выполнить команду:

```
podman run --mount
     type=bind,source=<монтируемый_каталог>,target=<точка_монтирования>
     <имя_образа>
```

10.2.3.3. Вывод списка контейнеров

Для вывода списка запущенных (работающих) контейнеров используется команда:

```
podman ps
```

Для вывода списка всех контейнеров (в том числе завершивших работу) используется команда:

Пример

```
podman ps -a
```

Пример вывода команды:

```
CONTAINER ID  IMAGE       COMMAND     CREATED
0468d9f62e2d  astralinux  /bin/bash   12 seconds ago

STATUS                     PORTS       NAMES
Exited (0) 5 seconds ago               priceless_hertz
```

10.2.3.4. Действия с сохраненными контейнерами

Возможно выполнение следующих действий с контейнерами (при этом в командах в качестве идентификатора контейнера используется его числовой идентификатор `CONTAINER ID` или имя `NAMES`, см. 10.2.3.3):

1) запуск сохраненного контейнера выполняется командой:

```
podman start <идентификатор_контейнера>
```

2) остановка контейнера выполняется командой:

```
podman stop <идентификатор_контейнера>
```

3) удаление контейнера выполняется командой:

```
podman rm <идентификатор_контейнера>
```

Перед удалением контейнер должен быть остановлен;

4) получение полной информации о контейнере выполняется командой:

```
podman inspect <идентификатор_контейнера>
```

Эта команда выводит большой объем информации, отфильтровать вывод информации можно применением параметра `--format`:

```
podman inspect <идентификатор_контейнера> --format '<параметр_фильтрации>'
```

5) вывод журналов контейнера выполняется командой:

```
podman logs <идентификатор_контейнера>
```

6) вывод статистики работы контейнеров выполняется командой:

```
podman stats
```

Эта команда после запуска не завершает свою работу, а продолжает выводить статистику с заданным интервалом (по умолчанию каждые 5 секунд). Для однократного вывода статистики с последующим завершением выполнения команды работы следует использовать параметр `--no-stream`:

```
podman stats --no-stream
```

10.2.3.5. Удаление образа

Для удаления образа предварительно необходимо остановить и удалить все созданные из него контейнеры. После этого использовать команду:

```
podman rmi <идентификатор_образа>
```

10.2.4. Создание собственного контейнера из существующего образа

Для создания собственного контейнера на основе имеющегося образа следует выполнить следующее:

1) создать докерфайл с указанием образа, из которого будет создаваться контейнер, и команд для его создания:

```
echo -e "FROM smolensk\nRUN
mkdir /testdir\nRUN echo test > /testdir/testfile" > Dockerfile
```

2) создать контейнер:

```
podman --runtime=runc --cgroup-manager=cgroupfs build -t testbuild .
```

10.2.5. Создание собственного образа

Создание собственного образа производится аналогично описанному в 10.1.2.1. При этом в командах следует заменять `docker` на `podman`.

10.2.6. Оркестрация контейнеров

Podman позволяет оркестрировать контейнеры — объединять контейнеры в группы («поды») и управлять ими как единым целым. Контейнеры в поде используют общие ресурсы и пространство имен. Это полезно в ситуациях, когда для выполнения одной задачи требуется одновременная работа нескольких контейнеров, например, база данных в одном контейнере и веб-сервер для доступа к ней в другом контейнере.

10.2.6.1. Создание нового пода

Для создания нового пода необходимо выполнить команду:

```
podman --cgroup-manager cgroupfs pod create <имя_пода>
```

Создание пода происходит следующим образом:

- если имя пода не задано, то используется случайно созданное имя;
- создается полный идентификатор (выводится на экран при успешном создании пода);
- в поде создается служебный контейнер (так называемый infra-контейнер), для чего загружается специальный образ `podman-pause`. Этот контейнер нужен для резервирования места для пода в пространстве имен. Это позволяет в дальнейшем подключать к поду другие (функциональные) контейнеры, а также останавливать все контейнеры пода, оставляя сам под запущенным.

В дальнейшем поды идентифицируются именами или идентификаторами — полным идентификатором, или кратким (первые символы полного идентификатора).

10.2.6.2. Список существующих подов

Для вывода списка подов следует использовать команду:

```
podman pod ps
```

Пример вывода команды:

```
POD ID        NAME       STATUS    CREATED         INFRA ID      # OF CONTAINERS
312fb1c5553f testpod   Created   22 minutes ago 76972a488dbb  1
```

Команда выводит краткий идентификатор пода (`POD ID`), имя пода (`NAME`), количество контейнеров (`# OF CONTAINERS`), а также идентификатор infra-контейнера (`INFRA ID`). Для отображения полных идентификаторов следует использовать параметр `--no-trunc`. Статус пода (`STATUS`) может иметь следующие значения:

- `Created` — в поде нет исполняющихся или остановленных контейнеров;
- `Running` — хотя бы один контейнер исполняется;
- `Stopped` — исполняющихся контейнеров нет, есть хотя бы один остановленный;
- `Exited` — все контейнеры остановлены.
- `Dead` — ошибка получения статуса.

Получить имена контейнеров в подах можно следующей командой:

```
podman pod ps --ctr-names
```

Пример вывода команды:

```
POD ID        NAME        STATUS      CREATED
312fb1c5553f  testpod     Created     56 minutes ago

INFRA ID      NAMES
76972a488dbb  friendly_rhodes,312fb1c5553f-infra,brave_turing
```

Имена контейнеров в поде перечислены в столбце `NAMES` через запятую.

10.2.6.3. Добавление контейнера в под

Для добавления контейнеров в под используются команды создания и запуска контейнеров с параметром `--pod=<идентификатор_пода>`.

Примеры:

1. Создание контейнера `nginx` и добавление его в под `testpod`:

```
podman create --pod=testpod nginx
```

2. Запуск одного контейнера в поде без запуска остальных:

```
podman run -it --pod=testpod mongodb
```

11. ЗАЩИЩЕННЫЙ КОМПЛЕКС ПРОГРАММ ГИПЕРТЕКСТОВОЙ ОБРАБОТКИ ДАННЫХ

Защищенный комплекс программ гипертекстовой обработки данных — это ПО, осуществляющее взаимодействие по HTTP-протоколу между сервером и веб-браузерами: прием запросов, поиск указанных файлов и передача их содержимого, выполнение приложений на сервере и передача клиенту результатов их выполнения. Комплекс представлен веб-сервером Apache2 и веб-браузером Mozilla Firefox.

ВНИМАНИЕ! Для обеспечения нормальной работы пользователя с сетевыми службами должна быть явно задана его классификационная метка (диапазон уровней конфиденциальности и категорий конфиденциальности) с помощью соответствующих утилит, даже если ему не доступны уровни и категории выше 0. Дополнительная информация приведена в документе РУСБ.10015-01 97 01-1.

11.1. Настройка сервера

После установки сервера Apache2 необходимо установить пакет `libapache2-mod-authnz-pam` для настройки сервера и подготовки к приему запросов на всех сетевых интерфейсах на 80 порту.

Если по каким-то причинам он не работоспособен, следует проверить минимально необходимые настройки сервера:

1) в файле `/etc/apache2/ports.conf` должен быть указан параметр:

```
Listen 80
```

2) в каталоге `/etc/apache2/sites-available` должны находиться файлы с настройками виртуальных хостов и как минимум один из них должен быть разрешен к использованию командой:

```
a2ensite <имя_файла>
```

ВНИМАНИЕ! В команде необходимо использовать только имя файла (без указания полного пути).

Для разрешенного к использованию виртуального хоста будет добавлена в каталог `/etc/apache2/sites-enabled` символьная ссылка на его конфигурационный файл.

Минимальное содержимое таких файлов с конфигурациями виртуальных хостов выглядит следующим образом:

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName server.domain.name
        DocumentRoot /var/www/html
```

```
        <Directory /var/www/html>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
        </Directory>
        ErrorLog /var/log/apache2/error.log
        LogLevel warn
        CustomLog /var/log/apache2/access.log combined
</VirtualHost>
```

В случае когда веб-сервер должен предоставлять пользователям доступ к объектам файловой системы с различными мандатными атрибутами, на корневой каталог виртуального хоста (по умолчанию `/var/www/html`) и все его родительские каталоги должны быть установлены значения мандатных атрибутов не меньше максимальных атрибутов объектов, к которым будет разграничиваться доступ. Кроме того, на корневой каталог виртуального хоста (по умолчанию `/var/www/html`) должен быть установлен тип метки `ccnr`. Операция может быть выполнена с использованием утилиты `pdpl-file` от имени учетной записи администратора через механизм `sudo`. Дополнительная информация приведена в документе РУСБ.10015-01 97 01-1.

После окончания правки конфигурационных файлов необходимо перезапустить сервер командой:

```
systemctl restart apache2
```

11.2. Режим работы AstraMode

Сервер гипертекстовой обработки данных Apache2, входящий в состав ОС, поддерживает работу с ненулевыми классификационными метками при включенном мандатном управлении доступом (см. РУСБ.10015-01 97 01-1). Для обеспечения работы средств разграничения доступа должна выполняться аутентификация и авторизация пользователей.

Для управления авторизацией пользователей в сервере Apache2 используется параметр `AstraMode`, который указывается в конфигурационном файле `/etc/apache2/apache2.conf`:

- для включения обязательной авторизации задать для параметра значение `on`:

```
AstraMode on
```

При этом отсутствие в конфигурационном файле параметра `AstraMode` (или закомментированная строка с параметром) соответствует значению `AstraMode on`;

- для отключения обязательной авторизации задать для параметра значение `off`:

```
AstraMode off
```

По умолчанию режим обязательной авторизации включен.

Отключение обязательной авторизации снижает защищенность ресурсов, но может быть необходимо для обеспечения совместимости с программами, не поддерживающими работу с классифицированными метками и не использующими авторизацию.

ВНИМАНИЕ! Для доступа к данным, имеющим ненулевые классификационные метки, должна выполняться авторизация пользователей. Анонимный доступ к данным, имеющим ненулевую классификационную метку, недопустим.

ВНИМАНИЕ! При выключенной авторизации сервер Apache2 осуществляет все запросы к своим ресурсам от имени одной системной учетной записи (по умолчанию `www-data`), которая в случае применения мандатного контроля целостности имеет категорию целостности `1`.

В конфигурационном файле `/etc/apache2/apache2.conf` задается глобальное значение параметра `AstraMode`, которое по умолчанию применяется для всех добавляемых файлов конфигурации и для всех активных веб-сайтов, запускаемых службой.

Глобальное значение параметра `AstraMode` может быть переопределено для добавляемого конфигурационного файла, при этом необходимость авторизации будет учитываться на основе последнего обнаруженного значения. Переопределение значения `AstraMode` для файла, добавляемого в конфигурацию с помощью `IncludeOption` и `Include`, выполняется путем добавления строки со значением параметра `AstraMode` в начале добавляемого файла или перед инструкцией добавления файла.

Пример

```
AstraMode off
IncludeOption conf-new/*.conf
```

Использовать множественные определения параметра не рекомендуется, так как неверное определение или изменение порядка их обработки может вызвать ошибки.

Глобальное значение параметра `AstraMode` может быть переопределено для конкретного виртуального веб-сайта путем указания нового значения в списке параметров для данного веб-сайта.

Пример

```
<VirtualHost *:443>
AstraMode off
...
</VirtualHost>
```

11.3. Настройка авторизации

Настройка сквозной аутентификации и авторизации для сервера и клиента, работающих в рамках ЕПП, описана в 11.4. Если не настроена аутентификация через Kerberos, то для всех ресурсов должна использоваться аутентификация и авторизация через PAM, при этом будет использоваться пользовательская БД, прописанная в настройках ОС. Для выполнения аутентификации и авторизации через PAM должен быть установлен пакет `libapache2-mod-authnz-pam` и выполнена следующая команда от имени учетной записи администратора:

```
a2enmod authnz_pam
```

В конфигурационных файлах виртуальных хостов веб-сервера Apache2 указать:

```
AuthType Basic
AuthName "PAM authentication"
AuthBasicProvider PAM
AuthPAMService apache2
Require valid-user
```

Логин и пароль пользователя будут передаваться от пользователя к серверу в открытом виде с использованием метода аутентификации Basic. Для корректного функционирования авторизации через PAM пользователю, от которого работает веб-сервер (по умолчанию `www-data`), необходимо выдать права на чтение информации из БД пользователей и сведений о метках безопасности:

```
usermod -a -G shadow www-data
setfacl -d -m u:www-data:r /etc/parsec/macdb
setfacl -R -m u:www-data:r /etc/parsec/macdb
setfacl -m u:www-data:rx /etc/parsec/macdb
```

Если установлен модуль веб-сервера Apache2 `auth_kerb` из пакета `libapache2-mod-auth-kerb` для аутентификации через Kerberos в соответствии с 11.4, выключить его использование при помощи команды:

```
a2dismod auth_kerb
```

Для передачи в http-заголовке текущего иерархического уровня конфиденциальности и текущих неиерархических категорий конфиденциальности пользователя может быть сконфигурирован модуль Apache2 `mod_headers`. Для этого необходимо:

1) в конфигурационном файле `/etc/apache2/apache2.conf` добавить строку:

```
Header set MyHeader "%m %c"
```

где `%m` — место подстановки текущего иерархического уровня конфиденциальности;
`%c` — текущих неиерархических категорий конфиденциальности;

2) включить модуль, выполнив команду:

```
a2enmod headers
```

3) перезапустить сервер Apache2:

```
systemctl restart apache2
```

Сервер для PAM-аутентификации использует сценарий PAM, содержащийся в конфигурационном файле `/etc/pam.d/apache2`. PAM-сценарий включает `common-auth` и `common-account`. По умолчанию в ОС для фиксации числа неверных попыток входа пользователей применяется PAM-модуль `pam_tally`. Использование `pam_tally` в секции `auth` в файле `/etc/pam.d/common-auth` обеспечивает увеличение счетчика неверных попыток входа пользователя при начале процесса аутентификации. Для корректной работы данного механизма необходимо разрешить пользователю `www-data` запись в `/var/log/faillog`, выполнив команду:

```
setfacl -m u:www-data:rw /var/log/faillog
```

Выполнить перезапуск сервера:

```
systemctl restart apache2
```

11.4. Настройка для работы со службой FreeIPA

Для обеспечения работы веб-сервера Apache2 со службой FreeIPA следует произвести настройку веб-сервера и контроллера домена FreeIPA. Порядок действий по настройке описан в 8.2.13.

12. ЗАЩИЩЕННАЯ ГРАФИЧЕСКАЯ ПОДСИСТЕМА

В ОС используется защищенная графическая подсистема, основанная на использовании оконной системы X Window System (реализация X.Org[1)]) со встроенной мандатной защитой.

Для установки пакетов графической подсистемы следует в процессе работы программы установки ОС отметить в окне «Выбор программного обеспечения» строку «Рабочий стол Fly».

Графический вход в систему осуществляется при помощи утилит `fly-dm` (запуск серверной части системы) и `fly-qdm` (поддержка графического интерфейса), переход к которым происходит после окончания работы загрузчика. Утилиты обеспечивают загрузку графической среды для работы в системе, соединение с удаленным XDMCP-сервером, а также завершение работы системы.

После установки ОС значения параметров графического входа устанавливаются по умолчанию. Изменение установленных значений осуществляется с помощью утилиты `fly-admin-dm` («Вход в систему»), запущенной от имени администратора. Описание утилиты приведено в электронной справке.

12.1. Конфигурирование менеджера окон и рабочего стола в зависимости от типа сессии

Выбор режима рабочего стола Fly выполняется в меню «Тип сессии» в окне графического входа в систему (утилита `fly-dm`). По умолчанию предусмотрено несколько режимов, но администратор системы может добавить новые режимы, например, для систем с низкими характеристиками производительности или удаленных терминалов можно создавать режим `fly-light` и т. д.

Для создания нового режима необходимо добавить файл (файлы) сессии с расширением `desktop` в `/usr/share/fly-dm/sessions` и создать соответствующие конфигурационные файлы для `fly-wm`.

При входе через `fly-dm` выставляется переменная `DESKTOP_SESSION=имя_режима`, например `fly`, `fly-desktop`, `fly-tablet`). Данная переменная является именем ярлыка сессии из `/usr/share/fly-dm/sessions` (но без расширения `.desktop`), которая указывает на тип сессии. Например:

`DESKTOP_SESSION=fly` — десктопный

`DESKTOP_SESSION=fly-tablet` — планшетный

[1)] Недоступно в режиме «Мобильный».

Данное имя сессии добавляется как суффикс «.$DESKTOP_SESSION» к базовому имени конфигурационного файла и используется для выбора конфигурационных файлов менеджера окон fly-wm в соответствии с типом сессии.

Если тип сессии десктопный, т. е. DESKTOP_SESSION=fly, то конфигурационные файлы остаются без суффикса для обратной совместимости.

Существуют следующие конфигурационные файлы в /usr/share/fly-wm/:

```
apprc
apprc.fly-mini
apprc.fly-tablet
en.fly-wmrc
en.fly-wmrc.fly-mini
en.fly-wmrc.fly-tablet
en.miscrc
en.miscrc.fly-mini
en.miscrc.fly-tablet
keyshortcutrc
keyshortcutrc.fly-mini
keyshortcutrc.fly-tablet
ru_RU.UTF-8.fly-wmrc
ru_RU.UTF-8.fly-wmrc.fly-mini
ru_RU.UTF-8.fly-wmrc.fly-tablet
ru_RU.UTF-8.miscrc
ru_RU.UTF-8.miscrc.fly-mini
ru_RU.UTF-8.miscrc.fly-tablet
sessrc
sessrc.fly-mini
sessrc.fly-tablet
theme/default.themerc
theme/default.themerc.fly-mini
theme/default.themerc.fly-tablet
```

Также есть конфигурационный файл fly-wmrc.mini, который служит для совместимости и включает все файлы с расширением *.fly-mini. Названия этих файлов определяют их назначение, а в комментариях в файлах приведены особенности использования.

При использовании файлов типа:

```
~/.fly/*rc
~/.fly/theme/*rc
/usr/share/fly-wm/*rc
/usr/share/fly-wm/theme/*rc
```

необходимо переделать формирование имени конфигурационного файла. Например, это сделано в утилитах `fly-admin-theme`, `fly-admin-hotkeys`, `fly-admin-winprops` и др.

В ярлыках в полях `NotShowIn` и `OnlyShowIn` можно использовать имена типов сессий (`fly`, `fly-tablet`). Функция `FlyDesktopEntry::isDisplayable()` из `libflycore` изменена с учетом нахождения в сессии какого-либо типа (`$DESKTOP_SESSION`), также в `libflycore` добавлены:

```
const char * flySessionName()
const char * flySessionConfigSuffix()
```

Используя имена типов сессий в `NotShowIn` и `OnlyShowIn`, можно скрывать/показывать определенные ярлыки из меню «Пуск», панели задач или автозапуска (в зависимости от текущего режима).

Если у какой-либо Qt-программы есть сохраняемые/восстанавливаемые параметры, «чувствительные» к типу сессии (планшет, десктоп и т. д.), то программа будет иметь такие параметры в отдельных экземплярах для каждого типа сессии, добавляя, например, суффиксы `$DESKTOP_SESSION` к именам параметров.

12.2. Рабочий стол как часть экрана

В файлах `*themerc` (прежде всего в `~/.fly/theme/current.themerc`) можно задавать параметры `FlyDesktopWidth` и `FlyDesktopHeight`, которые определяют размер (в пикселях) рабочего стола на экране. Это может быть полезно, например, для:

- деления широкоформатного монитора на две части: с рабочим столом и свободной областью, куда можно перетаскивать окна;
- для задания области рабочего стола только на левом мониторе в двухмониторной конфигурации с Xinerama.

12.3. Удаленный вход по протоколу XDMCP

По умолчанию в системе удаленный вход по протоколу XDMCP запрещен. Чтобы его разрешить необходимо:

1) в файле `/etc/X11/fly-dm/Xaccess` заменить `localhost` на символ `*`;
2) в файле `/etc/X11/fly-dm/fly-dmrc` убедиться, что `Enable=true`:

```
...
[Xdmcp]
Enable=true
...
```

12.4. Решение возможных проблем с видеодрайвером Intel

Видеодрайвер для систем на базе процессоров Intel может в некоторых случаях устранять возможные проблемы в работе графической подсистемы, например искажения на экране или отказ X-сервера. В ряде случаев это может быть вызвано типом используемого ускорения графики. По умолчанию в драйвере включен тип ускорения SNA. Для использования более старого, но более стабильного UXA можно в `/usr/share/X11/xorg.conf.d` разместить файл `10-intel.conf`:

```
Section "Device"
Identifier "intel"
Driver "intel"
Option "AccelMethod" "uxa"
EndSection
```

12.5. Автоматизация входа в систему

Для включения автоматизации входа пользователя в систему на разных разрешенных ему уровнях конфиденциальности с последующим переключением между такими входами необходимо в секции `[Service]` файла `/lib/systemd/system/fly-dm.service` задать переменную:

```
...
Environment=DM_LOGIN_AUTOMATION=value
...
```

Затем на рабочих столах пользователя создать, например, следующие ярлыки:

- ярлык для запуска или перехода в сессию с меткой `0:0:0x0:0x0`:

```
[Desktop Entry]
Name = session 0
Name[ru] = Сессия 0
Type = Application
NoDisplay = false
Exec = fly-dmctl maclogin user password 0:0:0x0:0x0
Icon = ledgreen
X-FLY-IconContext = Actions
Hidden = false
Terminal = false
StartupNotify = false
```

- ярлык для запуска или перехода в сессию с меткой `1:0:0x0:0x0`:

```
[Desktop Entry]
Name = session 1
Name[ru] = Сессия 1
Type = Application
NoDisplay = false
Exec = /usr/bin/fly-dmctl maclogin user password 1:0:0x0:0x0
Icon = ledyellow
X-FLY-IconContext = Actions
Hidden = false
Terminal = false
StartupNotify = false
```

- ярлык для запуска или перехода в сессию с меткой `2:0:0x0:0x0`:

```
[Desktop Entry]
Name = session 2
Name[ru] = Сессия 2
Type = Application
NoDisplay = false
Exec = fly-dmctl maclogin user password  2:0:0x0:0x0
Icon = ledred
X-FLY-IconContext = Actions
Hidden = false
Terminal = false
StartupNotify = false
```

С помощью ярлыков данного типа пользователь сможет максимально переключаться между сессиями с разными метками безопасности.

12.6. Рабочий стол Fly

В состав рабочего стола Fly входит оконный менеджер и графические утилиты, которые могут быть использованы для администрирования ОС. Большинство утилит представляет собой графические оболочки соответствующих утилит командной строки.

Основные графические утилиты для настройки и администрирования системы приведены в таблице 46.

Таблица 46

Утилита	Описание
`fly-admin-autostart` «Автозапуск»	Управление автозапуском программ, автоматическим открытием файлов и каталогов при входе в сессию

Продолжение таблицы 46

Утилита	Описание
`fly-admin-dm` «Вход в систему»	Настройка графического входа в сессию
`fly-admin-date` «Дата и время»	Просмотр времени и календаря, настройка формата отображения даты и времени
`fly-admin-time` «Синхронизация времени»	Настройка синхронизации времени
`fly-admin-grub2` «Загрузчик GRUB2»	Графическая утилита настройки загрузчика ОС GRUB 2
`systemdgenie` «Инициализация системы»	Управление службой инициализации системы Systemd
`synaptic` «Менеджер пакетов Synaptic»	Графическая утилита управления пакетами
`gufw` «Настройка межсетевого экрана»	Настройка межсетевого экрана UFW (Uncomplicated Firewall)
`fly-admin-reflex` «Обработка «горячего» подключения»	Настройка действий, выполняемых при подключении устройств
`fly-admin-env` «Переменные окружения»	Добавление, изменение и удаление переменных окружения
`astra-systemsettings` «Параметры системы»	Управление системой и локальной политикой безопасности. Программа состоит из модулей, которые позволяют настраивать пользовательское окружение и оборудование, управлять пользователями и группами, а также настраивать функции безопасности: мандатное управление доступом, мандатный контроль целостности, права доступа, привилегии, политики, регистрацию событий и правила подключения устройств, ограничивать программную среду и др.
`fly-mimeapps` «Приложения для типов файлов»	Просмотр доступных приложений и установка приложения по умолчанию для типов файлов
`fly-admin-printer` «Принтеры»	Управление принтерами, настройка печати и управление заданиями на печать
`fly-admin-policykit-1` «Санкции PolicyKit-1»	Просмотр, предоставление и аннулирование санкций на выполнение привилегированных действий, управляемых с использованием PolicyKit-1
`fly-admin-session` «Сессии Fly»	Настройки параметров входа и выхода из сессий пользователя
`nm-connection-editor` «Сетевые соединения»	Настройка сетевых соединений
`fly-admin-network` «Параметры сети»	Управление автозапуском сетевых служб

Продолжение таблицы 46

Утилита	Описание
`fly-admin-alternatives` «Системные альтернативы»	Управление системой альтернатив дистрибутивов, основанных на Debian
`fly-admin-kiosk` «Системный киоск»	Управление ограничением среды пользователя
`hp-setup` «Установка принтеров, факсов и сканеров HP»	Установка новых устройств HP
`hp-plugin` «Установка дополнительного плагина HP»	Установка драйверов HP
`fly-admin-power` «Электропитание»	Настройка и управление параметрами электропитания и энергосбережения
`ksystemlog` «Просмотр системных журналов Ksystemlog»	Просмотр журнала расширенной системы протоколирования
`system-config-audit` «Конфигурация аудита»	Настройки аудита системы
`fly-sosreport` «Центр системных отчетов»	Сбор данных о конфигурации системы и работе подсистем для последующей диагностики
`fly-run` «Запуск приложения»	Запуск программы или осуществление доступа к ресурсу из командной строки, в том числе от имени другого пользователя и/или с другими мандатными атрибутами
«Контроль целостности файлов»	Графическая утилита программы afick монитора изменений файлов системы
`fly-admin-device-manager` «Менеджер устройств»	Просмотр доступных устройств, настройка их драйверов и параметров
`fly-admin-usbip` «Сервис удаленных USB-накопителей»	Предоставление удаленного доступа к USB-носителям и токенам с помощью USB-over-IP.
`fly-admin-format` «Форматирование внешнего носителя»	Удаление данных и форматирование внешнего носителя и его разделов
`fly-admin-iso` «Запись ISO образа на USB носитель»	Программа записи iso-образа на USB-носитель
`fly-admin-int-check` «Проверка целостности системы»	Проверка целостности системы для рабочего стола Fly
`fly-admin-marker` «Редактор маркеров»	Просмотр и изменение настроек маркировки печати, редактирование шаблонов маркера
`fly-print-station` «Управление печатью документов»	Маркировка и печать документов ограниченного доступа с возможностью перемещений заданий на другой принтер

Продолжение таблицы 46

Утилита	Описание
`gparted` «Редактор разделов Gparted»	Создание, перераспределение или удаление системных разделов ОС
`ksysguard` «Системный монитор»	Просмотр информации о процессах и общей загрузке системы: ЦПУ, памяти, раздела подкачки и сети
`konsole` «Терминал»	Эмулятор терминала, позволяющий взаимодействовать с консолью
`mc` «Менеджер файлов MC»	Просмотр папок и элементов ФС, выполнение основных функций управления файлами, обращение к сетевым ресурсам, работа с архивами
`kgpg` «Управление ключами KGpg»	Управление ключами GPG
`fly-admin-ad-client` «Настройка клиента Active Directory Fly»	Ввод клиентского компьютера в существующий домен AD Windows
`fly-admin-ad-server` «Настройка сервера Active Directory Fly»	Запуск службы контроллера домена AD
`fly-admin-ad-sssd-client` «Настройка клиента SSSD Fly»	Ввод клиентского компьютера в существующий домен AD Windows, при этом будет задействована служба управления аутентификацией и авторизацией (System Security Services Daemon (SSSD))
`fly-admin-freeipa-server` «Настройка FreeIPA server Fly»	Установка и настройка сервера FreeIPA
`fly-admin-freeipa-client` «Настройка FreeIPA client Fly»	Ввод клиентского компьютера в существующий домен FreeIPA
`fly-admin-multiseat` «Мультитерминальный режим»	Подготовка компьютера для одновременной работы нескольких пользователей
`fly-admin-dhcp` «Настройка DHCP-сервера»	Настройка сервера DHCP
`fly-admin-openvpn-server` «Настройка OpenVPN сервера Fly»	Настройка сервера VPN
`fly-admin-ftp` «FTP-сервер»	Настройка сервера FTP
`fly-admin-samba` «Общие папки (Samba)»	Управление общими папками Samba
`fly-passwd` «Изменить пароль»	Смена пароля
`fly-su` «Подмена пользователя»	Выполнение команды от имени другого пользователя
`fly-astra-update` «Установка обновлений»	Программа установки обновлений

Окончание таблицы 46

Утилита	Описание
`fly-admin-repo` «Редактор репозиториев»	Создание и управление репозиториями
`fly-admin-driver` «Управление драйверами»	Установка и выбор графических драйверов

Не все из приведенных в таблице 46 утилит устанавливаются по умолчанию при установке ОС. Описание утилит доступно в электронной справке. Вызов электронной справки осуществляется с помощью ярлыка «Помощь», размещенного на рабочем столе, а также путем нажатия комбинации клавиш **<Alt+F1>** или путем нажатия клавиши **<F1>** в активном окне графической утилиты.

12.7. Блокировка экрана при бездействии

Блокировка экрана при неактивности задается в конфигурационных файлах типов сессий `*themerc*`, расположенных в каталоге пользователя `/home/<имя_пользователя>/.fly/theme/`, следующими параметрами:

```
ScreenSaverDelay=0/<время_неактивности_в_секундах>
LockerOnSleep=true/false
LockerOnDPMS=true/false
LockerOnLid=true/false
LockerOnSwitch=true/false
```

При этом имена актуальных для сессии пользователя конфигурационных файлов начинаются с `current`, а файлы, имена которых начинаются с `default`, используются для создания и восстановления файлов `current`.

При создании учетной записи пользователя и его первом входе конфигурационные файлы `default.themerc*` копируются из каталога `/usr/share/fly-wm/theme/` в каталог пользователя `/home/<имя_пользователя>/.fly/theme/`.

Пользователю доступно управление блокировкой экрана своей сессии при неактивности из графической утилиты `fly-admin-theme` (см. электронную справку).

Администратору для управления блокировкой экрана пользователей, в т.ч. централизованного, доступен конфигурационный файл `/usr/share/fly-wm/theme.master/themerc`. В файле указываются строки:

```
[Variables]
ScreenSaverDelay=0/<время_неактивности_в_секундах>
```

```
LockerOnSleep=true/false
LockerOnDPMS=true/false
LockerOnLid=true/false
LockerOnSwitch=true/false
```

При входе пользователя в сессию после считывания параметров из конфигурационных файлов пользователя проверяется наличие файла `/usr/share/fly-wm/theme.master/themerc` с секцией `[Variables]`. При наличии файла из него считываются параметры, и считанные параметры переопределяют аналогичные параметры, считанные ранее из конфигурационных файлов пользователя.

В ОС выполняется мониторинг каталога `/usr/share/fly-wm/theme.master/` и файла `/usr/share/fly-wm/theme.master/themerc`. При создании/изменении файла `/usr/share/fly-wm/theme.master/themerc` срабатывает механизм мониторинга и параметры из файла считываются и применяются к текущим сессиям всех пользователей.

Каталог `/usr/share/fly-wm/theme.master/` может являться разделяемым ресурсом.

Пользователю недоступна возможность переопределить параметры, заданные в `/usr/share/fly-wm/theme.master/themerc`.

12.8. Мандатное управление доступом

Мандатная защита, встроенная в рабочий стол Fly и устанавливаемая по умолчанию вместе с ОС, позволяет администратору задавать отдельно для каждого пользователя разрешенный диапазон иерархических уровней конфиденциальности и неиерархических категорий конфиденциальности. Для этой цели следует использовать модуль «Пользователи» в разделе «Пользователи и группы» графической утилиты `astra-systemsettings` («Параметры системы», см. электронную справку). Для вызова модуля можно использовать команду:

```
astra-systemsettings astra_kcm_users
```

После того как пользователь, для которого установлены возможные иерархические уровни конфиденциальности и неиерархические категории конфиденциальности, отличные от нуля, войдет в систему, ему будет предложено установить конкретный иерархический уровень конфиденциальности и конкретную неиерархическую категорию конфиденциальности для данной сессии в пределах разрешенных диапазонов. Выбранные значения этих параметров отображаются на цветном индикаторе с числом внутри, расположенном в области уведомлений на панели задач. Для получения информационного сообщения следует навести курсор на индикатор (рис. 8).

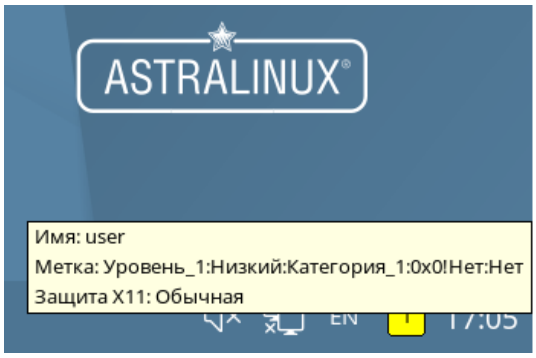
ASTRALINUX
Имя: user
Метка: Уровень_1:Низкий:Категория_1:0x0!Нет:Нет
Защита X11: Обычная
EN
17:05

Рис. 8

13. ГРАФИЧЕСКАЯ ПОДСИСТЕМА РЕЖИМА «МОБИЛЬНЫЙ»

Графическая подсистема режима «Мобильный» реализована на основе протокола Wayland, в качестве оконного менеджера используется KWin.

В графическом интерфейсе пользователя для создания окружения рабочего стола используется KDE Plasma Mobile (для мобильного вида) и KDE Plasma (для десктопного вида).

13.1. Отображение графического интерфейса

В режиме «Мобильный» графический интерфейс адаптирован для использования на устройствах, оснащенных сенсорным устройством указания на чувствительной области экрана дисплея при помощи прикосновения (типа «touch-screen»).

Данный режим поддерживает отображение графического интерфейса в мобильном виде и в десктопном виде. Мобильный вид используется по умолчанию при использовании ОС в режиме «Мобильный». Из мобильного вида возможно выполнить переход в десктопный вид. При подключении к устройству монитора, клавиатуры и мыши десктопный вид может быть использован в роли ПЭВМ для администрирования и настройки ОС на устройстве.

13.2. Автоматизация входа в систему

Для включения автоматического входа пользователя в систему используется конфигурационный файл `~/.config/kscreenlockerrc`, в котором задан параметр:

```
[<имя_пользователя>]
PermitEmptyPasswords=true
```

Параметр позволяет входить пользователю с пустым паролем.

Шаблон конфигурационного файла, который будет использоваться при создании конфигурационного файла для каждого пользователя, можно создать в `/etc/xdg/kscreenlockerrc` с необходимыми параметрами (доступные параметры приведены в `kscreenlocker/settings/kscreenlockersettings.kcfg`).

Пример

```
[Version]
update_info=kscreenlocker.upd:0.1-autolock

[User]
PermitEmptyPasswords=true
```

Настройка автоматического входа в систему также может быть настроена через панель быстрого доступа, кнопка **[Безопасность]**, пункт «Общие» (см. электронную справку «Документация — Графический интерфейс — Режим «Мобильный»).

13.3. Рабочий стол

В состав рабочего стола KDE Plasma входит оконный менеджер и приложения (в т. ч. адаптированные для работы на устройствах с сенсорным экраном), которые могут быть использованы для администрирования ОС. Перечень основных приложений приведен в таблице 47.

Таблица 47

Приложение	Описание
`plasma-settings` «Настройки»	Настройка и администрирование системы
`fly-admin-cron` «Планировщик задач»	Установка расписания задач для выполнения в фоновом режиме, настройка среды выполнения задачи (переменных окружения), разрешение или запрет на выполнение уже установленной задачи
`mc` «Менеджер файлов MC»	Просмотр папок и элементов ФС, выполнение основных функций управления файлами, обращение к сетевым ресурсам, работа с архивами
`gufw` «gufw»	Программа настройки межсетевого экрана UFW (Uncomplicated Firewall)
`fly-admin-autostart` «Автозапуск»	Управление автозапуском приложений, автоматическим открытием файлов и каталогов при загрузке рабочего стола
`fly-admin-device-manager` «Менеджер устройств»	Просмотр доступных устройств, настройка их драйверов и параметров
`fly-admin-policykit-1` «Санкции PolicyKit-1»	Просмотр, предоставление и аннулирование санкций на выполнение привилегированных действий, управляемых с использованием PolicyKit-1
`nm-connection-editor` «Сетевые соединения»	Настройки сетевых соединений (по умолчанию при загрузке системы выполняется автозапуск программы)
`fly-event-viewer` «Журнал системных событий»	Просмотр записей в журнале системных событий, печать и экспорт записей
`ksysguard` «Системный монитор»	Отслеживание системных параметров
`fly-term` «Терминал Fly»	Эмулятор терминала, позволяющий взаимодействовать с консолью
`ksystemlog` «Журнал аудита»	Просмотр журнала расширенной системы протоколирования

Окончание таблицы 47

Приложение	Описание
`hp-setup` «Установка принтеров, факсов и сканеров HP»	Установка новых устройств HP
`fly-admin-printer` «Принтеры»	Добавление, настройка, удаление и просмотр информации о принтерах, настройка печати и управление заданиями на печать
`fly-admin-format` «Форматирование внешнего носителя»	Удаление данных и форматирование внешнего носителя и его разделов
`hp-plugin` «Установка дополнительного плагина HP»	Установка драйверов HP
`fly-admin-env` «Переменные окружения»	Добавление, изменение и удаление переменных окружения
`systemdgenie` «Инициализация системы»	Управление службой инициализации системы Systemd
`fly-run` «Запуск приложения»	Запуск программы или осуществление доступа к ресурсу из командной строки, в т.ч. от имени другого пользователя и/или с другими мандатными атрибутами
`system-config-audit` «Конфигурация аудита»	Проверка и изменение статуса и настроек системы аудита

Описание приложений доступно в справке, вызываемой из приложения, или в электронной справке, вызываемой с помощью ярлыка «Помощь», размещенного на экране приложений.

14. ЗАЩИЩЕННЫЙ КОМПЛЕКС ПРОГРАММ ПЕЧАТИ И МАРКИРОВКИ ДОКУМЕНТОВ

Одной из основных служб, предоставляемых ОС, является служба печати, модифицированная для маркировки документов и позволяющая осуществлять печать документов в соответствии с требованиями, предъявляемыми к защищенным ОС.

Защищенный комплекс программ печати и маркировки документов CUPS обеспечивает:

- управление заданиями на печать;
- выполнение команд администратора печати;
- предоставление информации о состоянии принтеров локальным и удаленным программам;
- выдачу информационных сообщений пользователям;
- маркировку выводимых на печать документов.

14.1. Устройство системы печати

Состав защищенного комплекса программ печати и маркировки приведен в таблице 48.

Таблица 48

Название	Пакет	Описание
CUPS	`cups-daemon`	Сервер печати. Обрабатывает запросы от пользователя и выполняет запуск служебных программ
`fly-admin-printer`	`fly-admin-printer`	Графическая утилита для настройки принтеров и сервера печати CUPS, а также управления очередью печати. При установленном пакете `fly-admin-printer-mac` позволяет маркировать документы и настраивать метки безопасности принтера
`fly-print-monitor`	`fly-print-monitor`	Графическая утилита для отслеживания состояния принтеров и сервера печати
`fly-jobviewer`	`fly-jobviewer`	Графическая утилита для управления очередью печати. При установленном пакете `fly-admin-printer-mac` позволяет также маркировать документы

Окончание таблицы 48

Название	Пакет	Описание
psmarker	parsec-cups	Программа для маркировки документов в формате PostScript. Модифицирует исходный файл задания, добавляя в него маркеры. Запускается с помощью CUPS
fonarik	parsec-cups	Программа для создания файла маркировки на обратной стороне последнего листа. Запускается с помощью CUPS
markerdb	parsec-cups	Программа для записи журнала маркировки. Вызывается из CUPS после завершения задания маркировки. В процессе маркировки не участвует
pdfhelper	parsec-cups	Программа для определения размера и ориентации PDF-документов. Запускается с помощью CUPS перед маркировкой
markjob	parsec-cups-client	Инструмент для маркировки документов в консольном режиме
libfly-admin-printer-mac	libfly-admin-printer-mac3	Библиотека с функциями маркировки и просмотра журнала для графических клиентов. Упрощает взаимодействие с сервером CUPS при выполнении задач маркировки
fly-admin-printer-mac	fly-admin-printer-mac	Утилита, добавляющая функции маркировки в графические утилиты fly-admin-printer и fly-jobviewer
fly-print-station	fly-print-station	Графическая утилита для маркировки документов
fly-admin-marker	fly-admin-marker	Графическая утилита для редактирования шаблонов маркировки. Работает только локально вместе с сервером печати CUPS

Планировщик — это сервер, который управляет списком доступных принтеров и направляет задания на печать, используя подходящие фильтры и выходные буферы (backends).

Файлами конфигурации являются:

- файл конфигурации сервера;

- файлы определения принтеров и классов;
- типы MIME и файлы правил преобразования;
- файлы описания PostScript-принтеров (PPD).

Конфигурационный файл сервера похож на файл конфигурации веб-сервера и определяет все свойства управления доступом.

Файлы описания принтеров и классов перечисляют доступные очереди печати и классы. Классы принтеров — наборы принтеров. Задания, посланные классу принтеров, направляются к первому доступному принтеру данного класса.

Очередь печати — механизм, который позволяет буферизовать и организовать задания, посылаемые на принтер. Необходимость организации такого механизма обуславливается тем, что принтер является медленно действующим устройством, и задания не могут быть распечатаны мгновенно. Очевидно, что в многопользовательской среде возникает конкуренция со стороны пользователей при доступе к принтерам, поэтому задания необходимо располагать в очереди. Для этого используется буферный каталог `/var/spool/cups/`.

Файлы типов MIME перечисляют поддерживаемые MIME-типы (`text/plain`, `application/postscript` и т. д.) и правила для автоматического обнаружения формата файла. Они используются сервером для определения поля `Content-Type` для GET- и HEAD-запросов и обработчиком запросов IPP, чтобы определить тип файла.

Правила преобразования MIME перечисляют доступные фильтры. Фильтры используются, когда задание направляется на печать, таким образом, приложение может послать файл удобного (для него) формата системе печати, которая затем преобразует документ в требуемый печатный формат. Каждый фильтр имеет относительную «стоимость», связанную с ним, и алгоритм фильтрования выбирает набор фильтров, который преобразует файл в требуемый формат с наименьшей общей «стоимостью».

Файлы PPD описывают возможности всех типов принтеров. Для каждого принтера имеется один PPD-файл. Файлы PPD для не-PostScript-принтеров определяют дополнительные фильтры посредством атрибута `cupsFilter` для поддержки драйверов принтеров.

В ОС стандартным языком описания страниц является язык PostScript. Большинство прикладных программ (редакторы, браузеры) генерируют программы печати на этом языке. Когда необходимо напечатать ASCII-текст, программа печати может быть ASCII-текстом. Имеется возможность управления размером шрифтов при печати ASCII-текста. Управляющая информация используется для контроля доступа пользователя к принтеру и аудита печати. Также имеется возможность печати изображений в форматах GIF, JPEG, PNG, TIFF и документов в формате PDF.

Фильтр — программа, которая читает из стандартного входного потока или из файла, если указано его имя. Все фильтры поддерживают общий набор параметров, включающий

имя принтера, идентификатор задания, имя пользователя, имя задания, число копий и параметры задания. Весь вывод направляется в стандартный выходной поток.

Фильтры предоставлены для многих форматов файлов и включают, в частности, фильтры файлов изображения и растровые фильтры PostScript, которые поддерживают принтеры, не относящиеся к типу PostScript. Иногда несколько фильтров запускаются параллельно для получения требуемого формата на выходе.

Программа `backend` — это специальный фильтр, который отправляет печатаемые данные устройству, в т. ч. через сетевое соединение. В состав системы печати включены фильтры для поддержки устройств, подключаемых с помощью параллельного и последовательного интерфейсов, а также шины USB.

Клиентские программы используются для управления заданиями и сервером печати.

Управление заданиями включает:

- формирование;
- передачу серверу печати;
- мониторинг и управление заданиями в очереди на печать.

Управление сервером включает:

- запуск/остановку сервера печати;
- разрешение/запрет постановки заданий в очередь;
- разрешение/запрет вывода заданий на принтер.

В общем случае вывод данных на принтер происходит следующим образом:

1) программа формирует запрос на печать задания к серверу печати;

2) сервер печати принимает подлежащие печати данные, формирует в буферном каталоге файлы с содержимым задания и файлы описания задания, при этом задание попадает в соответствующую очередь печати;

3) сервер печати просматривает очереди печати для незанятых принтеров, находит в них задания и запускает конвейер процессов, состоящий из фильтров и заканчивающийся выходным буфером, информация из которого поступает в принтер посредством драйверов ОС;

4) контроль и мониторинг процесса печати выполняется с помощью программ `lpq`, `lpc`, `lprm`, `lpstat`, `lpmove`, `cancel`, а также с помощью графической утилиты `fly-admin-printer`.

Система печати ОС решает следующие задачи:

1) монопольная постановка задания в очередь на печать. Данная функция предполагает невозможность вывода документа на печать в обход системы печати;

2) маркировка каждого напечатанного листа. Каждый лист сопровождается автоматической маркировкой (учетными атрибутами документа).

ВНИМАНИЕ! Для обеспечения штатной работы пользователя с сетевыми службами должна быть явно задана его метка безопасности (диапазон уровней конфиденциальности, категорий конфиденциальности и меток целостности) с помощью соответствующих утилит, даже если ему не доступны уровни и категории выше 0. Дополнительная информация приведена в документе РУСБ.10015-01 97 01-1.

14.2. Установка комплекса программ печати

Основные компоненты защищенного комплекса программ печати и маркировки документов устанавливаются автоматически при установке ОС.

В случае необходимости возможно вручную установить защищенный комплекс программ печати и маркировки документов, выполнив команду:

```
apt install fly-print-station parsec-cups-client fly-admin-printer \
     fly-admin-printer-mac fly-admin-marker
```

14.3. Настройка комплекса программ печати

Настройка защищенного комплекса программ печати и маркировки документов выполняется путем корректировки конфигурационных файлов `/etc/cups/cupsd.conf` и `/etc/cups/cups-files.conf`. Копии конфигурационных файлов, устанавливаемые вместе с пакетом, размещаются в `/usr/share/cups` (файлы `cupsd.conf.default` и `cups-files.conf.default`), данные файлы могут использоваться при необходимости восстановить комплекс программ печати и маркировки документов в исходное состояние.

Предварительная настройка защищенного комплекса программ печати и маркировки документов должна выполняться от имени учетной записи администратора с использованием механизма `sudo`.

Ряд действий по администрированию CUPS (добавление и удаление принтеров, изменение политики для принтера, установка мандатных атрибутов для принтера) может выполняться от имени пользователя, входящего в локальную группу администраторов печати `lpadmin`. Данная группа администраторов печати указана в качестве значения параметра `SystemGroup` в файле `/etc/cups/cups-files.conf`.

Основные пользовательские настройки содержатся в файлах конфигурации `client.conf` и `~/.cups/lpoptions`.

14.3.1. Настройка для работы с локальной базой безопасности

Для разрешения серверу CUPS удаленно принимать задания и команды необходимо от имени учетной записи администратора через механизм `sudo`:

1) выполнить следующие команды:

```
cupsctl --remote-admin --share-printers --remote-any
cupsctl ServerAlias=*
cupsctl DefaultPolicy=authenticated
cupsctl DefaultAuthType=Basic
```

2) осуществить перезапуск сервера системы печати, выполнив команды:

```
systemctl stop cups
systemctl start cups
```

В файле конфигурации клиента `client.conf` должен быть задан один параметр `ServerName`, определяющий имя сервера печати, например:

```
ServerName computer.domain
```

14.3.2. Настройка для работы в ЕПП

Для работы системы печати в ЕПП необходимо выполнение следующих условий:

1) наличие в системах, на которых функционируют сервер и клиенты системы печати, установленного пакета клиента FreeIPA — `client.domain.ipa`;

2) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, `myserver.example.ru`);

3) клиент FreeIPA должен быть настроен на используемый FreeIPA домен в соответствии с 8.2.6.

Для проведения операций по настройке FreeIPA и администрированию Kerberos необходимо знание паролей администраторов FreeIPA и Kerberos.

14.3.2.1. Настройка сервера печати

Для выполнения действий по управлению принтерами и очередями печати необходимо создать в FreeIPA учетную запись администратора печати `ipa_print_admin` и добавить ее в локальную группу администраторов печати на сервере печати, выполнив команду:

```
sudo gpasswd -a ipa_print_admin lpadmin
```

Для обеспечения совместной работы сервера печати с FreeIPA необходимо:

1) на контроллере домена добавить службу `ipp`:

```
ipa service-add ipp/printserver.domain.ipa
```

2) выгрузить таблицу ключей для службы:

```
sudo ipa-getkeytab -p ipp/printserver.domain.ipa@DOMAIN.IPA -k
    /tmp/ipp.keytab
```

3) если сервер CUPS установлен не на контроллере домена, то необходимо перенести таблицу ключей на `printserver.domain.ipa` в `/tmp`:

```
sudo scp /tmp/ipp.keytab admin@printserver.domain.ipa:/tmp
```

4) на компьютере, где установлен сервер CUPS, добавить ключи в хранилище Kerberos:

```
admin@printserver:~$ sudo ktutil
ktutil:  rkt /tmp/ipp.keytab
ktutil:  wkt /etc/krb5.keytab
ktutil:  l
slot KVNO Principal
---- ---- ---------------------------------------------------------------------
1    1           ipp/printserver.domain.ipa@DOMAIN.IPA
2    1           ipp/printserver.domain.ipa@DOMAIN.IPA
ktutil:  q

admin@printserver:~$ sudo klist -kte /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
1 18.05.2020 12:10:17 host/printserver.domain.ipa@DOMAIN.IPA
    (aes256-cts-hmac-sha1-96)
1 18.05.2020 12:10:17 host/printserver.domain.ipa@DOMAIN.IPA
    (aes128-cts-hmac-sha1-96)
1 18.05.2020 13:10:27 ipp/printserver.domain.ipa@DOMAIN.IPA
    (aes256-cts-hmac-sha1-96)
1 18.05.2020 13:10:27 ipp/printserver.domain.ipa@DOMAIN.IPA
    (aes128-cts-hmac-sha1-96)
```

ВНИМАНИЕ! С включенной проверкой целостности администрировать сервер печати можно только локально.

Для настройки сервера печати CUPS от имени учетной записи администратора с использованием механизма sudo:

1) выполнить следующие команды:

```
cupsctl --remote-admin --share-printers --remote-any
cupsctl ServerAlias=*
cupsctl DefaultPolicy=authenticated
cupsctl ServerName=printserver.domain.ipa
cupsctl MacEnable=On
cupsctl DefaultAuthType=Negotiate
cupsctl MacFullyQualifiedNames=On
```

2) в конфигурационном файле /etc/cups/cupsd.conf заменить строки:

```
Port 631
Listen /var/run/cups/cups.sock
```

на строку:

```
Listen 0.0.0.0:631
```

3) осуществить перезапуск сервера системы печати, выполнив команду:

```
systemctl restart cups
```

ВНИМАНИЕ! В конфигурационном файле защищенного сервера печати из состава изделия /etc/cups/cupsd.conf не допускается установка значения None параметра DefaultAuthType (отключение аутентификации) и внесение изменений в параметры политики PARSEC, не соответствующих эксплуатационной документации.

Далее выполнить вход на сервере печати от имени учетной записи, входящей в локальную группу администраторов печати на сервере печати lpadmin, и настроить принтеры. Настройка принтеров может быть выполнена с использованием утилиты fly-admin-printer (см. электронную справку). После запуска утилиты необходимо указать, что для выполнения привилегированных действий не используется учетная запись root, и затем выполнять действия по настройке.

14.3.2.2. Настройка клиента системы печати

Для настройки клиента системы печати необходимо:

1) создать конфигурационный файл /etc/cups/client.conf;

2) задать в конфигурационным файле /etc/cups/client.conf для параметра ServerName в качестве значения имя сервера системы печати, например, printserver.domain.ipa.

14.3.3. Регистрация событий

Регистрация событий защищенного комплекса печати и маркировки выполняется в следующих журналах:

1) `/var/log/cups/error_log` — сообщения об ошибках сервера печати, принтеров или других программ печати защищенного комплекса;

2) `/var/log/cups/access_log` — запросы к серверу печати;

3) `/var/log/cups/page_log` — сообщения успешной обработки страниц задания на печать;

4) в журнале подсистемы регистрации событий из состава ОС (см. 17.2) — события заданий на печать (создано, завершено, отменено и т. д.), события маркировки и события изменения принтеров (добавлен, удален, изменен).

Регистрация в журналы `/var/log/cups/error_log`, `/var/log/cups/access_log` и `/var/log/cups/page_log` выполняется автоматически.

Включение и отключение регистрации событий в журнал подсистемы регистрации событий осуществляется в графической утилите `fly-admin-printer` (см. электронную справку) или в конфигурационном файле `/etc/cups/cupsd.conf` путем задания значения параметру `MacAudit` (`on` — регистрация включена, задано по умолчанию после установки пакета; `off` — регистрация отключена):

```
MacAudit on
```

14.4. Настройка принтера и управление печатью

14.4.1. Общие положения

Установку и настройку принтера следует производить после завершения установки и первоначальной настройки ОС.

При печати через локальный сервер печати данные сначала формируются на локальном сервере, как для любой другой задачи печати, после чего посылаются на принтер, подключенный к данному компьютеру.

Системные каталоги, определяющие работу системы печати ОС, содержат файлы, которые не являются исполняемыми и содержат необходимую для драйвера принтера информацию (используемое физическое устройство, удаленный компьютер и принтер для удаленной печати), а также файлы журнала:

- `/etc/cups/printers.conf` — описание принтеров в ОС;

- /etc/cups/ppd/<имя_очереди>.ppd — описание возможностей принтера, которые используются при печати заданий и при настройке принтеров;

- /var/log/cups/error_log — файл журнала, содержащий сообщения об ошибках сервера печати, принтера или других программ системы печати;

- /var/log/cups/access_log — файл журнала, содержащий все запросы к серверу печати;

- /var/log/cups/page_log — файл журнала, содержащий сообщения успешной обработки страниц задания фильтрами и принтером.

Далее термин «принтер» в настоящем подразделе используется для обозначения принтера, соответствующего одной записи в файле /etc/cups/printers.conf. Под термином «физический принтер» подразумевается устройство, с помощью которого производится вывод информации на бумажный носитель. В файле /etc/cups/printers.conf может быть несколько записей, описывающих один физический принтер различными способами.

14.4.2. Команды управления печатью

В систему печати ОС включены файлы, предоставляющие командный интерфейс пользователя в стиле BSD и System V. Перечень файлов приведен в таблице 49.

Таблица 49

Файл	Описание
/usr/bin/lpr	Постановка заданий в очередь. Совместима с командой lpr системы печати BSD UNIX
/usr/bin/lp	Постановка заданий в очередь. Совместима с командой lp системы печати System V UNIX
/usr/bin/lpq	Просмотр очередей печати
/usr/sbin/lpc	Управление принтером. Является частичной реализацией команды lpc системы печати BSD UNIX
/usr/bin/lprm	Отмена заданий, поставленных в очередь на печать
/usr/sbin/cupsd	Сервер печати
/usr/sbin/lpadmin	Настройка принтеров и классов принтеров
/usr/sbin/lpmove	Перемещение задания в другую очередь
/usr/bin/fly-admin-printer	Настройка системы печати, установка и настройка принтеров, управление заданиями

Описание данных команд приведено на страницах руководства man.

CUPS предоставляет утилиты командной строки для отправления заданий и проверки состояния принтера. Команды lpstat и lpc status также показывают сетевые принтеры (принтер@сервер), когда разрешен обзор принтеров.

Команды администрирования System V предназначены для управления принтерами и классами. Средство администрирования `lpc` поддерживается только в режиме чтения для проверки текущего состояния очередей печати и планировщика.

Остановить работу службы печати можно с помощью команды:

```
systemctl stop cups
```

Запустить службу печати можно с помощью команды:

```
systemctl start cups
```

14.4.2.1. lp

С помощью команды `lp` выполняется передача задачи принтеру, т. е. задача ставится в очередь на печать. В результате выполнения этой команды файл передается серверу печати, который помещает его в каталог `/var/spool/cups/`.

14.4.2.2. lpq

Команда `lpq` предназначена для проверки очереди печати, используемой LPD, и вывода состояния заданий на печать, указанных при помощи номера задания либо системного идентификатора пользователя, которому принадлежит задание (владельца задания). Команда выводит для каждого задания имя его владельца, текущий приоритет задания, номер задания и размер задания в байтах, без параметров выводит состояние всех заданий в очереди.

14.4.2.3. lprm

Команда `lprm` предназначена для удаления задания из очереди печати. Для определения номера задания необходимо использовать команду `lpq`. Удалить задание может только его владелец или администратор печати.

14.4.2.4. lpadmin

Команда `lpadmin` также используется для настройки принтера в ОС.

Ее запуск с параметром `-p` используется для добавления или модификации принтера:

```
/usr/sbin/lpadmin -p printer [параметры]
```

Основные параметры команды `lpadmin` приведены в таблице 50.

Таблица 50

Параметр	Описание
`-c class`	Добавляет названный принтер к классу принтеров `class`. Если класс не существует, то он создается
`-m model`	Задает стандартный драйвер принтера, обычно файл `PPD`. Файлы `PPD` обычно хранятся в каталоге `/usr/share/cups/model/`. Список всех доступных моделей можно вывести командой `lpinfo` с параметром `-m`
`-r class`	Удаляет указанный принтер из класса `class`. Если в результате класс становится пустым, он удаляется
`-v device-uri`	Указывает адрес устройства для связи с принтером
`-D info`	Выдает текстовое описание принтера
`-E`	Разрешает использование принтера и включает прием заданий
`-L location`	Выводит расположение принтера
`-P ppd-file`	Указывает локальный файл `PPD` для драйвера принтера

Для данной команды существуют также параметры по регулированию политики лимитов и ограничений по использованию принтеров и политики доступа к принтерам.

Запуск команды `lpadmin` с параметром `-x` используется для удаления принтера:

```
/usr/sbin/lpadmin -x printer
```

14.4.3. Графическая утилита настройки сервера печати

Утилита `fly-admin-printer` предназначена для настройки печати в графическом режиме. Позволяет в режиме администратора печати устанавливать, настраивать и удалять принтеры и классы принтеров, а также настраивать сервер печати и управлять заданиями на печать. В режиме обычного пользователя позволяет устанавливать настройки печати и параметры принтера, а также управлять заданиями на печать (удалять, приостанавливать, возобновлять печать и устанавливать отложенную печать). Для вызова привилегированных действий запрашивается авторизация. Подробную информацию по использованию утилиты `fly-admin-printer` см. в электронной справке.

Для установки драйверов принтеров производства Hewlett Packard рекомендуется использовать утилиту `hp-setup`.

14.5. Маркировка документа

При поступлении задания на печать считывается метка безопасности сетевого соединения и копируется в атрибут задания `mac-job-mac-label`.

При печати задания с нулевой меткой безопасности (нулевой уровень конфиденциальности, без категорий конфиденциальности и нулевая метка целостности) маркировка листов не выполняется и печать осуществляется в штатном режиме. При этом атрибут принтера `mac-printer-mac-min` должен быть нулевым, иначе задание на печать будет завершено с ошибкой.

При печати задания с ненулевой меткой безопасности оно принудительно переводится сервером печати в состояние «отложено» до проведения привилегированным пользователем маркировки выводимых на печать листов. Файлы заданий (в каталоге `/var/spool/cups`) маркируются согласно мандатному контексту документа.

ВНИМАНИЕ! Задания печати администратора печати (учетная запись, входящая в группу `lpadmin`), отправляются сразу на печать без задержки на маркировку.

Для печати заданий с ненулевой меткой безопасности необходимо соответствующим образом настроить принтер, а также маркеры печати (при необходимости). Описание настройки принтеров, маркеров печати и порядка маркировки приведено в РУСБ.10015-01 97 01-1.

ВНИМАНИЕ! Мандатный контекст задания должен находиться в диапазоне между минимальным и максимальным мандатным контекстом принтера, на который отправлено задание. Если метка безопасности задания ненулевая, но не попадает в множество разрешенных меток для данного принтера, заданных атрибутами `mac-printer-mac-min` и `mac-printer-mac-max`, то задание на печать будет завершено с ошибкой.

ВНИМАНИЕ! Контроль метки целостности работает только для локальных соединений (через Unix Domain Socket). Любому соединению по TCP/IP будет присваиваться нулевая метка целостности. Поэтому для возможности печати с удаленного компьютера необходимо разрешить принтеру печать с нулевой меткой целостности.

Маркировка осуществляется «наложением» маркеров с учетными атрибутами документа, включающими:

- уровень конфиденциальности документа;
- номер экземпляра;
- количество листов в экземпляре;
- дату вывода документа на печать;
- номер каждого входящего документа;
- имя исполнителя;
- имя пользователя, производившего печать.

Система печати является инвариантной по отношению к приложениям, которые обращаются к службе печати. Это означает, что приложения, выводящие на печать, должны учитывать

маркировку листов и оставлять для этого свободное место. В противном случае маркеры могут наложиться на фрагменты печатаемой информации.

Маркировка задания выполняется в пять этапов:

1) блокировка задания. Если задание в процессе маркировки другим пользователем или соединением, то выдается ошибка;
2) проверка наличия и установка атрибутов задания;
3) с помощью переменных маркировки запрос у пользователя атрибутов задания;
4) выставление атрибутов задания, полученных на предыдущем этапе;
5) непосредственно маркировка задания.

Маркировка документов при использовании локальной базы осуществляется от имени пользователя, входящего в группу `lpmac`. Если группа отсутствует в системе, она должна быть создана.

Маркировка документов в ЕПП осуществляется от имени доменного пользователя, входящего в локальную группу `lpmac` на сервере печати. Для добавления пользователя в локальную группу `lpmac` необходимо на сервере печати выполнить команду:

```
sudo gpasswd -a ipa_marker_user lpmac
```

Маркировка документа выполняется с помощью инструмента командной строки `markjob`, описанного в 14.6, или с помощью графической утилиты `fly-print-station`, описанной в 14.7.

14.6. Маркировка документа в командной строке

Маркировка документа в командной строке выполняется с помощью инструмента `markjob`. Инструмент `markjob` требует наличия утилиты `lpq`, входящей в состав пакета `cups-bsd`.

Для маркировки с помощью `markjob` выполнить команду:

```
markjob -m
```

или

```
markjob
```

Подробное описание инструмента `markjob` приведено в `man markjob`.

В процессе работы инструмента `markjob` у пользователя запрашиваются настроенные атрибуты для маркера печати, например:

- `mac-inv-num` — инвентарный номер;
- `mac-owner-phone` — телефон исполнителя;
- `mac-workplace-id` — идентификатор рабочего места;
- `mac-distribution` — список рассылки.

При вводе списка рассылки адреса разделяются символом «^». Если в значении списка рассылки используется пробел, то значение атрибута необходимо взять в кавычки целиком.

Пример

Выдается запрос на ввод списка рассылки:

```
Enter mac-distribution - Distribution list, addresses separated by '^':
```

Ввести список рассылки:

```
"В дело^В адрес"
```

После выполнения маркировки в очереди формируются два дополнительных задания в состояние «отложено»: первое (с меньшим номером) представляет собой промаркированный документ, а второе (с большим номером) — размещаемую на обороте последнего листа документа маркировку.

Для печати промаркированного документа необходимо возобновить печать первого отложенного задания. Затем на обороте последнего листа документа печатается маркировка путем возобновления выполнения второго дополнительного задания.

При выполнении маркировки от имени пользователя, входящего в группу `lpmac`, возможно получение сообщения:

```
Невозможно выполнить запрос: запрещено
```

В данном случае необходимо выполнить команду `id` от имени пользователя, выполняющего маркировку, и повторно запустить инструмент `markjob`.

Если ведение журнала маркировки включено, то после завершения задания данные маркировки будут записаны в него. Описание журнала маркировки приведено в 14.9.

14.7. Графическая утилита управления печатью

Для печати документов с маркировкой используется графическая утилита `fly-print-station`. Утилита предназначена для управления заданиями на печать, для маркировки документов, отправленных на печать, а также для просмотра журнала маркировки.

Описание использования утилиты приведено в электронной справке.

14.8. Маркировка нескольких экземпляров документа

Для печати нескольких экземпляров документа с ненулевой меткой безопасности пользователь должен отправить на печать только одну копию документа.

Пользователь, осуществляющий маркировку, должен выполнить следующую последовательность действий:

1) получить номер задания для маркировки, выполнив команду:

```
lpq -a
```

2) задать число копий для печати, выполнив команду:

```
lpattr -j <номер_задания> -s copies=<число_копий>
```

3) произвести маркировку с помощью инструмента `markjob` или графической утилиты `fly-print-station`.

После выполнения маркировки в очереди формируются по два дополнительных задания для каждого экземпляра документа, располагаемых в очереди последовательно. Первое (с меньшим номером) представляет собой промаркированный экземпляр документа, а второе (с большим номером) — маркировку, размещаемую на обороте последнего листа экземпляра документа. Для печати экземпляра документа необходимо возобновить выполнение первого соответствующего ему задания, что приведет к печати промаркированного экземпляра документа. Затем на обороте последнего листа экземпляра документа печатается маркировка посредством возобновления выполнения второго соответствующего экземпляру документа дополнительного задания.

14.9. Журнал маркировки

Журнал маркировки ведется при установке в конфигурационном файле `/etc/cups/cupsd.conf` для параметра `MacJournal` значения `on`. По умолчанию журнал записывается в базу данных SQLITE `/var/spool/cups/parsec/marking-journal.sqlite`.

Включить журнал маркировки возможно путем редактирования конфигурационного файла или выполнив команду от имени администратора:

```
cupsctl MacJournal=On
```

Просмотр журнала возможен с использованием графической утилиты `fly-print-station` и графической утилиты `fly-admin-printer` с установленным плагином `fly-admin-printer-mac`.

15. ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

В качестве защищенной СУБД в составе ОС используется СУБД Tantor (в исполнении Basic), доработанной в соответствии с требованием интеграции с ОС в части защиты информации, в том числе мандатного управления доступом.

СУБД предназначена для создания и управления реляционными БД и предоставляет многопользовательский доступ к расположенным в них данным. Данные в реляционной БД хранятся в отношениях (таблицах), состоящих из строк и столбцов. При этом единицей хранения и доступа к данным является строка, состоящая из полей. Каждое поле строки идентифицируется именами столбцов. Кроме таблиц существуют другие объекты БД (виды, процедуры и т. п.), которые предоставляют доступ к данным, хранящимся в таблицах.

Подробное описание настройки и управления защищенной СУБД приведено в документе РУСБ.10015-01 97 01-3. Описание работы пользователя с СУБД приведено в документе РУСБ.10015-01 93 01 «Операционная система специального назначения «Astra Linux Special Edition». Руководство пользователя».

16. ЗАЩИЩЕННЫЙ КОМПЛЕКС ПРОГРАММ ЭЛЕКТРОННОЙ ПОЧТЫ

В качестве защищенного комплекса программ электронной почты используется сервер электронной почты, состоящий из агента передачи электронной почты (Mail Transfer Agent, MTA) Exim4, агента доставки электронной почты (Mail Delivery Agent, MDA) Dovecot и клиента электронной почты (Mail User Agent, MUA) Mozilla Thunderbird, доработанных для реализации следующих дополнительных функциональных возможностей:

- интеграции с ядром ОС и базовыми библиотеками для обеспечения разграничения доступа;
- реализации мандатного управления доступом к почтовым сообщениям;
- автоматической маркировки создаваемых почтовых сообщений, отражающих уровень их конфиденциальности;
- регистрации попыток доступа к почтовым сообщениям.

Агент передачи электронной почты использует протокол SMTP и обеспечивает решение следующих задач:

- доставку исходящей почты от авторизованных клиентов до сервера, который является целевым для обработки почтового домена получателя;
- прием и обработку почтовых сообщений доменов, для которых он является целевым;
- передачу входящих почтовых сообщений для обработки агентом доставки электронной почты.

Агент доставки электронной почты предназначен для обслуживания почтового каталога и предоставления удаленного доступа к почтовому ящику по протоколу IMAP.

Клиент электронной почты — это прикладное ПО, устанавливаемое на рабочем месте пользователя и предназначенное для получения, создания, отправки и хранения сообщений электронной почты пользователя.

16.1. Состав

Защищенный комплекс программ электронной почты состоит из следующих пакетов:

- `exim4-daemon-heavy` — агент передачи сообщений Exim4. Пакет `exim4-daemon-light` не поддерживает работу с классификационными метками, отличными от `0:0`;
- `dovecot-imapd` — агент доставки сообщений Dovecot. Работает только по протоколу IMAP, протокол POP3 отключен. Серверная часть защищенного комплекса программ электронной почты использует в качестве почтового хранилища MailDir

(mailbox не поддерживает работу с классификационными метками, отличными от `0:0`);

- `thunderbird` — клиент электронной почты Mozilla Thunderbird.

16.2. Настройка серверной части

Настройки по умолчанию:

1) прием почтовых сообщений по протоколу SMTP только от MUA из доменов `relay-domens` и из подсети;

2) отправка почтовых сообщений по протоколу SMTP в соответствии с DNS;

3) хранение локальной почты в MailDir в `/var/mail/%u`, где `%u` — локальная часть адресата;

4) выдача локальных почтовых сообщений по протоколу IMAP.

ВНИМАНИЕ! Для обеспечения нормальной работы пользователя с сетевыми службами должна быть явно задана его классификационная метка (диапазон уровней конфиденциальности и категорий конфиденциальности) с помощью соответствующих утилит, даже если ему не доступны уровни и категории выше 0. Дополнительная информация приведена в документе РУСБ.10015-01 97 01-1.

ВНИМАНИЕ! Редактирование конфигурационных файлов и выполнение команд по настройке необходимо выполнять от имени учетной записи администратора с использованием механизма `sudo`.

16.2.1. Настройка агента доставки сообщений

Настройка агента доставки сообщений Dovecot осуществляется путем правки конфигурационного файла `/etc/dovecot/dovecot.conf` и конфигурационных файлов в каталоге `/etc/dovecot/conf.d`.

В файле `/etc/dovecot/dovecot.conf` необходимо задать список интерфейсов, с которых будут приниматься соединения, и установить протокол IMAP, например:

```
protocols = imap
listen =  192.168.2.55
```

Для настройки аутентификации с использованием PAM в конфигурационном файле `/etc/dovecot/conf.d/10-auth.conf` необходимо установить:

```
disable_plaintext_auth = no
auth_mechanisms = plain
```

Агент доставки сообщений Dovecot для PAM-аутентификации использует сценарий PAM, содержащийся в конфигурационном файле `/etc/pam.d/dovecot`. PAM-сценарий для Dovecot включает `common-auth` и `common-account`. По умолчанию в ОС для фиксации числа неверных попыток входа пользователей применяется PAM-модуль `pam_tally`. Использование `pam_tally` в секции `auth` в файле `/etc/pam.d/common-auth` обеспечивает увеличение счетчика неверных попыток входа пользователя при начале процесса аутентификации. Для сброса счетчика неверных попыток входа пользователя после успешной аутентификации в Dovecot необходимо в сценарий PAM, содержащийся в конфигурационном файле `/etc/pam.d/dovecot`, добавить использование `pam_tally` в секции `account`. PAM-сценарий для Dovecot будет иметь следующий вид:

```
@include common-auth
@include common-account
@include common-session
account required pam_tally.so
```

В случае когда SSL не будет использоваться в конфигурационном файле `/etc/dovecot/conf.d/10-ssl.conf`, необходимо установить:

```
ssl = no
```

Для настройки встроенного в MDA Dovecot сервера SASL, к которому будет обращаться MTA Exim4 для аутентификации пользователей, в конфигурационном файле `/etc/dovecot/conf.d/10-master.conf` в секцию `service auth` необходимо добавить:

```
unix_listener auth-client {
mode = 0600
user = Debian-exim
}
```

Перезапустить MDA Dovecot, выполнив команду:

```
systemctl restart dovecot
```

16.2.2. Настройка агента передачи сообщений

Для настройки агента передачи сообщений Exim4 требуется инициировать переконфигурирование пакета `exim4-config`, для этого выполнить в эмуляторе терминала команду:

```
sudo dpkg-reconfigure exim4-config
```

В появившемся окне настройки для указанных ниже параметров необходимо установить следующие значения:

1) «Общий тип почтовой конфигурации» — выбрать пункт «интернет-сайт; прием и отправка почты напрямую, используя SMTP»;

2) «Почтовое имя системы» — ввести имя домена;

3) «IP-адреса, с которых следует ожидать входящие соединения» — ввести IP-адрес сервера;

4) «Другие места назначения, для которых должна приниматься почта» — ввести имя домена;

5) «Домены, для которых доступна релейная передача почты» — оставить пустым;

6) «Машины, для которых доступна релейная передача почты» — оставить пустым;

7) «Сокращать количество DNS-запросов до минимума» — выбрать пункт «Нет»;

8) «Метод доставки локальной почты» — выбрать пункт «Maildir формат в /var/mail/»;

9) «Разделить конфигурацию на маленькие файлы» — выбрать пункт «Да».

При необходимости изменить расположение каталога хранилища локальной почты `/var/mail` следует убедиться, что на новый каталог установлены права `1777`, максимально доступный в системе уровень конфиденциальности, полный набор категорий конфиденциальности и атрибут `ccnr`. Если это не так, то установить права и классификационную метку командами:

```
sudo chmod 1777 new_dir
sudo pdpl-file <макс_уровень>::0xffffffffffffffff:ccnr new_dir
```

Для штатной работы `exim4-daemon-heavy` необходимо удалить файлы, созданные в каталоге `/var/mail` при установке пакета.

В каталоге `/etc/exim4/conf.d/auth` необходимо создать файл с именем `05_dovecot_login` и следующим содержимым:

```
dovecot_plain:
        driver = dovecot
        public_name = plain
        server_socket = /var/run/dovecot/auth-client
        server_set_id = $auth1
```

Для запрета отправки писем без аутентификации необходимо в конфигурационном файле `/etc/exim4/conf.d/acl/30_exim4-config_check_rcpt` в начало секции `acl_check_rcpt` добавить строки:

```
deny
    message = "Auth required"
    hosts = *:+relay_from_hosts
    !authenticated = *
```

Настройку сквозной аутентификации для сервера и клиента, работающих в рамках ЕПП, см. в 16.4.

Настроить автоматический запуск службы MTA Exim4, выполнив команду:

```
sudo systemctl enable exim4
```

Перезапустить MTA Exim4, выполнив команду:

```
systemctl restart exim4
```

16.3. Настройка клиентской части

Первичное создание для пользователя учетной записи сервера электронной почты в MUA Mozilla Thunderbird должно производиться с нулевой классификационной меткой (значение уровня конфиденциальности `0`, категорий конфиденциальности нет). Далее для каждой конкретной классификационной метки (значение уровня и набор категорий) создание учетной записи необходимо повторить.

При создании учетной записи пользователя в MUA Mozilla Thunderbird необходимо выбрать тип используемого сервера входящей почты IMAP.

При настройке учетной записи установить в параметрах сервера и параметрах сервера исходящей почты:

- «Защита соединения» — из выпадающего списка выбрать «Нет»;
- «Метода аутентификации» — выбрать «Обычный пароль».

16.4. Настройка для работы со службой FreeIPA

Для обеспечения совместной работы сервера электронной почты с FreeIPA должны быть установлены:

- агент передачи сообщений Exim4 — из пакета `exim4-daemon-heavy`;

- агент доставки сообщений Dovecot — из пакета `dovecot-imapd`;
- пакет `dovecot-gssapi` поддержки GSSAPI-аутентификации для MDA Dovecot;
- клиент Mozilla Thunderbird — из пакета `thunderbird`.

Для настройки совместной работы сервера электронной почты с FreeIPA должно быть предварительно выполнено:

- установлен сервер контроллера домена FreeIPA (например, домен `astra.mta`);
- на отдельном компьютере установлен почтовый сервер, введенный в домен FreeIPA (например, сервер `exim1.astra.mta` с IP-адресом `192.168.32.3`).

16.4.1. Настройка почтового сервера

Установить на почтовом сервере необходимые пакеты следующей командой:

```
sudo apt install exim4-daemon-heavy dovecot-imapd dovecot-gssapi
```

При установке пакетов `dovecot-imapd` и `dovecot-gssapi` создается файл `/etc/dovecot/conf.d/10-master.conf`. В секции `service auth` этого файла необходимо добавить следующие строки:

```
unix_listener auth-client {
mode = 0600
user = Debian-exim
}
```

После внесения изменений следует выполнить команду для реконфигурации Exim:

```
sudo dpkg-reconfigure exim4-config
```

В появившемся окне настройки для указанных ниже параметров необходимо установить следующие значения:

1) «Общий тип почтовой конфигурации» — выбрать пункт «доставка только локальной почты; доступа к сети нет»;

2) «Почтовое имя системы» — ввести имя домена, например «astra.mta»;

3) «IP-адреса, с которых следует ожидать входящие соединения» — указать IP-адрес сервера или оставить поле пустым;

4) «Другие места назначения, для которых должна приниматься почта» — ввести имя домена, например «astra.mta»;

5) «Машины, для которых доступна релейная передача почты» — указать IP-адреса, например «192.168.32.0/24»;
6) «Сокращать количество DNS-запросов до минимума» — выбрать пункт «Нет»;
7) «Метод доставки локальной почты» — выбрать пункт «Maildir формат в /var/mail/»;
8) «Разделить конфигурацию на маленькие файлы» — выбрать пункт «Да».

В журнале Exim (файл `/var/log/exim4/paniclog`) могут появляться сообщения об ошибках вида:

```
Failed to create spool file /var/spool/exim4//input//1jb2ok-00031u-5R-D:
Operation not permitted
```

В этом случае следует исправить права доступа к каталогу `/var/spool/exim4`:

```
sudo chown -R Debian-exim:Debian-exim /var/spool/exim4/
```

16.4.2. Регистрация почтовых служб на контроллере домена

На контроллере домена необходимо добавить принципалов служб:

- `imap/exim1.astra.mta@ASTRA.MTA`
- `smtp/exim1.astra.mta@ASTRA.MTA`

Это можно сделать через веб-интерфейс FreeIPA, перейдя «Идентификация — Службы» и нажав кнопку **[Добавить]** (см. рис. 9).

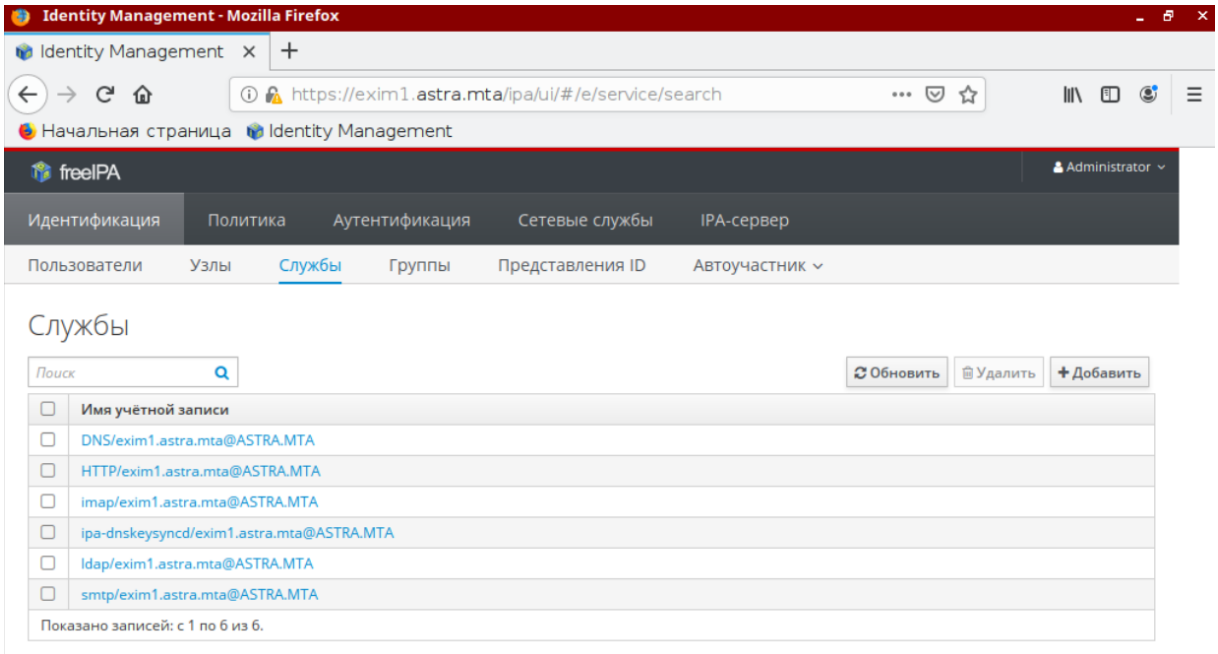

Рис. 9

Также данное действие возможно выполнить из командной строки, предварительно получив полномочия администратора домена:

```
sudo kinit admin
sudo ipa service-add imap/exim1.astra.mta@ASTRA.MTA
sudo ipa service-add smtp/exim1.astra.mta@ASTRA.MTA
```

16.4.3. Получение таблицы ключей на почтовом сервере

На почтовом сервере следует получить таблицу ключей для службы `imap`, затем добавить таблицу ключей для службы `smtp`:

```
sudo kinit admin
sudo ipa-getkeytab --principal=imap/exim1.astra.mta@ASTRA.MTA \
     --keytab=/var/lib/dovecot/dovecot.keytab
sudo ipa-getkeytab --principal=smtp/exim1.astra.mta@ASTRA.MTA \
     --keytab=/var/lib/dovecot/dovecot.keytab
```

Проверить полученную таблицу ключей:

```
sudo klist -k /var/lib/dovecot/dovecot.keytab
```

Вывод в терминале будет иметь следующий вид:

```
Keytab name: FILE:/var/lib/dovecot/dovecot.keytab
KVNO Principal
----------------------------------------------------------------------------
1 imap/exim1.astra.mta@ASTRA.MTA
1 imap/exim1.astra.mta@ASTRA.MTA
1 smtp/exim1.astra.mta@ASTRA.MTA
1 smtp/exim1.astra.mta@ASTRA.MTA
```

После этого следует выдать пользователю `dovecot` права на чтение файла ключа Kerberos:

```
sudo setfacl -m u:dovecot:x /var/lib/dovecot
sudo setfacl -m u:dovecot:r /var/lib/dovecot/dovecot.keytab
```

Далее убедиться, что в конфигурационном файле `/etc/dovecot/dovecot.conf` отключено использование протоколов POP3, и отключить неиспользуемые протоколы, оставив только IMAP:

```
protocols = imap
```

После этого следует выполнить настройки в конфигурационном файле `/etc/dovecot/conf.d/10-auth.conf`:

- для отключения передачи при аутентификации пароля открытым текстом установить:

```
disable_plaintext_auth = yes
```

- для настройки аутентификации посредством Kerberos с использованием метода GSSAPI установить:

```
auth_gssapi_hostname = exim1.astra.mta
auth_krb5_keytab = /var/lib/dovecot/dovecot.keytab
auth_mechanisms = gssapi
```

Затем перезапустить Dovecot:

```
sudo systemctl restart dovecot
```

16.4.4. Настройка аутентификации через Kerberos

Для настройки аутентификации в Exim следует создать конфигурационный файл `/etc/exim4/conf.d/auth/33_exim4-dovecot-kerberos-ipa` со следующим содержимым:

```
dovecot_gssapi:
driver = dovecot
public_name = GSSAPI
server_socket = /var/run/dovecot/auth-client
server_set_id = $auth1
```

Далее запустить сервер Exim и разрешить его автоматический запуск после перезагрузки:

```
sudo systemctl start exim4
sudo systemctl enable exim4
```

После настройки аутентификации через Kerberos в домене FreeIPA требуется настройка параметров почтового сервера (параметров пересылки почты) и настройка клиентской части на клиентах.

17. СРЕДСТВА АУДИТА И ЦЕНТРАЛИЗОВАННОГО ПРОТОКОЛИРОВАНИЯ

17.1. Аудит

В ОС отправка и регистрация информации о событиях в системе осуществляется в соответствии со стандартом Syslog. Стандарт определяет формат сообщений о событиях и правила их передачи и регистрации в журналах. Основное расположение файлов журналов — системный каталог `/var/log`.

Аудит основных системных событий с момента запуска ОС ведется в системном журнале `/var/log/syslog`.

Аудит событий постановки/снятия с контроля целостности исполняемых модулей и файлов данных, а также событий неудачного запуска неподписанных файлов осуществляется в журнале ядра `/var/log/kern.log`.

Аудит событий создания/удаления/изменения настроек учетных записей пользователей и начала/окончания сеансов работы учетных записей пользователей осуществляется в журнале `/var/log/auth.log`.

Аудит событий изменения для учетных записей полномочий по доступу к информации осуществляется в журнале `/var/log/auth.log`.

Аудит событий смены аутентифицирующей информации учетных записей осуществляется в журнале `/var/log/auth.log`.

Аудит событий вывода текстовых (графических) документов на бумажный носитель осуществляется в журнале `/var/log/cups/page_log`.

Для аудита ОС также могут использоваться журналы различных служб и программ.

Для регистрации событий безопасности в ОС используется служба аудита `auditd`, описание которой приведено в РУСБ.10015-01 97 01-1, и подсистема регистрации событий (см. 17.2).

17.2. Подсистема регистрации событий

В ОС реализована подсистема регистрации событий, которая собирает информацию о событиях, выполняет ее регистрацию и предоставляет инструменты для просмотра собранных данных и реагирования на события. Регистрация событий безопасности выполняется с учетом требований ГОСТ Р 59548-2022.

Сбор и регистрацию событий осуществляет служба `syslog-ng`. Служба `syslog-ng` принимает информацию о событиях из различных источников (события от `auditd`, собственные подключаемые модули, файлы, прикладное ПО и др.), выполняет фильтрацию и обработ-

ку полученных данных, регистрирует события в журнал `/parsec/log/astra/events`, а также, в зависимости от конфигурации, может сохранять в файл, отправлять по сети и т. д.

Для управления подсистемой регистрации событий используются графические утилиты `fly-admin-events` («Настройка регистрации системных событий») и `fly-event-viewer` («Журнал системных событий»). Описание использования утилит приведено в электронной справке.

Также для управления регистрацией событий могут использоваться инструменты командной строки `astra-admin-events` и `astra-event-viewer`. Порядок использования инструментов приведен на соответствующих страницах помощи:

```
astra-admin-events -h
astra-event-viewer -h
```

Информирование (оповещение) о событиях осуществляется с помощью утилиты `fly-notifications` («Центр уведомлений»). Описание утилиты приведено в электронной справке.

Подробное описание подсистемы регистрации событий приведено в РУСБ.10015-01 97 01-1.

17.3. Средства централизованного протоколирования

Для решения задач централизованного протоколирования и анализа журналов аудита, а также организации распределенного мониторинга сети, жизнеспособности и целостности серверов используется программное решение Zabbix, реализованное на веб-сервере Apache, СУБД (MySQL, Oracle, PostgreSQL, SQLite) и языке сценариев PHP.

Zabbix предоставляет гибкий механизм сбора данных. Все отчеты и статистика Zabbix, а также параметры настройки компонентов Zabbix доступны через веб-интерфейс. В веб-интерфейсе реализован следующий функционал:

- вывод отчетности и визуализация собранных данных;
- создание правил и шаблонов мониторинга состояния сети и узлов;
- определение допустимых границ значений заданных параметров;
- настройка оповещений;
- настройка автоматического реагирования на события безопасности.

17.3.1. Архитектура

Zabbix состоит из следующих основных программных компонентов:

1) сервер — является основным компонентом, который выполняет мониторинг, взаимодействует с прокси и агентами, вычисляет триггеры, отправляет оповещения. Является главным хранилищем данных конфигурации, статистики, а также оперативных данных;

2) агенты — разворачиваются на наблюдаемых системах для активного мониторинга за локальными ресурсами и приложениями и для отправки собранных данных серверу или прокси;

3) прокси — может собирать данные о производительности и доступности от имени сервера. Прокси является опциональной частью Zabbix и может использоваться для снижения нагрузки на сервер;

4) база данных — вся информация о конфигурации, а также собранные Zabbix данные, хранятся в базе данных;

5) веб-интерфейс — используется для доступа к Zabbix из любого места и с любой платформы.

17.3.2. Сервер

Для установки сервера с СУБД выполнить команду:

```
apt install zabbix-server-pgsql zabbix-frontend-php
```

Для создания базы данных сервера используются сценарии по созданию базы данных для СУБД, например:

```
psql -U <username>
create database zabbix;
\q
cd database/postgresql
psql -U <username> zabbix < schema.sql
psql -U <username> zabbix < images.sql
psql -U <username> zabbix < data.sql
```

Далее необходимо импортировать исходную схему и данные сервера на:

```
zcat /usr/share/doc/zabbix-server-pgsql/create.sql.gz | psql -U <username>
     zabbix
```

Для настройки базы данных сервера откорректировать конфигурационный файл `zabbix_server.conf`.

Пример

```
vi /etc/zabbix/zabbix_server.conf
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=<пароль>
```

В параметре `DBPassword` указывается пароль пользователя СУБД.

Основные параметры конфигурационного файла сервера приведены в таблице 51.

Таблица 51

Параметр	Описание
`AllowRoot`	Разрешение серверу запускаться от имени пользователя `root`. Если не разрешено (значение «0») и сервер запускается от имени `root`, сервер попробует переключиться на пользователя `zabbix`. Не влияет, если сервер запускается от имени обычного пользователя. Значение по умолчанию — 0
`CacheSize`	Размер кэша конфигурации в байтах для хранения данных узлов сети, элементов данных и триггеров. Возможные значения от 128 КБ до 8 ГБ, значение по умолчанию — 8 МБ
`CacheUpdateFrequency`	Частота выполнения процедуры обновления кэша конфигурации, в секундах. Возможные значения от 1 до 3600 сек, значение по умолчанию — 60 сек
`DBHost`	Имя хоста базы данных. В случае пустой строки СУБД будет использовать сокет. Значение по умолчанию — `localhost`
`DBName`	Обязательный параметр. Имя базы данных
`DBPassword`	Пароль к базе данных
`DBPort`	Порт базы данных, когда не используется `localhost`. Значение по умолчанию — 3306
`DBSchema`	Имя схемы
`DBUser`	Пользователь базы данных
`HousekeepingFrequency`	Частота выполнения автоматической процедуры очистки базы данных от устаревшей информации, в часах. Возможные значения от 0 до 24 ч, значение по умолчанию — 1

Сервер работает как служба (демон). Для запуска сервера выполнить команду:

```
systemctl start zabbix-server
```

Соответственно для остановки, перезапуска и просмотра состояния сервера используются следующие команды:

```
systemctl stop zabbix-server
systemctl restart zabbix-server
systemctl status zabbix-server
```

ВНИМАНИЕ! Для работы сервера необходима кодировка UTF-8 иначе некоторые текстовые элементы данных могут быть интерпретированы некорректно.

В таблице 52 приведены основные параметры, используемые при управлении сервером.

Таблица 52

Параметр	Описание
`-c --config <файл>`	Путь к файлу конфигурации. Значение по умолчанию `/usr/local/etc/zabbix_server.conf`
`-R --runtime-control <параметр>`	Выполнение административных функций
`config_cache_reload`	Перезагрузка кэша конфигурации. Игнорируется, если кэш загружается в данный момент: `zabbix_server -c /usr/local/etc/zabbix_server.conf -R config_cache_reload`
`housekeeper_execute`	Запуск процедуры очистки базы данных. Игнорируется, если процедура очистки выполняется в данный момент `zabbix_server -c /usr/local/etc/zabbix_server.conf -R housekeeper_execute`
`log_level_increase[=<цель>]`	Увеличение уровня регистрации событий. Если цель не указана, затрагивает все процессы. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номер процесса: `zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase` `zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234` `zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2`
`log_level_decrease[=<цель>]`	Уменьшение уровня регистрации событий. Если цель не указана, затрагивает все процессы. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номер процесса: `zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"`

17.3.3. Агенты

Агенты могут выполнять пассивные и активные проверки.

При пассивной проверке агент отвечает на запрос от сервера или прокси.

При активной проверке агент получает от сервера перечень данных для мониторинга, затем осуществляет сбор данных согласно полученному перечню и периодически отправляет собранные данные серверу.

Выбор между пассивной и активной проверкой осуществляется выбором соответствующего типа элемента данных. Агент обрабатывает элементы данных типов «Zabbix агент» и «Zabbix агент (активный)».

Для установки агента в UNIX-системах выполнить команду:

```
apt install zabbix-agent
```

Основные параметры конфигурационного файла агента UNIX приведены в таблице 53.

Таблица 53

Параметр	Описание
AllowRoot	Разрешение серверу запускаться от имени пользователя root. Если не разрешено (значение «0») и сервер запускается от имени root, сервер попробует переключиться на пользователя zabbix. Не влияет, если сервер запускается от имени обычного пользователя. Значение по умолчанию — 0
EnableRemoteCommands	Указывает разрешены ли удаленные команды с сервера: - 0 — не разрешены; - 1 — разрешены
Hostname	Уникальное, регистрозависимое имя машины. Требуется для активных проверок и должно совпадать с именем машины, указанным на сервере
ListenIP	Список IP-адресов, разделенных запятой, которые агент должен слушать
ListenPort	Порт, который необходимо слушать для подключений с сервера
LogFile	Имя файла журнала. Обязательный параметр, если тип журнала указан file (см. параметр LogType)
LogType	Тип вывода журнала: - file — запись журнала в файл, указанный в параметре LogFile; - system— запись журнала в syslog; - console — вывод журнала в стандартный вывод

Окончание таблицы 53

Параметр	Описание
`Server`	Список IP-адресов или имен серверов, разделенных запятой. Входящие соединения будут приниматься только с адресов, указанных в параметре
`TLSAccept`	Опциональный параметр. Является обязательным в случае, если заданы TLS-сертификат или параметры PSK. Указывает какие входящие подключения принимаются. Используется пассивными проверками. Можно указывать несколько значений, разделенных запятой: - `unencrypted` — принимать подключения, не использующие ключи (по умолчанию); - `psk` — принимать подключения с TLS и pre-shared ключом (PSK); - `cert` — принимать подключения с TLS и сертификатом
`TLSConnect`	Опциональный параметр. Является обязательным в случае, если заданы TLS-сертификат или параметры PSK. Как агент должен соединяться с сервером или прокси. Используется активными проверками. Можно указать только одно значение: - `unencrypted` — подключаться без использования ключей (по умолчанию); - `psk` — подключаться, используя TLS и pre-shared ключ (PSK); - `cert` — подключаться, используя TLS и сертификат
`User`	Использование привилегий указанного (существующего) пользователя системы. Значение по умолчанию — `zabbix`. Актуально только если запускается от имни пользователя `root` и параметр `AllowRoot` не разрешен

Агент UNIX работает как служба, для запуска выполнить команду:

```
systemctl start zabbix-agent
```

Соответственно для остановки, перезапуска и просмотра состояния агента UNIX используются следующие команды:

```
systemctl stop zabbix-agent
systemctl restart zabbix-agent
systemctl status zabbix-agent
```

В среде Windows агент работает как служба. Агент Windows распространяется в виде `zip`-архива. Агент `bin\win64\zabbix_agentd.exe` и файл конфигурации `conf\zabbix_agentd.win.conf` из `zip`-архива необходимо скопировать в один каталог, например, `C:\zabbix`.

При необходимости откорректировать конфигурационный файл `c:\zabbix\zabbix_agentd.win.conf`.

Основные параметры конфигурационного файла агента Windows приведены в таблице 54.

Таблица 54

Параметр	Описание
`EnableRemoteCommands`	Указывает разрешены ли удаленные команды с сервера: - 0 — не разрешены; - 1 — разрешены
`Hostname`	Уникальное, регистрозависимое имя машины. Требуется для активных проверок и должно совпадать с именем машины, указанным на сервере
`ListenIP`	Список IP-адресов, разделенных запятой, которые агент должен слушать
`ListenPort`	Порт, который необходимо слушать для подключений с сервера
`LogFile`	Имя файла журнала. Обязательный параметр, если тип журнала указан `file` (см. параметр `LogType`)
`LogType`	Тип вывода журнала: - `file` — запись журнала в файл, указанный в параметре `LogFile`; - `system`— запись журнала в Журнал событий Windows; - `console` — вывод журнала в стандартный вывод
`Server`	Список IP-адресов или имен серверов, разделенных запятой. Входящие соединения будут приниматься только с адресов, указанных в параметре
`TLSAccept`	Опциональный параметр. Является обязательным в случае, если заданы TLS-сертификат или параметры PSK. Указывает какие входящие подключения принимаются. Используется пассивными проверками. Можно указывать несколько значений, разделенных запятой: - `unencrypted` — принимать подключения, не использующие ключи (по умолчанию); - `psk` — принимать подключения с TLS и pre-shared ключом (PSK); - `cert` — принимать подключения с TLS и сертификатом
`TLSConnect`	Опциональный параметр. Является обязательным в случае, если заданы TLS-сертификат или параметры PSK. Как агент должен соединяться с сервером или прокси. Используется активными проверками. Можно указать только одно значение: - `unencrypted` — подключаться без использования ключей (по умолчанию); - `psk` — подключаться, используя TLS и pre-shared ключом (PSK); - `cert` — подключаться, используя TLS и сертификат

Окончание таблицы 54

Параметр	Описание
User	Использование привилегий указанного (существующего) пользователя системы. Значение по умолчанию — zabbix. Актуально только если запускается от имени пользователя root и параметр AllowRoot не разрешен

Для установки агента Windows как службы используется следующая команда:

```
C:\> c:\zabbix\zabbix_agentd.exe -c c:\zabbix\zabbix_agentd.win.conf -i
```

В таблице 55 приведены основные параметры, используемые при управлении агентом.

Таблица 55

Параметр	Описание
Агент UNIX и Windows	
-c --config <файл_конфигурации>	Путь к файлу конфигурации, размещенному в каталоге, отличном от заданного по умолчанию. В UNIX путь по умолчанию /usr/local/etc/zabbix_agentd.conf. В Windows — c:\zabbix_agentd.conf
-p --print	Вывод известных данных
-t --test <ключ_элемента_данных>	Тестирование указанного элемента данных
Агент UNIX	
-R --runtime-control <параметр>	Выполнение административных функций согласно назначенному уровню регистрации событий у процессов агента
log_level_increase[=<цель>]	Увеличение уровня регистрации событий. Если цель не указана, затрагивает все процессы. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номер процесса: zabbix_agentd -R log_level_increase zabbix_agentd -R log_level_increase=1234 zabbix_agentd -R log_level_increase=listener,2
log_level_decrease[=<цель>]	Уменьшение уровня регистрации событий. Если цель не указана, затрагивает все процессы. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номер процесса: zabbix_agentd -R log_level_decrease="active checks"
Агент Windows	
-m --multiple-agents	Использование нескольких экземпляров агента (с -i,-d,-s,-x функциями). Для отделения имени экземпляров служб каждое имя службы будет в значении Hostvalue из указанного файла конфигурации
-i --install	Установка агента как службы
-d --uninstall	Удаление службы агента

Окончание таблицы 55

Параметр	Описание
`-s --start`	Запуск службы агента
`-x --stop`	Остановка службы агента

17.3.4. Прокси

Для прокси требуется отдельная база данных. Для установки СУБД для прокси выполнить команду:

```
apt install zabbix-proxy-pgsql
```

Для создания базы данных прокси используются сценарии по созданию базы данных:

```
psql -U <username>
create database zabbix;
\q
cd database/postgresql
psql -U <username> zabbix < schema.sql
```

Далее необходимо импортировать исходную схему и данные прокси:

```
zcat /usr/share/doc/zabbix-proxy-pgsql/create.sql.gz | psql -U <username>
   zabbix
```

Для настройки базы данных прокси изменить конфигурационный файл `zabbix_proxy.conf`.

Пример

```
vi /etc/zabbix/zabbix_proxy.conf
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=<пароль>
```

В параметре `DBPassword` указать пароль пользователя СУБД.

Основные параметры конфигурационного файла прокси приведены в таблице 56.

Таблица 56

Параметр	Описание
AllowRoot	Разрешение прокси запускаться от имени пользователя root. Если не разрешено (значение «0») и прокси запускается от имени root, прокси попробует переключиться на пользователя zabbix. Не влияет, если прокси запускается от имени обычного пользователя. Значение по умолчанию — 0
CacheSize	Размер кэша конфигурации в байтах для хранения данных узлов сети, элементов данных и триггеров. Возможные значения от 128 КБ до 8 ГБ, значение по умолчанию — 8 МБ
ConfigFrequency	Частота получения данных конфигурации от сервера, в секундах. Параметр активного прокси. Игнорируется пассивными прокси (см. параметр ProxyMode). Возможные значения от 1 до 604800 сек, значение по умолчанию — 3600 сек
DBHost	Имя хоста базы данных. В случае пустой строки СУБД будет использовать сокет. Значение по умолчанию — localhost
DBName	Обязательный параметр. Имя базы данных, должно отличаться от имени базы данных сервера
DBPassword	Пароль к базе данных
DBPort	Порт базы данных, когда не используется localhost. Значение по умолчанию — 3306
DBSchema	Имя схемы
DBUser	Пользователь базы данных
DataSenderFrequency	Частота отправки собранных значений серверу, в секундах. Параметр активного прокси. Игнорируется пассивными прокси (см. параметр ProxyMode). Возможные значения от 1 до 3600 сек, значение по умолчанию — 1 сек
Hostname	Уникальное регистрозависимое имя прокси
HousekeepingFrequency	Частота выполнения автоматической процедуры очистки базы данных от устаревшей информации, в часах. Возможные значения от 0 до 24 ч, значение по умолчанию — 1
ProxyMode	Режим работы прокси: - 0 — прокси в активном режиме; - 1 — прокси в пассивном режиме
Server	IP-адрес или имя сервера для доступа к данным конфигурации с сервера. Параметр активного прокси, игнорируется пассивными прокси (см. ProxyMode)

Окончание таблицы 56

Параметр	Описание
TLSAccept	Опциональный параметр. Является обязательным в случае, если заданы TLS-сертификат или параметры PSK. Указывает какие входящие подключения принимаются от сервера. Используется пассивным прокси, игнорируется активным прокси. Можно указывать несколько значений, разделенных запятой: - unencrypted — принимать подключения, не использующие ключи (по умолчанию); - psk — принимать подключения с TLS и pre-shared ключом (PSK); - cert — принимать подключения с TLS и сертификатом
TLSConnect	Опциональный параметр. Является обязательным в случае, если заданы TLS-сертификат или параметры PSK. Как прокси должен соединяться с сервером. Используется активным прокси, игнорируется пассивным прокси. Можно указать только одно значение: - unencrypted — подключаться без использования ключей (по умолчанию); - psk — подключаться, используя TLS и pre-shared ключом (PSK); - cert — подключаться, используя TLS и сертификат

Прокси работает как служба. Для запуска прокси выполнить команду:

```
systemctl start zabbix-proxy
```

Соответственно для остановки, перезапуска и просмотра состояния прокси используются следующие команды:

```
systemctl stop zabbix-proxy
systemctl restart zabbix-proxy
systemctl status zabbix-proxy
```

В таблице 57 приведены основные параметры командной строки zabbix-proxy.

Таблица 57

Параметр	Описание
-c --config <файл>	Путь к файлу конфигурации. Значение по умолчанию /etc/zabbix/zabbix_proxy.conf
-R --runtime-control <параметр>	Выполнение административных функций

Окончание таблицы 57

Параметр	Описание
`config_cache_reload`	Перезагрузка кэша конфигурации. Игнорируется, если кэш загружается в данный момент. Активный прокси подключится к серверу и запросит данные конфигурации: `zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R config_cache_reload`
`housekeeper_execute`	Запуск процедуры очистки базы данных. Игнорируется, если процедура очистки выполняется в данный момент: `zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R housekeeper_execute`

`log_level_increase[=<цель>]`	Увеличение уровня регистрации событий. Если цель не указана, затрагивает все процессы. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номера процесса: `zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase`
	`zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234`
	`zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2`

`log_level_decrease[=<цель>]`	Уменьшение уровня регистрации событий. Если цель не указана, затрагивает все процессы. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номера процесса: `zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"`

17.3.5. Веб-интерфейс

Настройка и управление работой Zabbix осуществляется посредством веб-интерфейса.

Установка веб-интерфейса производится путем копирования `php`-файлов в папку HTML веб-сервера. Далее необходимо:

1) ввести URL Zabbix `http://<ip_или_имя_сервера>/zabbix` в браузере — откроется первая страница помощника установки веб-интерфейса;

2) указать данные для подключения к базе данных. База данных должна быть создана;

3) указать данные сервера;

4) подтвердить данные для настройки;

5) скачать конфигурационный файл и поместить его в каталог `conf/` (если веб-сервер имеет право на запись в каталог `conf/`, файл будет сохранен автоматически);

6) завершить установку.

Для входа по умолчанию используется имя пользователя `Admin` и пароль `zabbix`.

18. РЕЗЕРВНОЕ КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ ДАННЫХ

Система резервного копирования является составной частью плана восстановления системы.

Резервное копирование выполняется с целью обеспечения возможности восстановления отдельных файлов или ФС в целом с минимальными затратами труда и времени в случае утери рабочей копии информации. Резервные копии должны создаваться периодически, в соответствии с заранее установленным графиком (см. 18.2).

Процесс резервного копирования должен быть максимально автоматизирован и требовать наименьшего участия со стороны администратора системы.

Резервное копирование — это процесс, влияющий на работоспособность системы. Резервное копирование и восстановление увеличивает текущую нагрузку на систему, что может вызывать замедление работы системы. Кроме того, в зависимости от вида резервного копирования и восстановления, может потребоваться монопольный доступ к системе или полная остановка ее работы.

Основная идея резервного копирования — создание копий критической части содержания резервируемой системы. Основными исключениями, как правило, не входящими в процедуру резервного копирования функционирующей ОС, являются каталоги, содержащие служебные данные, меняющиеся в процессе функционирования (`/dev`, `/media`, `/mnt`, `/parsecfs`, `/proc`, `/run`, `/sys`, `/tmp`), а также сетевые каталоги (смонтированная NFS, Samba и прочие виды сетевых данных).

Элементы системы резервного копирования должны включать необходимое оборудование, носители резервных копий и ПО. Для хранения резервных копий могут быть использованы различные носители информации: дисковые накопители, отчуждаемые носители информации или специально выделенные разделы жесткого диска. Тип и количество носителей определяются используемым оборудованием, объемами обрабатываемых данных и выбранной схемой резервирования данных. ПО резервного копирования и восстановления из состава ОС включает утилиты командной строки и распределенные системы управления хранилищами данных:

1) комплекс программ Bacula (18.3);
2) утилита копирования `rsync` (18.4);
3) утилиты архивирования `tar`, `cpio`, `gzip` (18.5).

ВНИМАНИЕ! Для восстановления мандатных атрибутов файлов из резервных копий процесс должен иметь PARSEC-привилегию `0x1000` (`PARSEC_CAP_UNSAFE_SETXATTR`). Привилегия может быть получена с использованием утилиты `execaps`:

```
sudo execaps -c 0x1000 tar .......
```

ВНИМАНИЕ! Восстановление расширенных атрибутов файлов с использованием `unsecure_setxattr` возможно только в случае, если атрибуты восстанавливаются с помощью системного вызова `setxattr` путем установки атрибута `security.PDPL`. Использование `unsecure_setxattr` не влияет на возможность изменения мандатных атрибутов файлов системными вызовами `pdpl_set_path`, `pdpl_set_fd`.

Комплекс программ Bacula позволяет системному администратору управлять процессами резервного копирования и восстановления данных, находить и восстанавливать утраченные или поврежденные файлы, а также проверять резервные копии, в том числе в гетерогенных сетях.

Утилита `rsync` предоставляет возможности для локального и удаленного копирования (резервного копирования) или синхронизации файлов и каталогов с минимальными затратами трафика.

Утилиты командной строки `tar`, `cpio`, `gzip` представляют собой традиционные инструменты создания резервных копий и архивирования ФС.

Порядок выполнения операций резервного копирования и восстановления объектов ФС с сохранением и восстановлением мандатных атрибутов и атрибутов аудита описан в РУСБ.10015-01 97 01-1.

18.1. Виды резервного копирования

Существуют следующие виды резервного копирования:

- полное резервное копирование — сохранение резервной копии всех файлов системы. Процедура занимает много времени и требует место для хранения большого объема. Как правило, выполняется в тех случаях, когда не влияет на основную работу системы, или для создания базовой резервной копии данных. В дальнейшем может выполняться дифференциальное или инкрементное резервное копирование;
- дифференциальное резервное копирование — сохранение копий изменившихся с последнего полного резервного копирования файлов. Требования к объему хранения и времени создания меньше, чем при полном копировании. Время восстановления незначительно за счет прямой перезаписи файлов;
- инкрементное резервное копирование — сохранение изменений файлов с момента последнего инкрементного копирования. Требует минимального количества времени и места для создания копии, но усложняет последующее восстановление, поскольку необходимо последовательное восстановление всех инкрементных копий с момента последнего полного резервного копирования.

18.2. Планирование резервного копирования

Планирование резервного копирования заключается в рассмотрении и определении следующих вопросов:

- что именно и как часто должно архивироваться;
- какие виды резервного копирования и на какие носители должны применяться;
- как часто и каким образом будут восстанавливаться файлы при необходимости;
- каким образом пользователи могут запросить ранее сохраненные файлы.

План резервного копирования должен периодически пересматриваться для отражения изменений как в системе, так и в используемых технологиях или условиях функционирования.

18.2.1. Составление расписания резервного копирования

При составлении расписания резервного копирования определяется что, когда и на каком носителе должно сохраняться.

Должна существовать возможность восстановления любого файла в любой момент времени. Например, требуется восстановить файл не более, чем однодневной давности. Для этого может использоваться комбинация полного и обновляемого (дифференциального или инкрементного) резервного копирования. Полное резервное копирование позволяет сохранить копии всех файлов системы, обновляемое — только изменившиеся со времени последнего архивирования. Обновляемое может иметь несколько уровней, например, обновление по отношению к последней обновляемой резервной копии.

Для восстановления отдельных файлов при таком многоуровневом расписании может понадобиться полная резервная копия, если файл не изменялся в течение месяца; копия первого уровня, если файл не изменялся в течение недели; копия второго уровня при ежедневной работе с этим файлом. Такая схема несколько сложнее, однако требует меньших ежедневных временных затрат.

Примечание. Расписание резервного копирования должно быть доведено до пользователей.

18.2.2. Планирование восстановления системы

При составлении плана резервного копирования следует определить:

1) план действий на случай аварийной ситуации;
2) как при необходимости может быть восстановлена система или отдельные файлы;
3) где хранятся и насколько доступны носители с резервными копиями и не могут ли они быть повреждены при сбоях на компьютере.

Примечание. Необходимо периодически выполнять проверку исправности носителей с архивами резервных копий. Проверка может включать в себя чтение содержимого копии после сохранения или выборочную проверку файлов резервной копии.

18.3. Комплекс программ Bacula

Bacula — это сетевая клиент-серверная система резервного копирования. Благодаря модульной архитектуре Bacula может масштабироваться от небольших автономных систем до больших сетей, состоящих из сотен компьютеров.

Bacula состоит из следующих основных компонентов:

- Bacula Director — центральная программа, координирующая все выполняемые операции (функционирует в фоновом режиме);
- Bacula Console — консоль Bacula, позволяющая администратору взаимодействовать с центральной программой;
- Bacula File — клиентская программа, устанавливаемая на каждый обслуживаемый компьютер;
- Bacula Storage — программа, обычно функционирующая на компьютере, к которому присоединены внешние устройства для хранения резервных копий;
- Catalog — программа, отвечающая за индексирование и организацию базы резервных данных.

Программа Bacula обеспечивает поддержку сохранения расширенных атрибутов каталогов и файлов и, при необходимости, их последующее восстановление (см. РУСБ.10015-01 97 01-1).

Все действия выполняются от имени учетной записи администратора с использованием механизма `sudo`.

Порядок использования Bacula описан на примере системы со следующей инфраструктурой:

- выделенный сервер `bacula1.my.dom` с IP-адресом `11.11.11.21` для функционирования Bacula Director — главный сервер, осуществляющий резервное копирование;
- выделенный сервер `bacula2.my.dom` с IP-адресом `11.11.11.22` для функционирования Bacula Storage — машина, на которой будут размещаться резервные копии данных;
- персональный компьютер `bacula3.my.dom` с IP-адресом `11.11.11.23` для функционирования Bacula File — машина, с которой будут копироваться данные и на которую будут восстанавливаться резервные копии данных.

18.3.1. Подготовка инфраструктуры

Для подготовки инфраструктуры к управлению системой резервного копирования необходимо выполнить следующие действия:

1) установить СУБД на сервер, где будет работать Bacula Director:

```
aptitude install postgresql
```

2) установить `pgadmin4` на сервер, где будет работать Bacula Director:

```
aptitude install pgadmin4
```

3) предполагается, что на всех машинах изначально установлены все пакеты, касающиеся Bacula, из состава ОС. Через менеджер пакетов Synaptic по ключевому слову «bacula» необходимо установить все пакеты, кроме тех, где в названии фигурирует «-sqlite3».
При настройке Bacula в появившемся интерфейсе настройки совместимости с БД в качестве имени БД необходимо указать `bacula` и пароль `bacula`.
В случае возникновения ошибки игнорировать ее на данном этапе, БД будет настроена позднее;

4) подготовить БД для Bacula, выполнив следующие действия:
- в файле `/etc/postgresql/11/main/postgresql.conf` указать `listen_addresses = '*'`;

- в файле `/etc/postgresql/11/main/pg_hba.conf` внести необходимые изменения, для простоты можно указать метод `trust` для всех соединений, удалить любую дополнительную конфигурацию после метода типа `mod=`;

- обязательно добавить `host` с IP-адресом, где будет работать `bacula-dir`. В случае если все службы Bacula будут установлены на одну машину, указывать IP-адрес не обязательно, т. к. работа будет выполняться через `localhost`.

Пример
Файл `pg_hba.conf`

```
local all postgres trust
local all all trust
host all all 127.0.0.1/32 trust
host all all 11.11.11.21/24 trust
```

- выполнить запуск БД:

```
pg_ctlcluster 11 main restart
```

- присвоить пароль `postgres`:

```
passwd postgres
```

- присвоить для Bacula пароль bacula:

```
passwd bacula
```

- создать пользователя БД для работы с Bacula (выполнять не от имени учетной записи администратора):

```
# psql template1 postgres
postgres=# CREATE ROLE bacula;
postgres=# ALTER USER bacula PASSWORD 'bacula';
postgres=# ALTER USER bacula LOGIN SUPERUSER CREATEDB CREATEROLE;
```

5) создать БД bacula (выполнять не от имени учетной записи администратора):

- выполнить pgadmin4;

- указать имя template1, пользователя postgres, пароль postgres;

- в секции Роли входа добавить роль входа bacula. Создать БД bacula, владельцем назначить bacula;

6) на сервере bacula1.my.dom необходимо запустить сценарии, которые создадут все необходимые таблицы и привилегии, предварительно отредактировав их:

- в /usr/share/bacula-director/make_postgresql_tables внести следующие изменения:
  - в строке db_name указать имя –bacula;
  - в строке psql после psql вписать –U bacula;
- в /usr/share/bacula-director/grant_postgresql_privileges внести следующие изменения:
  - в строке db_user указать имя –bacula;
  - в строке db_name указать имя –bacula;
  - в строке db_password указать пароль bacula;
  - в строке $bindir/psql после psql вписать –U bacula;
- сохранить изменения и выполнить сценарии:

```
make_postgresql_tables
grant_postgresql_privileges
```

7) на машине, где будет работать Bacula Storage, необходимо создать каталог /back, в котором будут храниться резервные копии данных, и присвоить каталогу владельца bacula:

```
mkdir /back
chown –R bacula /back
```

8) на машине, где будет работать Bacula File, необходимо создать каталог /etc2, в который будут восстанавливаться данные из резервной копии:

```
mkdir /etc2
```

Если подготовительные настройки выполнены корректно, БД стартует без ошибок и сценарии выполнились без ошибок, то можно приступить к настройке Bacula.

18.3.2. Настройка Bacula

Подготовка Bacula к работе заключается в настройке каждого компонента в отдельности и последующей настройке их взаимодействия.

18.3.2.1. Настройка Bacula Director

Настройка Bacula Director осуществляется путем корректировки конфигурационного файла `/etc/bacula/bacula-dir` сервера `bacula1.my.dom`.

В первую очередь необходимо определить основные параметры в секции `Director`. На начальном этапе важно установить параметры `Name` и `Password`. `Name` задает уникальное имя Bacula Director, а `Password` — пароль, который будет использоваться при соединениях `BC` с `DD`. Остальные параметры можно оставить со значениями по умолчанию:

```
Director { # define myself
Name = bacula-dir
DIRport = 9101 # where we listen for UA connections
QueryFile = "/etc/bacula/scripts/query.sql"
WorkingDirectory = "/var/lib/bacula"
PidDirectory = "/var/run/bacula"
Maximum Concurrent Jobs = 1
Password = "1" # Console password
Messages = Daemon
DirAddress = 11.11.11.21
}
```

Следующей группой параметров, которые необходимо определить, является секция `Catalog`. В ней необходимо указать реквизиты доступа к БД, а также назначить уникальное имя данного Bacula Catalog с помощью параметра `Name`:

```
Catalog {
Name = MyCatalog
# Uncomment the following line if you want the dbi

PS. driver
# dbdriver = "dbi:sqlite3"; dbaddress = 127.0.0.1; dbport =
dbname = "bacula"; dbuser = "bacula"; dbpassword = "bacula"
DB Address = 11.11.11.21
}
```

Далее необходимо определить SD, на который будет производиться передача данных для дальнейшей записи на устройство хранения. Когда Bacula Storage настроен и готов к работе, необходимо определить реквизиты доступа к нему в секции Storage файла bacula-dir.conf. Основные параметры:

1) Name — уникальное имя, использующееся для адресации секции Storage в рамках файла bacula-dir.conf;

2) Device и MediaType — дублируют одноименные параметры файла bacula-sd.conf;

3) Password — содержит пароль, который будет использоваться при подключении к Bacula Storage:

```
Storage {
Name = File
# Do not use "localhost" here
Address = 11.11.11.22 # N.B. Use a fully qualified name here
SDPort = 9103
Password = "1"
Device = FileStorage
Media Type = File
}
```

Секция Pool определяет набор носителей информации и параметры, используемые SD при их обработке. Каждый Pool взаимодействует с устройством хранения данных, поэтому необходимо создать столько пулов, сколько определено устройств хранения. Фактически если для каждого Bacula File определено отдельное устройство, то для каждого FD необходимо определить и Pool. Основные параметры:

1) Name — определяет уникальное имя пула;

2) Pool Type — определяет тип, для резервных копий должно быть установлено значение Backup;

3) Maximum Volume Jobs — рекомендуется установить значение 1. Данное значение указывает, что в рамках одного носителя данных могут быть размещены резервные данные, полученные в ходе выполнения только одного задания. Если размер созданной резервной копии намного меньше размера носителя, то возможно сохранять на него резервные копии, которые будут создаваться в будущем. Но если говорится о файлах, то желательно придерживаться правила «один файл — одна копия», т. е. в одном файле Bacula должны храниться резервные данные, которые были сформированы в рамках выполнения одного задания. Для каждого последующего будут создаваться новые файлы;

4) Volume Retention — время, по прошествии которого данные о резервной копии, хранящейся на носителе, будут удалены из каталога. Для обеспечения работоспособности Bacula при указании значения данного параметра необходимо учитывать,

что информация обо всех зарезервированных файлах хранится в БД, по записи на каждый файл. Если резервируются тысячи файлов, то за непродолжительное время БД станет очень большой, что может затруднить работу Bacula. Поэтому важно своевременно очищать БД от устаревшей информации. При этом сам носитель информации не будет очищен автоматически. Он будет промаркирован как устаревший, но всегда можно будет использовать его для восстановления данных в ручном режиме;

5) `Maximum Volumes` — максимальное количество носителей (в данном случае файлов), доступных в пуле;

6) `Recycle` — указывает на необходимость повторного использования носителей, помеченных как устаревшие. При этом реальная перезапись носителя произойдет лишь в случае, когда свободных носителей не останется. Свободные носители определяются из параметра `Maximum Volumes`;

7) `AutoPrune` — указывает на необходимость удаления устаревших записей из Bacula Catalog автоматически после завершения выполнения очередного задания;

8) `Label Format` — определяет префикс, который будет использован Bacula для маркирования носителей информации, в данном случае — для именования файлов;

9) `Storage` — указывает на имя устройства хранения данных, указанного в параметре `Name` секции `Storage` файла `bacula-dir.conf`.

```
Pool {
Name = Default
Pool Type = Backup
Recycle = yes # Bacula can automatically recycle Volumes
AutoPrune = yes # Prune expired volumes
Volume Retention = 1 month # one year
Maximum Volume Jobs = 1
Maximum Volumes = 32
Storage = File
Label Format = "volume-"
}
```

Секция `FileSet` позволяет предопределить несколько наборов резервируемых файлов. Например, один набор для Windows, другой — для Linux или один для серверов, а другой — для рабочих станций. Параметр `Name` определяет уникальное имя набора.

Секция `Include` содержит пути к резервируемым файлам/каталогам, а `Exclude` — пути к файлам и каталогам, которые необходимо исключить из списка резервируемых. В секции `Include` возможна секция `Options`, в которой определяются параметры резервирования. Основные параметры:

1) `signature` — указывает алгоритм вычисления контрольных сумм файлов;

2) `compression` — указывает алгоритм компрессии файлов;

3) `recurse` — указывает на необходимость рекурсивного резервирования, включая подкаталоги и файлы;

4) `File` — указывает копируемый каталог;

5) `xattrsupport` — указывает на возможность включения поддержки расширенных атрибутов, это обязательный параметр для работы с метками безопасности:

```
FileSet {
Name = "Catalog"
Include {
Options {
signature = MD5
compression = GZIP
# recurse = yes
aclsupport = yes
xattrsupport = yes
}
File = /etc
}
}
```

Все настройки связываются воедино с помощью секции `Job`, в которой дается задание планировщику по выполнению резервирования данных. Основные параметры:

1) `Type` — указывает на тип задания. Типов существует несколько. Здесь достаточно указать `Backup`;

2) `Schedule` — указывает на предопределенное расписание, согласно которому будет выполняться резервирование данных. Все расписания определены в файле `bacula-dir.conf`;

3) `Where` — указывает на каталог, в котором будут восстанавливаться данные из резервной копии;

4) `Write Bootstrap` — указывает путь к файлу, в который будет сохраняться информация, позволяющая восстанавливать данные из резервной копии без наличия подключения к Bacula Catalog. Вместо `%n` будет подставлено значение параметра `Name`:

```
Schedule {
Name = "DailyCycle"
Run = Full daily at 16:10
# Run = Differential 2nd-5th sun at 23:05
Run = Incremental mon-sat at 23:05
}

Job {
Name = "RestoreFiles"
```

```
Type = Restore
Client= bacula-fd
FileSet="Catalog"

Storage = File
Pool = Default
Messages = Standard
Where = /etc2
}

Job {
Name = "BackupCilent1"
Type = Backup
Client = bacula-fd
FileSet = "Catalog"
Schedule = "DailyCycle"
Messages = Standard
Pool = Default
Write Bootstrap = "/var/lib/bacula/Client1.bsr"
Priority = 1
}
```

Затем необходимо указать параметры единственного Агента:

```
Client {
Name = bacula-fd
Address = 11.11.11.23
FDPort = 9102
Catalog = MyCatalog
Password = "1" # password for FileDaemon
File Retention = 30 days # 30 days
Job Retention = 6 months # six months
AutoPrune = yes # Prune expired Jobs/Files
}
```

Остальные секции (`Job`, `JobDefs`, `Client` и `Console`) необходимо закомментировать. Трафик данных будет идти по портам, указанным в конфигурационных файлах каждого из компонентов Bacula.

Настроить доступ к `DD` со стороны Bacula Console в файле `/etc/bacula/bconsole.conf` сервера `bacula1.my.dom`:

```
Director {
Name = bacula-dir
```

```
DIRport = 9101
address = 11.11.11.21
Password = "1"
}
```

На машине, где будет функционировать Bacula Director, следует удалить пакеты `bacula-sd` и `bacula-fd`:

```
apt remove bacula-sd
apt remove bacula-fd
```

Конфигурационные файлы `bacula-sd` и `bacula-fd` в `/etc/bacula` следует переименовать либо удалить.

Службы `bacula-sd` и `bacula-fd` остановить:

```
systemctl stop bacula-sd
systemctl stop bacula-fd
```

18.3.2.2. Настройка Bacula Storage

Bacula Storage отвечает за непосредственную работу с устройством хранения данных. Bacula поддерживает широкий спектр устройств от оптических дисков до полнофункциональных ленточных библиотек. В описываемой системе используется самый распространенный вариант — жесткий диск с существующей файловой системой (например, ext3).

Для настройки Bacula Storage необходимо на сервере `bacula2.my.dom` отредактировать конфигурационный файл `/etc/bacula/bacula-sd.conf`.

В секции основных параметров `Storage` определить параметр `Name`, который задает уникальное имя Bacula Storage. Для остальных параметров возможно оставить значения по умолчанию.

Секция `Director` необходима для указания уникального имени `DD` и пароля, с которым данный `DD` может подключаться к `SD`. Секций `Director` в файле может быть несколько, что дает возможность использовать единый сервер хранения данных для нескольких систем резервирования. Все остальные секции `Director`, найденные в файле, необходимо закомментировать:

```
Storage { # definition of myself
Name = bacula-sd
SDPort = 9103 # Director’s port
```

```
WorkingDirectory = "/var/lib/bacula"
Pid Directory = "/var/run/bacula"
Maximum Concurrent Jobs = 20
SDAddress = 11.11.11.22
}

Director {
Name = bacula-dir
Password = "1"
}
```

Основные настройки, определяющие взаимодействие с устройствами хранения, находятся в секции `Device`. Параметры, необходимые для хранения резервных копий в рамках существующей ФС, примонтированной в каталог `/back`:

1) `Name` — определяет уникальное имя подключенного устройства. Если планируется создавать изолированные друг от друга резервные копии для каждого из Bacula File, то необходимо создать несколько секций `Device` с уникальными именами. В противном случае резервируемые файлы со всех `FD` будут размещаться в одном и том же файле, что может затруднить дальнейшее обслуживание системы;

2) `Media Type` — определяет произвольное уникальное имя, которое будет использоваться Bacula при восстановлении данных. Согласно ему определяется устройство хранения, с которого будет производиться восстановление. Если резервные копии хранятся в файлах, то для каждой секции `Device` должен быть задан уникальный `Media Type`;

3) `Archive Device` — указывает путь к файлу устройства в каталоге `/dev` или путь к каталогу, в котором будут размещаться резервные копии;

4) `Device Type` — определяет тип устройства. Для размещения в существующей ФС указывается `File`;

5) `Random Access` — указывает на возможность случайной (непоследовательной) адресации. Для файлов указывается `Yes`;

6) `RemovableMedia` — указывает, возможно ли извлечение устройства хранения. Необходимо для ленточных устройств, приводов оптических дисков и т. д. Для файлов устанавливается значение `No`;

7) `LabelMedia` — указывает на необходимость автоматического маркирования носителей информации:

```
Device {
Name = FileStorage
Media Type = File
Archive Device = /back
LabelMedia = yes; # lets Bacula label unlabeled media
```

```
Random Access = Yes;
AutomaticMount = yes; # when device opened, read it
RemovableMedia = no;
AlwaysOpen = no;
}
```

На машине, где будет функционировать Bacula Storage, следует удалить пакет `bacula-fd`:

```
apt remove bacula-fd
```

Конфигурационный файл `bacula-fd` в `/etc/bacula` следует переименовать либо удалить.

Службу `bacula-fd` остановить:

```
systemctl stop bacula-fd
```

18.3.2.3. Настройка Bacula File

Для настройки Bacula File на рабочей станции `bacula3.my.dom` используется конфигурационный файл `/etc/bacula/bacula-fd`. Для базовой настройки достаточно определить параметры секций `Director` и `FileDaemon`.

В секции `Director` указывается пароль, который будет использовать `DD` при подключении к `FD`. Секций `Director` в файле может быть несколько, все остальные секции `Director`, найденные в файле, необходимо закомментировать:

```
Director {
Name = bacula-dir
Password = "1"
}
```

В секции `FileDaemon` указываются настройки `FD`, в ней необходимо определить параметр `Name`, в котором указывается уникальное имя Bacula File:

```
FileDaemon { # this is me
Name = bacula-fd
FDport = 9102 # where we listen for the director
WorkingDirectory = /var/lib/bacula
Pid Directory = /var/run/bacula
Maximum Concurrent Jobs = 20
FDAddress = 11.11.11.23
}
```

На машине, где будет функционировать Bacula File, следует удалить пакет `bacula-sd`:

```
apt remove bacula-sd
```

Конфигурационный файл `bacula-sd` в `/etc/bacula` следует переименовать либо удалить.

Службу `bacula-sd` следует остановить:

```
systemctl stop bacula-sd
```

Далее необходимо запустить все компоненты соответствующими командами, выполненными на соответствующих серверах:

```
systemctl restart bacula-director
systemctl restart bacula-sd
systemctl restart bacula-fd
```

18.3.2.4. Проверка Bacula

После настройки Bacula Director, Bacula Storage и Bacula File программа Bacula готова к работе. Управление Bacula осуществляется через `bconsole`. Настройки каталогов, заданий, расписаний и других функций выполняются в конфигурационных файлах.

Для тестовой проверки необходимо:

- выполнить `bconsole`;
- выполнить `run`;
- выбрать `job 1`;
- войти в меню, набрав `mod`;
- выбрать `1` (Level);
- выбрать `1` (Full);
- подтвердить выполнение, набрав `yes`.

В результате будет создана резервная копия данных в каталоге `/back` на машине с Bacula Storage.

Для восстановления объектов ФС с установленными мандатными атрибутами необходимо запустить консоль управления Bacula с PARSEC-привилегией 0x1000, выполнив команду:

```
sudo execaps -c 0x1000 -- bconsole
```

Для восстановления данных из резервной копии необходимо:

- выполнить `restore`;
- выбрать пункт `12`;
- ввести номер `job id`;
- указать параметр маркировки `mark *`;
- подтвердить выполнение командой `done`.

Данные из резервной копии будут восстановлены в каталоге `/etc2` на машине с Bacula File.

Также управление Bacula возможно с помощью графической утилиты `bacula-console-qt`.

18.4. Утилита копирования `rsync`

Все действия при использовании команды `rsync` выполняются от имени учетной записи администратора с использованием механизма `sudo`.

В таблице 58 приведены некоторые наиболее часто используемые параметры команды `rsync`.

Таблица 58

Параметр	Назначение
`-v, --verbose`	Подробный вывод
`-z, --compress`	Сжимать трафик
`-r, --recursive`	Выполнять копирование рекурсивно
`-p, --perms`	Сохранять дискретные права доступа
`-t, --times`	Сохранять время доступа к файлам
`-g, --group`	Сохранять группу
`-o, --owner`	Сохранять владельца
`-A, --acls`	Сохранять списки контроля доступа ACL (включает `-p`)
`-X, --xattrs`	Сохранять расширенные атрибуты (в том числе мандатные атрибуты)

Подробное описание команды приведено в `man` для `rsync`.

Пример

Следующая команда сделает копию домашнего каталога на 192.168.0.1

```
sudo rsync -vzrptgoAX /home/ admin@192.168.0.1:/home_bak
```

В данном примере должен быть создан каталог `/home_bak` на сервере и установлены на него максимальные метки с `ccnr`.

ВНИМАНИЕ! Не рекомендуется использовать параметр `-l` для копирования символических ссылок при создании резервной копии домашних каталогов пользователей.

18.5. Утилиты архивирования

При создании архива командами `tar` и `gzip` передается список файлов и каталогов, указываемых как параметры командной строки. Любой указанный каталог просматривается рекурсивно. При создании архива с помощью команды `cpio` ей предоставляется список объектов (имена файлов и каталогов, символические имена любых устройств, гнезда доменов UNIX, поименованные каналы и т. п.).

Все действия при использовании команд `tar`, `cpio` и `gzip` выполняются от имени учетной записи администратора с использованием механизма `sudo`.

Подробное описание команд приведено в руководстве `man` для `tar`, `cpio` и `gzip`.

18.5.1. tar

Команда `tar` может работать с рядом дисковых накопителей, позволяет просматривать архивы в ОС.

В таблице 59 приведены основные параметры команды `tar`.

Таблица 59

Параметр	Назначение
`--acls`	Сохраняет (восстанавливает) списки контроля доступа (ACL) каталогов и файлов, вложенных в архив
`-c, --create`	Создает архив
`-x, --extract, --get`	Восстанавливает файлы из архива на устройстве, заданном по умолчанию или определенном параметром `f`
`--xattrs`	Сохраняет (восстанавливает) расширенные атрибуты каталогов и файлов, вложенных в архив
`-f, --file name`	Создает (или читает) архив с `name`, где `name` — имя файла или устройства, определенного в `/dev`, например, `/dev/rmt0`
`-Z, --compress, --uncompress`	Сжимает или распаковывает архив с помощью `compress`
`-z, --gzip, --gunzip`	Сжимает или распаковывает архив с помощью `gzip`
`-M, --multi-volume`	Создает многотомный архив
`-t, --list`	Выводит список сохраненных в архиве файлов
`-v, --verbose`	Выводит подробную информацию о процессе

Подробное описание команды приведено в man для tar.

В примерах приведены варианты использования команды tar.

Примеры:

1. Копирование каталога /home на специальный раздел жесткого диска /dev/hda4

```
tar -cf /dev/hda4 /home
```

Параметр f определяет создание архива на устройстве /dev/hda4.

2. Применение сжатия при архивировании

```
tar -cvfz /dev/hda4 /home | tee home.index
```

Параметр v заставляет tar выводить подробную информацию, параметр z указывает на сжатие архива с помощью утилиты gzip. Список скопированных файлов направляется в home.index.

3. Использование команды find для поиска измененных в течение одного дня файлов в каталоге /home и создание архива home.new.tar с этими файлами:

```
find /home -mtime 1 -type f -exec tar -rf home.new.tar {} \;
```

4. Если надо посмотреть содержимое архива, то можно воспользоваться параметром -t команды tar:

```
tar -tf home.new.tar
```

5. Для извлечения файлов из архива необходимо указать путь к архиву либо устройству и путь к месту извлечения. Если архив (каталога /home) был создан командой:

```
tar -czf /tmp/home.tar /home
```

то извлекать его надо командой:

```
tar -xzf /tmp/home.tar /
```

6. Использование команды tar для создания архивов в ФС ОС, а не только на устройствах для архивирования (можно архивировать группу файлов с их структурой каталогов в один файл, для чего передать имя создаваемого файла с помощью параметра f вместо имени устройства)

```
tar cvf /home/backup.tar /home/dave
```

С помощью tar архивируется каталог с вложенными подкаталогами.

При этом создается файл /home/backup.tar, содержащий архив каталога /home/dave и всех файлов в его подкаталогах.

Обычно при использовании команды tar следует делать входом верхнего уровня каталог. В таком случае файлы при восстановлении будут располагаться в подкаталоге рабочего каталога.

Предположим, в рабочем каталоге имеется подкаталог data, содержащий несколько сотен файлов. Существует два основных пути создания архива этого каталога. Можно войти в подкаталог и создать в нем архив, например:

```
pwd
/home/dave
cd data
pwd
/home/dave/data
tar cvf .. /data.tar *
```

Будет создан архив в каталоге /home/dave, содержащий файлы без указания их расположения в структуре каталогов. При попытке восстановить файлы из архива data.tar подкаталог не будет создан, и все файлы будут восстановлены в текущем каталоге.

Другой путь состоит в создании архива каталога, например:

```
pwd
/home/dave
tar cvf data.tar data
```

Будет создан архив каталога, в котором первой будет следовать ссылка на каталог. При восстановлении файлов из такого архива будет создан подкаталог в текущем каталоге, и файлы будут восстанавливаться в нем.

Можно автоматизировать выполнение данных команд, поместив их в файл crontab суперпользователя. Например, следующая запись в файле crontab выполняет резервное копирование каталога /home ежедневно в 01:30:

```
30 01 *** tar -cvfz /dev/hda4 /home > home index
```

При необходимости более сложного архивирования используется язык сценариев оболочки, которые также могут быть запущены с помощью cron (см. 4.3.1.2).

Порядок использования команды `tar` для сохранения и восстановления мандатных атрибутов файлов описан в РУСБ.10015-01 97 01-1.

18.5.2. cpio

Для копирования файлов используется команда общего назначения `cpio`.

Команда используется с параметром `-o` для создания резервных архивов и с параметром `-i` — для восстановления файлов. Команда получает информацию от стандартного устройства ввода и посылает выводимую информацию на стандартное устройство вывода.

Команда `cpio` может использоваться для архивирования любого набора файлов и специальных файлов. Она пропускает сбойные сектора или блоки при восстановлении данных, архивы могут быть восстановлены в ОС

Недостатком команды `cpio` является необходимость использовать язык программирования оболочки для создания соответствующего сценария, чтобы обновить архив.

В таблице 60 приведены основные параметры команды `cpio`.

Таблица 60

Параметр	Назначение
`-o`	Создание архива в стандартное устройство вывода
`-l`	Восстановление файлов из архива, передаваемого на стандартное устройство ввода
`-t`	Создание списка содержимого стандартного устройства ввода

Подробное описание команды приведено в `man cpio`.

Примеры:

1. Копирование файлов из каталога `/home` в архив `home.cpio`

```
find /home/* | cpio -o > /tmp/home.cpio
```

2. Восстановление файлов из архива `home.cpio` с сохранением дерева каталогов и создание списка в файле `bkup.index`

```
cpio -id < /tmp/home.cpio > bkup.index
```

3. Использование команды `find` для поиска измененных за последние сутки файлов и сохранение их в архив `home.new.cpio`

```
find /home -mtime 1 -type f | cpio -o > /tmp/home.new.cpio
```

4. Восстановление файла `/home/dave/notes.txt` из архива `home.cpio`

```
cpio -id /home/dave/notes.txt < home.cpio
```

Для восстановления файла с помощью `cpio` следует указывать его полное имя.

Можно автоматизировать выполнение данных команд, поместив их в файл `crontab` суперпользователя. Например, следующая запись в файле `crontab` выполняет резервное копирование каталога `/home` ежедневно в 01:30:

```
30 01 *** ls /home : cpio -o > /tmp/home.cpio
```

При необходимости более сложного резервного копирования можно создать соответствующий сценарий оболочки. Запуск подобных сценариев также может быть осуществлен посредством `cron`.

Создание резервных копий означает определение политики создания резервных копий для снижения потерь и восстановления информации после возможной аварии системы.

19. КОНТРОЛЬ ПОДКЛЮЧАЕМЫХ УСТРОЙСТВ

В ОС поддерживается разграничение доступа к символьным и блочным устройствам, для которых в каталоге `/dev` создаются файлы устройств. Для разграничения доступа к устройствам типа видеокарт, сетевых карт и т. д. данный метод не используется.

Для контроля подключаемых устройств и решения задачи разграничения доступа к устройствам в ОС реализованы:

- средство разграничения доступа к устройствам на основе правил менеджера устройств `udev`;
- средство регистрации (учета) устройств.

Включение и выключение контроля подключаемых устройств описано в 19.1.

Порядок монтирования подключаемых устройств описан в 19.2.

Средство разграничения доступа к устройствам на основе генерации правил `udev` обеспечивает дискреционное и мандатное управление доступом пользователей к устройствам, подключаемым, в первую очередь, через интерфейс USB: сканерам, съемным накопителям, видеокамерам и т. п. Описание правил генерации и порядок их применения приведены в 19.3–19.7.

Средство регистрации устройств обеспечивает учет подключаемых устройств и съемных носителей в системе, установку дискреционных и мандатных атрибутов доступа пользователей к устройствам и создание дополнительных правил доступа к устройству (например, ограничение на подключение устройства только в определенный USB-порт). Описание порядка регистрации устройств приведено в 19.9.

Примечание. Присвоение устройствам мандатных атрибутов доступа, а также мандатное управление доступом к устройствам реализуется только на уровне защищенности «Смоленск» при включенном мандатном управлении доступом. Правила разграничения доступа к устройствам, применяемые на уровне защищенности, отличном от «Смоленск», не должны содержать мандатные атрибуты доступа.

19.1. Включение и выключение контроля подключения устройств

Включение и выключение контроля подключения устройств на основе правил менеджера устройств `udev` используется инструмент командной строки `pdac-adm`.

Синтаксис команды:

```
sudo pdac-adm state [параметр]
```

Выполнение команды без параметра отображает текущее состояние контроля подключения устройств.

Описание параметров инструмента приведено в таблице 61.

Таблица 61

Параметр	Описание
`enable`	Включение средства контроля подключения устройств
`disable`	Выключение средства контроля подключения устройств

Для вступления изменений в силу требуется перезагрузка ОС.

19.2. Монтирование съемных накопителей

При монтировании блочных устройств используется утилита `mount`, модифицированная для монтирования устройства владельцем или пользователем, входящим в группу-владельца.

Процесс монтирования с использованием командной строки доступен только администратору. При этом необходимо указать два параметра: наименование файла устройства и наименование точки монтирования. Остальные параметры монтирования выбираются из файлов `/etc/fstab` и `/etc/fstab.pdac` с использованием регулярных выражений.

Для непривилегированных пользователей доступен процесс монтирования с использованием графических утилит `fly-fm` или `fly-wm`. При этом монтирование ФС съемных накопителей осуществляется в каталог `/run/user/$uid/media`.

Для предоставления локальным пользователям возможности монтирования ФС съемных накопителей необходимо наличие в файле `/etc/fstab` следующей записи:

```
/dev/s*  /home/*/media/*  auto owner,group,noauto,noexec 0 0
```

По умолчанию для монтирования различных ФС, содержащихся в учтенных разделах на блочных устройствах USB-накопителей, в файл `/etc/fstab.pdac` включены следующие записи:

```
/dev/*fat  /run/user/*/media/*  auto
owner,group,noauto,nodev,noexec,iocharset=utf8,defaults  0  0
/dev/*ntfs*  /run/user/*/media/*  auto
owner,group,noauto,nodev,noexec,iocharset=utf8,defaults  0  0
/dev/sd*ext*  /run/user/*/media/*  auto
owner,group,noauto,nodev,noexec,defaults  0  0
```

По умолчанию для монтирования различных ФС, содержащихся на учтенных CD/DVD-дисках, в файл /etc/fstab.pdac включены следующие записи:

```
/dev/s*udf  /run/user/*/media/*  udf
owner,group,nodev,noexec,noauto,defaults  0  0
/dev/s*iso9660  /run/user/*/media/*  iso9660
owner,group,nodev,noexec,noauto,defaults  0  0
```

По умолчанию монтирование ФС, содержащихся в неучтенных разделах на блочных устройствах USB-накопителей, разрешено пользователям, входящим в группу floppy. В данном случае монтирование будет осуществляться в соответствии со следующей записью из файла /etc/fstab.pdac:

```
/dev/sd*  /run/user/*/media/*  auto
owner,group,noauto,nodev,noexec,iocharset=utf8,defaults  0  0
```

Для возможности монтирования ФС ext*, содержащихся в неучтенных разделах на блочных устройствах USB-накопителей, необходимо в файле /etc/fstab.pdac из записи для устройств /dev/sd* удалить неподдерживаемый для данной ФС параметр монтирования iocharset=utf8:

```
/dev/sd*  /run/user/*/media/*  auto
owner,group,noauto,nodev,noexec,defaults  0  0
```

Для монтирования пользователями ФС, содержащихся на неучтенных CD/DVD-дисках, в конец файла /etc/fstab необходимо включить следующую запись:

```
/dev/sr*  /*home/*/media/*  udf,iso9660  user,noauto  0  0
```

ВНИМАНИЕ! При монтировании ФС, поддерживающей атрибуты UNIX и расширенные атрибуты, права доступа на файл учтенного устройства не будут совпадать с правами доступа в ФС. Использование мандатных атрибутов будет ограничено атрибутами, установленными для файла устройства.

ВНИМАНИЕ! Использование учтенного USB-носителя с ФС VFAT возможно только при входе в систему на том уровне конфиденциальности, который назначен администратором для этого устройства.

ВНИМАНИЕ! При включении режима работы с отчуждаемыми носителями с конфиденциальной информацией все непривилегированные пользователи должны быть исключены из группы floppy.

ВНИМАНИЕ! При включении режима работы с CD/DVD-дисками с конфиденциальной информацией все непривилегированные пользователи должны быть исключены из группы `cdrom`.

ВНИМАНИЕ! Использование учтенного USB-носителя с ФС ext4 (ext3) возможно пользователями на разных доступных им уровнях конфиденциальности. При этом администратор должен зарегистрировать носитель для данного пользователя на требуемых уровнях и создать на ФС носителя систему каталогов с необходимыми уровнями конфиденциальности. Например, для обеспечения работы на нескольких уровнях на USB-носителе с ФС ext4 администратор может использовать следующий сценарий, задав необходимые переменные `USERNAME` и `DEVICE`:

```
USERNAME="user"
DEVICE="/dev/sdc1"
mkfs.ext4 $DEVICE
mkdir -p /media/usb
mount $DEVICE /media/usb
#multilevel
pdpl-file 3:0:-1:ccnr /media/usb/
mkdir /media/usb/{0,1,2,3}
pdpl-file 0:0:0:0 /media/usb/0
pdpl-file 1:0:0:0 /media/usb/1
pdpl-file 2:0:0:0 /media/usb/2
pdpl-file 3:0:0:0 /media/usb/3
chown -R ${USERNAME}:${USERNAME} /media/usb/{0,1,2,3}
ls -la /media/usb/
pdp-ls -M /media/usb/
umount /media/usb
```

19.3. Перехват события менеджером устройств udev

Менеджер устройств `udev` перехватывает события, возникающие при изменении статуса подключенных устройств. Основные события:

- подключение устройства (событие `add`);
- отключение устройства (событие `remove`).

Перехват событий осуществляется на основе правил `udev`. Правила перехвата событий записываются в файлы с расширением `.rules` и располагаются в нескольких каталогах, при этом каталог определяет приоритет правил. Правила обрабатываются в следующей последовательности:

1) правила из каталога `/lib/udev/rules.d/`;
2) правила из каталога `/run/udev/rules.d/`;

3) правила из каталога /etc/udev/rules.d/.

Перед обработкой правил файлы упорядочиваются по алфавиту. Файлы с одинаковыми именами перезаписываются последним найденным файлом, например файл, найденный в каталоге /etc/udev/rules.d/, перезапишет ранее найденный одноименный файл.

Пример

Правило перехвата события /etc/udev/rules.d/99-local.rules

```
KERNEL=="sd[a-z][0-9]", SUBSYSTEMS=="usb", ACTION=="add",
     RUN+="/bin/systemctl start usb-mount@%k.service"
KERNEL=="sd[a-z][0-9]", SUBSYSTEMS=="usb", ACTION=="remove",
     RUN+="/bin/systemctl stop usb-mount@%k.service"
```

Данное правило обрабатывает события подключения (add) и отключения (remove) дисковых устройств с именами, начинающимися с букв sd, после которых следует одна любая строчная буква ([a-z]) и одна цифра ([0-9]).

Правило при этом не выполняет прямых действий, а вызывает системную службу usb-mount@%k.service, то есть вызывает сценарий обработки события как системную службу.

При выполнении сценария обработки событий служба udev вместо переменной %k подставляет имя устройства, т. е. при подключении устройства /dev/sdb1 будет выполняться команда:

```
systemctl start usb-mount@sdb1.service
```

При запуске службы, имя которой содержит символ @, системный диспетчер служб разделит это имя на составляющие и передаст часть, находящуюся после символа @, в качестве параметра вызываемой службы. Таким образов, вызов:

```
systemctl start usb-mount@sdb1.service
```

будет обработан как вызов службы usb-mount с параметрами start и sdb1.

Подробное описание параметров и переменных правил udev приведено в руководстве man udev.

19.4. Разграничение доступа к устройствам на основе генерации правил udev

Разграничение доступа к устройству осуществляется на основе генерации правил для менеджера устройств udev, которые хранятся в соответствующих файлах в каталогах

/etc/udev/rules.d и /run/udev/rules.d. Генерация правил осуществляется автоматически для символьных и блочных устройств с использованием локальной базы учета устройств (файл /etc/parsec/PDAC/devices.cfg) либо базы учета устройств FreeIPA (см. раздел 8).

Для устройств, учитываемых в локальной базе, генерация правила udev осуществляется при сохранении информации об устройстве с использованием модуля «Устройства и правила» графической утилиты astra-systemsettings («Параметры системы», см. электронную справку). Для вызова модуля можно использовать команду:

```
astra-systemsettings astra_kcm_devices_and_rules
```

Для устройств, учтенных в базе FreeIPA, генерация правил udev осуществляется службой sssd на хосте при входе пользователя домена. Генерация правил осуществляется для всех устройств, учтенных для данного пользователя, на основе прав доступа к ним, также определенных в базе FreeIPA.

Пример

Правило для съемного USB-накопителя

```
ENV{ID_SERIAL}=="JetFlash_TS256MJF120_OYLIXNA6-0:0", OWNER="user",
     GROUP="users" PDPL="3:0:f:0!:" AUDIT="0:0x0:0x0"
```

Правило для съемного USB-накопителя с серийным номером JetFlash_TS256MJF120_OYLIXNA6-0:0 разрешает использование данного накопителя владельцу устройства (пользователю user) и пользователям, входящим в группу users. Для устройства флаги аудита не установлены, но установлены мандатные атрибуты:

- уровень конфиденциальности — 3;
- категория целостности — 0;
- категории — f;
- роли и административные роли отсутствуют.

19.5. Вызов сценария обработки события как системной службы

Для вызова системных служб используются соответствующие юниты службы systemd, расположенные в каталоге /etc/systemd/system/.

Юнит с именем `usb-mount@.service` для вызова службы перехвата события `udev` может быть записан в следующем виде:

```
[Unit]
Description=Mount USB Drive on %i
[Service]
Type=oneshot
RemainAfterExit=true
ExecStart=/usr/local/bin/usb-mount.sh add %i
ExecStop=/usr/local/bin/usb-mount.sh remove %i
```

Данный юнит при выполнении команд `start` и `stop` вызывает исполняемый файл сценария обработки события `/usr/local/bin/usb-mount.sh`.

При вызове сценария вместо параметра `%i` будет подставлена часть имени вызова службы, находящаяся после символа `@`.

19.6. Сценарий обработки события

Сценарий обработки события может быть размещен в любом каталоге.

Пример

Сценарий обработки события `/usr/local/bin/usb-mount.sh`

```
# Этот сценарий вызывается из системного юнита как сценарий обработки
# подключения/отключения накопителей.
usage() {
echo "Использование: $0 {add|remove} device_name (например, sdb1)"
exit 1
}

if [[ $# -ne 2 ]]; then
usage
fi

ACTION=$1
DEVBASE=$2
DEVICE="/dev/${DEVBASE}"

# Проверяем, не примонтировано ли уже устройство
MOUNT_POINT=$(/bin/mount | /bin/grep ${DEVICE} | /usr/bin/awk
     '{ print $3 }')
```

```
do_mount() {
if [[ -n ${MOUNT_POINT} ]]; then
echo "Предупреждение: ${DEVICE} уже смонтировано в ${MOUNT_POINT}"
exit 1
fi

# Получаем информацию об устройстве : метка $ID_FS_LABEL, идентификатор
# $ID_FS_UUID, и тип файловой системы $ID_FS_TYPE
eval $(/sbin/blkid -o udev ${DEVICE})

# Создаем точку монтирования:
LABEL=${ID_FS_LABEL}
if [[ -z "${LABEL}" ]]; then
LABEL=${DEVBASE}
elif /bin/grep -q " /media/${LABEL} " /etc/mtab; then
# Если точка монтирования уже существует, то изменяем имя:
LABEL+="-${DEVBASE}"
fi
MOUNT_POINT="/media/${LABEL}"
echo "Точка монтирования: ${MOUNT_POINT}"
/bin/mkdir -p ${MOUNT_POINT}

# Глобальные параметры монтирования
OPTS="rw,relatime"

# Специфические параметры монтирования:
if [[ ${ID_FS_TYPE} == "vfat" ]]; then
OPTS+=",users,gid=100,umask=000,shortname=mixed,utf8=1,flush"
fi

if ! /bin/mount -o ${OPTS} ${DEVICE} ${MOUNT_POINT}; then
echo "Ошибка монтирования ${DEVICE} (статус = $?)"
/bin/rmdir ${MOUNT_POINT}
exit 1
fi

echo "**** Устройство ${DEVICE} смонтировано в ${MOUNT_POINT} ****"
}

do_unmount() {
if [[ -z ${MOUNT_POINT} ]]; then
echo "Предупреждение: ${DEVICE} не смонтировано"
else
/bin/umount -l ${DEVICE}
echo "**** Отмонтировано ${DEVICE}"
```

```
fi

# Удаление пустых каталогов
for f in /media/* ; do
if [[ -n $(/usr/bin/find "$f" -maxdepth 0 -type d -empty) ]]; then
if ! /bin/grep -q " $f " /etc/mtab; then
echo "**** Удаление точки монтирования $f"
/bin/rmdir "$f"
fi
fi
done
}

case "${ACTION}" in
add) do_mount ;;
remove) do_unmount ;;
*) usage ;;
esac
```

После создания файла сценария сделать его исполнимым, выполнив от имени администратора команду:

```
chmod +x <сценарий_обработки_события>
```

19.7. Порядок генерации правил udev для учета съемных накопителей

Съемный накопитель всегда является блочным устройством (`block`). Съемный накопитель всегда является устройством типа «диск» (`disk`) или типа «дисковый раздел» (`partition`), при этом правила мандатного управления доступом, применяемые для реализации учета съемных накопителей, применяются к дисковыми разделам.

Назначение мандатных атрибутов съемному накопителю выполняется при его подключении, при этом операции подключения выполняются раздельно для самого накопителя и для всех находящихся на этом накопителе дисковых разделов.

Правила `udev` применяются к устройствам при совпадении заданных в правиле параметров и параметров устройства. Все параметры подключенного устройства можно просмотреть, выполнив команду:

```
sudo udevadm info --query=property --name=/dev/<имя_устройства>
```

При генерации правил для блочных устройств не рекомендуется использовать параметры, относящиеся к подключению этих устройств в ОС (например, параметры DEVNAME, ID_BUS и др.), так как данные параметры:

- могут повторяться для разных устройств (присвоение имени sdX);
- могут зависеть от порядка подключения устройств (присвоение имени sdX);
- могут изменяться при изменении аппаратной конфигурации;
- могут отличаться на разных доменных компьютерах, имеющих разную аппаратную конфигурацию.

Параметры, применимые для идентификации устройств типа «диск» и типа «дисковый раздел» (параметры наследуются от устройства «диск»): ID_VENDOR, ID_VENDOR_ID, ID_VENDOR_ENC, ID_MODEL, ID_MODEL_ID, ID_MODEL_ENC, ID_SERIAL, ID_SERIAL_SHORT.

Дополнительно к устройствам типа «дисковый раздел» применимы параметры: ID_FS_LABEL, ID_FS_LABEL_ENC, ID_PART_ENTRY_NUMBER, ID_FS_TYPE, ID_FS_USAGE, ID_FS_UUID, ID_FS_UUID_ENC, ID_FS_VERSION, ID_PART_ENTRY_NUMBER.

Основным минимальным параметром идентификации съемного накопителя является его серийный номер (ID_SERIAL или ID_SERIAL_SHORT).

Для идентификации накопителей при использовании оборудования разных моделей и разных производителей можно использовать набор параметров «Производитель» — «Модель» — «Серийный номер» (например, ID_VENDOR, ID_MODEL, ID_SERIAL или ID_VENDOR_ID, ID_MODEL_ID, ID_SERIAL и т.д.).

С учетом того, что на одном устройстве может располагаться несколько дисковых разделов, для идентификации дисковых разделов в дополнение к параметрам идентификации накопителя можно использовать метку файловой системы (ID_FS_LABEL), универсальный идентификатор файловой системы UUID (ID_FS_UUID) и номер раздела на накопителе (ID_PART_ENTRY_NUMBER).

Пример

Правило идентификации дискового раздела по серийному номеру устройств и UUID

```
# отсекаются ненужные устройства - вероятность несовпадения серийного
# номера выше, правило сработает чаще

ENV{ID_SERIAL}!="SanDisk_Cruzer_Glide_XXXXXXXXXXXXX-0:0", GOTO="END"
ENV{ID_FS_UUID}!="0047-C44D", GOTO="END"

# отсекаются ненужные события
ACTION!="add",              GOTO="END"
```

```
ENV{SUBSYSTEM}!="block",   GOTO="END"
ENV{DEVTYPE}!="partition", GOTO="END"

# настройка правил Parsec
OWNER="user", GROUP="root", MODE="740", PDPL="0:0:0x0:0x0!:",
     AUDIT="o:0x0:0x0"
ENV{ID_FS_TYPE}=="?*", SYMLINK+="%k_$env{ID_FS_TYPE}",
RUN+="/bin/ln -f /dev/%k /dev/%k_$env{ID_FS_TYPE}"

LABEL="END"
```

19.8. Отладка правил

Включение вывода отладочных сообщений в файл `/var/log/syslog`:

```
udevadm control -l debug
```

Тестовая отработка правил `udev` без их загрузки:

```
udevadm test /dev/sdb1
```

Мониторинг событий `udev`:

```
udevadm monitor -k -u -p
```

Путь к устройству:

```
udevadm info -q path -n /dev/sdd1
```

Полная информация об устройстве:

```
udevadm info -a -p $(udevadm info -q path -n /dev/sdd1)
```

19.9. Регистрация устройств

Регистрация устройств в локальной базе учета устройств осуществляется с использованием модуля «Устройства и правила» графической утилиты `astra-systemsettings` («Параметры системы», см. электронную справку). Для вызова модуля можно использовать команду:

```
astra-systemsettings astra_kcm_devices_and_rules
```

Регистрация устройств в базе учета устройств FreeIPA осуществляется с использованием веб-интерфейса контроллера домена путем создания записей об этих устройствах и глобальных правил.

Устройства идентифицируются на основе атрибутов менеджера устройств `udev`. В большинстве случаев достаточно использовать серийный номер `ID_SERIAL`. В случае, когда использование для идентификации устройства серийного номера невозможно, необходимо выбрать один или несколько других атрибутов, обеспечивающих идентификацию устройства.

Для регистрации устройства (USB-накопитель, сканер, оптический носитель) и создания правил доступа к нему по классификационной метке для локальных пользователей необходимо выполнить следующие действия:

1) запустить от имени администратора через механизм `sudo` утилиту `astra-systemsettings` («Параметры системы», см. электронную справку) и выбрать на боковой панели «Устройства и правила»;

2) нажать кнопку **[Добавить устройство]**. Дождаться появления графического окна и подключить устройство одним из следующих способов в зависимости от типа устройства:

   а) подключить USB-накопитель к USB-порту компьютера;

   б) подключить кабель USB-сканера к USB-порту компьютера;

   в) вставить оптический носитель в устройство чтения CD/DVD-дисков.

3) в появившемся перечне выбрать устройство и открыть его «Свойства»;

4) в списке свойств устройства должны быть отмечены строки следующего вида:

   а) для USB-накопителей (отмечено по умолчанию):

```
ID_SERIAL       Значение
```

   б) для сканеров (отмечено по умолчанию):

```
ID_SERIAL       Значение
PRODUCT         Значение
```

   в) для оптических носителей (отмечено по умолчанию):

```
ID_SERIAL       Значение
```

   (позволяет идентифицировать устройства, на которых будет осуществляться работа с оптическими носителями)

```
ID_FS_LABEL     Значение
```

   (позволяет идентифицировать оптический носитель);

5) при необходимости выбрать другие свойства;

6) добавить устройство, нажав кнопку **[Да]**;

7) в поле «Наименование» указать наименование устройства;

8) во вкладке «Общие» необходимо выбрать пользователя, группу (владельца устройства) и задать права доступа для пользователя, группы и всех остальных;

9) указать классификационную метку, для этого во вкладке «МРД» выбрать иерархический уровень конфиденциальности и указать набор неиерархических категорий конфиденциальности;

10) назначить параметры регистрации событий, связанных с устройством. Для этого во вкладке «Аудит» необходимо выбрать событие и результат («Успех», «Отказ»), подлежащие регистрации;

11) во вкладке «Правила» назначить дополнительные наборы правил для устройства (из списка правил для менеджера устройств `udev`, созданных в разделе «Устройства и правила», см. 19.4);

12) применить изменения, нажав кнопку **[Применить]**.

Для регистрации устройства (USB-накопитель) и создания правил доступа к нему по классификационной метке для пользователей FreeIPA необходимо выполнить следующие действия:

1) запустить от имени администратора через механизм `sudo` утилиту управления политикой безопасности `astra-systemsettings` («Параметры системы», см. электронную справку) и выбрать на боковой панели «Устройства и правила»;

2) нажать кнопку **[Добавить устройство]**. Дождаться появления графического окна и подключить USB-накопитель к USB-порту компьютера;

3) в появившемся перечне выбрать устройство и открыть его «Свойства»;

4) в списке свойств устройства должны быть отмечены строки следующего вида:

```
ID_SERIAL      Значение
```

5) скопировать значение правила;

6) в веб-интерфейсе контроллера домена FreeIPA перейти «Политика — Политика PARSEC» и в выпадающем списке выбрать «Registerd device»;

7) задать имя регистрируемого носителя, права для пользователя и группы, в поле «Device attributes» вставить скопированное из `astra-systemsettings` правило, установить флаг «Device is ON» и сохранить правило, нажав кнопку **[Добавить]**;

8) прервать процедуру создания локального правила в `astra-systemsettings` без сохранения изменений;

9) для подготовки USB-носителя к работе в ненулевой сессии на одном уровне конфиденциальности:

а) создать сценарий `singlelevel.sh` со следующим текстом, задав соответствующие значения для параметров `USERNAME` и `DEVICE`:

```
#!/bin/bash
USERNAME="user"
```

```
DEVICE="/dev/sdc1"
mkfs.ext4 $DEVICE
mkdir -p /media/usb
mount $DEVICE /media/usb
#one level
pdpl-file 2:0:0:0 /media/usb/
chown -R ${USERNAME}:${USERNAME} /media/usb/
ls -la /media/usb/
pdp-ls -M /media/usb/
umount /media/usb
```

б) сделать сценарий исполняемым, выполнив команду:

```
sudo chmod +x singlelevel.sh
```

в) в веб-интерфейсе управления доменом FreeIPA («Политика — Политика PARSEC») создать глобальное правило использования требуемого устройства для соответствующего уровня;

10) для подготовки USB-носителя к работе в ненулевой сессии на нескольких уровнях конфиденциальности:

а) создать сценарий `multilevel.sh` со следующим текстом, задав соответствующие значения для параметров `USERNAME` и `DEVICE`:

```
#!/bin/bash
USERNAME="user"
DEVICE="/dev/sdc1"
mkfs.ext4 $DEVICE
mkdir -p /media/usb
mount $DEVICE /media/usb
#multilevel
pdpl-file 3:0:-1:ccnr /media/usb/
mkdir /media/usb/{0,1,2,3}
pdpl-file 0:0:0:0 /media/usb/0
pdpl-file 1:0:0:0 /media/usb/1
pdpl-file 2:0:0:0 /media/usb/2
pdpl-file 3:0:0:0 /media/usb/3
chown -R ${USERNAME}:${USERNAME} /media/usb/{0,1,2,3}
ls -la /media/usb/
pdp-ls -M /media/usb/
umount /media/usb
```

б) сделать сценарий исполняемым, выполнив команду:

```
sudo chmod +x multilevel.sh
```

в) в веб-интерфейсе управления доменом FreeIPA («Политика — Политика PARSEC») создать глобальные правила использования требуемого устройства для каждого из созданных уровней.

Для применения созданных/измененных правил доступа к устройству следует выполнить команду:

```
sudo udevadm trigger
```

Также правила будут применены после переподключения устройства или перезагрузки ОС.

После применения правил владелец устройства или пользователи из группы смогут монтировать данное устройство, при этом на точку монтирования устройства будет устанавливаться заданная классификационная метка (уровень и категории конфиденциальности).

ВНИМАНИЕ! В случае если включен мандатный контроль целостности, то действия по предоставлению пользователям доступа к устройствам должны осуществляться от имени администратора с высокой меткой целостности.

19.10. Блокировка USB-устройств в режиме «Мобильный»

Блокировка USB-устройств в режиме «Мобильный» осуществляется с помощью утилиты USBGuard. Утилита позволяет управлять блокировкой подключаемых устройств, создавая правила.

Настройка работы USBGuard осуществляется в конфигурационном файле `/etc/usbguard/usbguard-daemon.conf`.

Для управления блокировкой USB-устройств в графическом интерфейсе реализован модуль KCM. Доступ к модулю ограничивается политикой Polkit.

Администратор при подключении USB-устройства настраивает доступ к нему, создавая правила. Если при включенной службе блокировки USB-устройств будет подключено USB-устройство, для которого отсутствует правило, то данное устройство будет заблокировано. Порядок использования модуля KCM для блокировки USB-устройств описан в электронной справке («Документация — Графический интерфейс — Режим «Мобильный»).

20. ПОДДЕРЖКА СРЕДСТВ ДВУХФАКТОРНОЙ АУТЕНТИФИКАЦИИ

Повышение надежности аутентификации возможно достичь путем использования многофакторной аутентификации, предполагающей применение нескольких типов аутентификационных факторов

К факторам, которые могут быть использованы, относятся:

- ввод пароля или PIN-кода;
- ввод одноразовых паролей (скрэтч-карты);
- предоставление физического устройства или носителя, содержащего аутентификационную информацию (смарт-карта, USB-токен и т. п.);
- предоставление биометрической информации (отпечатки пальцев, изображение сетчатки глаза и т. п.).

На практике в большинстве случаев используется двухфакторная аутентификация на основе ввода пароля с одновременным предоставлением пользователем физического устройства (носителя), содержащего дополнительную аутентификационную информацию. Дополнительной аутентификационной информацией в этом случае обычно является размещенный на устройстве сертификат пользователя.

Для обеспечения двухфакторной аутентификации с помощью внешнего носителя используются следующие средства и технологии:

- PKCS (Public Key Cryptography Standard) — группа стандартов защитного преобразования с открытым ключом, в частности, стандарты PKCS-11, PKCS-12, PKCS-15, относящиеся к работе с токенами;
- X.509 — стандарт, определяющий форматы данных и процедуры распределения открытых ключей с помощью сертификатов с цифровыми подписями, которые предоставляются центрами аутентификации;
- OpenSC — набор программных утилит и библиотек для работы с носителями аутентификационной информации пользователя (смарт-карты, USB-токены), содержащие функции аутентификации, преобразования и цифровой подписи. Поддерживает стандарты PKCS-11, PKCS-15;
- OpenCT — набор драйверов устройств для работы с носителями аутентификационной информации (устаревший);
- OpenSSL — программное средство для работы с протоколом SSL/TLS. Позволяет создавать ключи RSA, DH, DSA и сертификаты X.509, подписывать их, формировать файлы сертификатов CSR и CRT. Также имеется возможность тестирования SSL/TLS соединений. Поддерживает механизм динамически подключаемых библиотек алгоритмов защитного преобразования данных, т.е. механизм подключения внешних модулей, содержащих дополнительные алгоритмы. С использованием ука-

занного механизма обеспечивает работу с алгоритмами защитного преобразования данных в соответствии с требованиями ГОСТ (пакет библиотеки алгоритмов ГОСТ `libgost-astra`);

- PC/SC — набор спецификаций для доступа к смарт-картам;

- PKINIT (Public Key Cryptography for Initial Authentication in Kerberos) — стандарт использования защитного преобразования с открытым ключом в качестве фактора аутентификации в протоколе аутентификации Kerberos (см. 8.1.4).

Двухфакторная аутентификация может применяться как в случае использования локальной аутентификации, так и в случае использования ЕПП.

20.1. Аутентификация с открытым ключом (инфраструктура открытых ключей)

Аутентификация на основе ключей использует ключевую пару: один ключ «открытый» (публичный), который доступен каждому, и второй ключ «закрытый» (секретный), который доступен только владельцу. В процессе аутентификации используются алгоритмы с открытым ключом для проверки подлинности пользователя. При этом закрытый ключ находится непосредственно у пользователя, а открытый ключ по защищенным каналам связи передается в те системы, которые должны с его помощью проверять подлинность пользователя.

В качестве электронного представления ключей используются цифровые сертификаты. Сертификат является подтверждением принадлежности открытого ключа. Цифровой сертификат устанавливает и гарантирует соответствие между открытым ключом и его владельцем. Сертификаты выдаются специальными уполномоченными организациями — центрами аутентификации. Сертификаты могут быть использованы не только для аутентификации, но и для предоставления избирательных прав доступа, в том числе и права подписи других сертификатов.

В рамках изолированной информационной системы средством выработки и подписывания цифровых сертификатов могут быть использованы различные программные средства, например, `openssl`. В этом случае такое средство может выступать в роли локального центра аутентификации для создания ключевых пар и сертификатов клиентов и серверов системы.

При доступе к ресурсам информационных систем часто используются механизмы защитного преобразования, основанные на ассиметричных алгоритмах и сертификатах открытого ключа. Применение указанных механизмов в информационных системах обеспечивается инфраструкторой открытых ключей PKI, которая включает в себя набор аппаратных и программных средств, политик и процедур создания, управления, распространения, использования и отзыва цифровых сертификатов.

В основе PKI лежит использование защитного преобразования с открытым ключом и несколько основных принципов:

- закрытый ключ известен только его владельцу;
- центр аутентификации создает сертификат открытого ключа, таким образом подтверждая этот ключ;
- никто не доверяет друг другу, но все доверяют центру аутентификации;
- центр аутентификации подтверждает или опровергает принадлежность открытого ключа заданному лицу, которое владеет соответствующим закрытым ключом.

20.2. Средства поддержки двухфакторной аутентификации

20.2.1. Общие сведения

В ОС поддерживается механизм двухфакторной аутентификации пользователей с использованием токенов.

Для аутентификации пользователей используется модуль `pam-csp`, реализованный на основе стандартного PAM-модуля `libpam-csp`.

PAM-модуль `libpam-csp` обрабатывает два события:

- аутентификация пользователя;
- смена пароля пользователя.

Для доступа к токенам используется стандартная библиотека `opensc-pkcs11`, позволяющая модулю `libpam-csp` работать с любыми токенами различных производителей, поддерживающими эту библиотеку.

Контроль пользовательской сессии осуществляется с использованием службы `csp-monitor`. Для взаимодействия службы с токенами используется библиотека `opensc-pkcs11`.

Служба `csp-monitor` принимает от `pam_csp` по шине DBus сообщения о входе и выходе пользователя с использованием токена и поддерживает список текущих пользовательских сессий с информацией об использованных для входа токенах.

Служба `csp-monitor` осуществляет мониторинг подключений и отключений USB-устройств и если какой-либо токен из числа участвующих в аутентификации пользователя был извлечен, то блокирует все сессии данного пользователя. Для разблокировки сессии пользователь должен подключить токен и ввести PIN-код.

Служба `csp-monitor` управляется как юнит `systemd`. Для просмотра статуса службы выполнить команду:

```
systemctl status csp-monitor
```

ВНИМАНИЕ! При использовании решения `pam_csp` совместно с FreeIPA для параметра доменной политики паролей «минимальный срок действия пароля» должно быть задано значение 0.

20.2.2. Настройка клиентской машины

Для установки модуля `libpam-csp` выполнить установку соответствующего пакета от имени администратора командой:

```
apt install libpam-csp
```

Далее необходимо задать команду принудительной смены пароля. Для локальных пользователей на компьютере пользователя выполнить команду от имени администратора:

```
passwd --expire <имя_пользователя>
```

Для доменных пользователей необходимо использовать соответствующие инструменты администрирования домена.

При установке пакета `libpam-csp` автоматически будет установлен пакет для службы `csp-monitor`.

Во время установки пакета модуль `pam_csp` регистрируется первым в цепочках PAM-модулей в двух PAM-профилях:

```
/etc/pam.d/common-auth
/etc/pam.d/common-password
```

20.2.3. Инициализация токена

Процесс инициализации токена одинаков для локальных и доменных пользователей.

До передачи токена пользователю выполняется его подготовка на компьютере администратора, ответственного за подготовку.

Для выполнения подготовки токена на компьютере должны быть установлены пакеты:

- `opensc-pkcs11` версии не ниже 0.19.0-2;
- `ifd-rutokens` версии не ниже 1.0.4 (для Rutoken S и Rutoken ECP);
- пакеты других интерфейсных модулей, необходимые для используемой модели токена.

Для установки пакета `opensc-pkcs11` выполнить от имени администратора команду:

```
sudo apt install opensc-pkcs11
```

Установка интерфейсных модулей выполняется в соответствии с инструкциями производителей соответствующих токенов.

Процедура инициализации зависит от используемой модели токена.

Для инициализации токена Rutoken S выполнить последовательно следующие команды:

```
pkcs15-init --erase-card
pkcs15-init --create-pkcs15 --so-pin "87654321" --so-puk "" --pin "12345678"
pkcs15-init --store-pin --label "User PIN" --auth-id 02 --pin "12345678" \
     --puk ""
```

Для инициализации токена Rutoken ECP выполнить последовательно следующие команды:

```
pkcs15-init --erase-card -p rutoken_ecp
pkcs15-init --create-pkcs15 --so-pin "87654321" --so-puk ""
pkcs15-init --store-pin --label "User PIN" --auth-id 02 --pin "12345678" \
     --puk "" --so-pin "87654321" --finalize
```

Проверить, что токен успешно инициализирован, можно с помощью команды:

```
pkcs15-tool -D
```

20.2.4. Использование токена

При первичном использовании токена для входа в свою сессию пользователь должен подключить токен и в соответствующих полях ввести свои логин и пароль. При появлении окна с дополнительным приглашением:

```
Supply token PIN:
```

ввести PIN токена (текущий PIN токена пользователю сообщает администратор).

Далее пользователю будет предложено сменить PIN:

```
Supply new token PIN:
Retype new token PIN:
```

При этом можно указать новый PIN (рекомендуется), введя его два раза, или два раза нажать клавишу **<Enter>**, чтобы оставить текущий PIN (не рекомендуется).

После первичного ввода PIN произойдет генерация нового случайного пароля, его назначение учетной записи пользователя и будет выполнен вход в систему. В дальнейшем в токене будет храниться 16-символьный пароль, недоступный без знания PIN.

При последующих входах в систему пользователю нужно подключить токен и далее в соответствующих полях ввести логин и PIN токена.

Пример

Диалог при терминальном входе

```
login: user
Supply token PIN:
```

При необходимости сменить пароль пользователь должен подключить токен, войти в систему и затем:

1) при первичном входе — выполнить в командной строке команду `passwd`. При этом будут запрошены текущие пароль и PIN:

```
passwd
Введите ПИН-код :
Введите текущий пароль :
Введите новый ПИН-код :
Введите новый ПИН-код еще раз :
```

2) при последующих входах — выполнить в командной строке команду `passwd`. При этом будет запрошен текущий PIN:

```
passwd
Введите ПИН-код :
Введите новый ПИН-код :
Введите новый ПИН-код еще раз :
```

Для локального пользователя администратор может подготовить токен со сгенерированным паролем заранее. Для этого следует подключить токен и выполнить команду:

```
passwd <имя_пользователя>
Введите ПИН-код :
Введите новый ПИН-код :
Введите новый ПИН-код еще раз :
ПИН-код успешно изменен.
passwd: пароль успешно изменен
```

20.2.5. Разблокировка сессии с ненулевой меткой конфиденциальности с помощью PIN-кода

Токен возможно использовать для входа в сессию с ненулевым уровнем конфиденциальности. При этом для того чтобы функция разблокировки сессии по PIN-коду работала корректно, необходимо произвести следующие настройки:

1) присвоить сокету `/var/run/pcscd/pcscd.comm` привилегию `PARSEC_CAP_PRIV_SOCK`, добавив в раздел `[Socket]` файла `/lib/systemd/system/pcscd.socket` строку:

```
CapabilitiesParsec=PARSEC_CAP_PRIV_SOCK
```

2) перезапустить службу `pcscd`:

```
sudo systemctl daemon-reload
sudo systemctl stop pcscd.service
sudo systemctl stop pcscd.socket
sudo systemctl start pcscd.service
```

3) обеспечить корректную работу модуля `pam_p11`, входящего в состав `libpam-p11`. Для этого в каталоге `/home` каждого доступного пользователю уровня конфиденциальности должен находиться файл `.euid/authorized_certificates`. Данный файл можно после настройки модуля `pam_p11` скопировать из каталога `/home` пользователя нулевого уровня конфиденциальности в каталоги `/home` других уровней конфиденциальности.

20.3. Управление сертификатами

Для обеспечения аутентификации с открытым ключом в информационной системе необходимо иметь набор ключевых пар и сертификатов ресурсов сети (серверов или служб) и ее клиентов (пользователей). Формирование и подписывание сертификатов выполняет-

ся с помощью центра аутентификации информационной системы. Процедура получения необходимого набора сертификатов заключается в следующем:

1) формируются ключи и корневой сертификат центра аутентификации;
2) для каждого сервера или клиента генерируется ключевая пара;
3) на основе полученной ключевой пары формируется заявка (запрос) на сертификат;
4) с помощью центра аутентификации по заявке выписывается сертификат;
5) полученная ключевая пара и сертификат сохраняются в соответствующие места системы.

Генерация ключевых пар и работа с сертификатами осуществляется согласно инструкции производителя соответствующего токена.

20.4. Настройка доменного входа (ЕПП)

При использовании ЕПП для аутентификации пользователей применяется доверенная аутентификация Kerberos (см. 8.1.4). По умолчанию аутентификация производится по паролю пользователя. В то же время существует стандарт использования защитного преобразования с открытым ключом в качестве фактора аутентификации в протоколе аутентификации Kerberos PKINIT (Public Key Cryptography for Initial Authentication in Kerberos). Это позволяет применять сертификаты и, следовательно, устройства PKCS-11 для аутентификации по Kerberos.

Для используемого варианта Kerberos (MIT Kerberos V5) возможности PKINIT реализуются пакетом расширения `krb5-pkinit`. При этом для проведения аутентификации используется подгружаемый модуль аутентификации `libpam-krb5`.

ВНИМАНИЕ! Перед настройкой доменного входа с помощью сертификатов с устройств PKCS-11 должны быть выполнены следующие условия:

1) установлена и соответствующим образом настроена служба домена;
2) настроен домен ЕПП и созданы необходимые пользователи;
3) на компьютеры домена установлен пакет расширения `krb5-pkinit`;
4) получен или создан корневой сертификат CA.

21. СООБЩЕНИЯ АДМИНИСТРАТОРУ И ВЫЯВЛЕНИЕ ОШИБОК

21.1. Диагностические сообщения

При возникновении проблем в процессе функционирования ОС появляются диагностические сообщения трех типов: информационные, предупреждающие и сообщения об ошибках (примеры приведены в таблицах 62–64, соответственно). Администратор должен проанализировать диагностические сообщения и принять меры по устранению появившихся проблем.

Таблица 62 – Информационные сообщения

Сообщение ОС	Описание	Файл
`Setting hostname to <>`	Установка имени хоста <>	`hostname`
`Setting domainname to <>`	Установка имени домена <>	`hostname`
`Statistics dump initiated`	Вывод статистики запущен	`named`
`Query logging is now on`	Регистрация очередей включена	`named`
`Query logging is now off`	Регистрация очередей выключена	`named`
`Unknown host`	Неизвестный хост	`dnsquery`
`Non reloadable zone`	Неперезагружаемая зона	`named`
`Reconfig initiated`	Переконфигурирование запущено	`named`
`Zone not found`	Зона не найдена	`named`

Таблица 63 – Предупреждающие сообщения

Сообщение ОС	Описание	Действия по устранению проблемы	Файл
`<>: You can't change the DNS domain name with this command`	Неверное использование команды	Использовать соответствующую команду	`hostname`
`Could not find any active network interfaces`	Активные сетевые интерфейсы не найдены	Активировать сетевой интерфейс	`sendmail`
`You must be root to change the host name`	Недостаточно прав для изменения имени хоста	Обратиться к администратору	`dnsdomainname`

Таблица 64 – Сообщения об ошибках

Сообщение ОС	Описание	Действия по устранению проблемы	Файл
`Unknown server error`	Неизвестная ошибка сервера	Изменить права доступа	`dnsquery`
`Resolver internal error`	Внутренняя ошибка резольвера	Изменить права доступа	`dnsquery`

Окончание таблицы 64

Сообщение ОС	Описание	Действия по устранению проблемы	Файл
`Superblock last mount time (значение времени) is in the future`	Неверная установка времени	См. 21.3	См. 21.3

21.2. Выявление ошибок

В состав ОС входит инструмент `sosreport`, предназначенный для сбора информации о конфигурации системы и диагностических данных о работе ОС и ее компонентов. Инструмент включает модули для сбора информации о работе отдельных подсистем и программ из состава ОС.

На основе собранных данных создается диагностический архив с отчетом, который может храниться локально, централизованно или отправляться техническим специалистам. Дополнительно возможно создавать XML/HTML-отчеты.

Перечень основных параметров, используемых с инструментом `sosreport`, приведен в таблице 65.

Таблица 65

Параметр	Описание
`-l`	Вывести список доступных модулей и их параметры. Модули, которые не могут использоваться с текущей конфигурацией, выводятся отдельно
`-n <имя_модуля>`	Отключить указанный модуль. Отключение нескольких модулей выполняется повторением параметра или заданием списка модулей через запятую
`-e <имя_модуля>`	Включить указанный модуль. Включение нескольких модулей выполняется повторением параметра или заданием списка модулей через запятую
`-o <имя_модуля>`	Включить только указанный модуль (неуказанные модули будут автоматически отключены). Включение нескольких модулей выполняется повторением параметра или заданием списка модулей через запятую
`-k <имя_модуля>.<параметр_модуля> [=<значение>]`	Задать параметры модуля. Включает указанный параметр модуля, может также задавать значение параметра модуля
`-a`	Установить для всех логических параметров всех включенных модулей значение `True`
`-v`	Увеличить детализацию протоколирования. Может выполняться несколько раз для добавления дополнительных сообщений

Продолжение таблицы 65

Параметр	Описание
`--no-postproc`	Отключить постобработку собранных данных для всех модулей. В архиве с собранными данными не будет замаскирована/очищена конфиденциальная информация. Такие данные, как пароли, SSH-ключи, сертификаты будут сохранены в виде простого текста. Чтобы отключить постобработку для определенного модуля, использовать с параметром `-k` параметр `postproc` модуля, например `-k logs.postproc=off`
`-s <корневая_файловая_система>`	Указать другую корневую файловую систему. Возможно использовать для создания отчета работы контейнера или образа
`-c {auto/always/never}`	Установить режим использования `chroot`. Когда используется `-s`, команды по умолчанию выполняются с заданной файловой системой (если только они не отключены определенным модулем). Параметр `-c` переопределяет использование заданной корневой файловой системы: - значение `always` — всегда использовать корневую файловую систему, заданную параметром `-s`; - `never` — никогда не использовать корневую файловую систему, заданную параметром `-s` (команды всегда будут выполняться в пространстве хоста)
`--tmp-dir <путь>`	Задать временный каталог для копирования данных и архива отчета
`--list-profiles`	Вывести список доступных профилей и включенных в них модулей
`-p <имя_профиля>`	Выполнить модули, включенные в указанный профиль. Несколько профилей могут быть заданы через запятую, при этом будут выполнены модули всех указанных профилей
`--log-size`	Установить ограничение на размер (в МиБ) набора журналов. Ограничение применяется отдельно для каждого набора журналов, собранных любым модулем
`--all-logs`	Собирать данные всех возможных журналов регистрации событий, включая из незаданных областей и игнорируя ограничения по размеру. В данном случае может быть значительно увеличен размер отчетов
`-z <метод_сжатия>`	Задать метод сжатия отчета
`--encrypt-pass <пароль>`	Аналогично `--encrypt-key`, но защита архива выполняется установкой пароля
`--batch`	Создать архив отчета без интерактивных запросов пользователю

Окончание таблицы 65

Параметр	Описание
`--case-id <идентификатор_архива>`	Задать идентификатор архива. Может содержать цифры, латинские буквы, запятые и точки

Более подробное описание инструмента доступно в `man sosreport`.

Для использования инструмента `sosreport` в графическом режиме доступна утилита `fly-sosreport`. Описание утилиты приведено в электронной справке.

ВНИМАНИЕ! В настоящее время инструмент `sosreport` отмечен как устаревший и в будущем будет исключен из состава. Вместо него следует использовать новый инструмент, вызываемый командой `sos report`. Новый инструмент работает идентично старому и использует те же параметры.

21.3. Циклическая перезагрузка компьютера по причине неверной установки времени

При возникновении сбоя, связанного с циклической перезагрузкой компьютера, необходимо во время загрузки ОС при появлении на экране заставки с мерцающей надписью «Astra Linux Special Edition» нажать клавишу **<Esc>**. Если среди отобразившихся сообщений есть сообщение вида:

```
/dev/sda1: Superblock last mount time (Wed Feb 15 12:41:05 2017,
            now = Mon Feb 15 12:45:37 2016) is in the future.
```

то сбой связан с неверной установкой времени.

Для устранения сбоя необходимо войти в меню настройки BIOS (UEFI) и проверить выставленное системное время. Если системное время отстает от реального, то, возможно, это связано с отказом элемента питания системной платы. В этом случае необходимо заменить элемент питания на системной плате в соответствии с указаниями инструкции к техническому средству и установить корректное системное время.

Если системное время в меню настроек BIOS (UEFI) установлено верно, но циклическая перезагрузка продолжается, то сбой может быть связан с неверным переводом времени на будущую дату и обратно. Данный сбой происходит если установить системное время на будущую дату, затем загрузить ОС и установить верное текущее время или сразу установить системное время на прошедшую дату. Для устранения данного сбоя необходимо:

1) в меню настроек BIOS (UEFI) установить системное время на будущую дату, при этом дата должна быть позже даты, указанной в сообщении об ошибке при загрузке;

2) загрузить ОС;

3) создать файл `/etc/ef2fsck.conf` с содержимым:

```
[options]
broken_system_clock = true
```

4) создать файл `/etc/initramfs-tools/hooks/e2fsck-conf.sh` с содержимым:

```
#!/bin/sh

PREREQ=""
prereqs()
{
   echo "$PREREQ"
}

case $1 in
prereqs)
   prereqs
   exit 0
   ;;
esac

. /usr/share/initramfs-tools/hook-functions
CONFFILE=/etc/e2fsck.conf
CONFDIR=`dirname "$CONFFILE"`
if [ -f "$CONFFILE" ]
then
   mkdir -p ${DESTDIR}${CONFDIR}
   cp $CONFFILE ${DESTDIR}${CONFDIR}
fi
```

5) в терминале выполнить команду:

```
sudo update-initramfs -u
```

6) перезагрузить ОС и установить текущее время в качестве системного.

ПЕРЕЧЕНЬ ТЕРМИНОВ

Закрытый ключ — сохраняемый в тайне ключ из ключевой пары, принадлежащий владельцу и не подлежащий распространению.

Ключ — параметр в виде последовательности псевдослучайных чисел (не предназначен для защиты информации в контексте использования для целей, установленных в документации изделия; к ключам не предъявляются требования по источнику псевдослучайных чисел, криптографической стойкости, времени действия и т. п.).

Ключевая пара — упорядоченная пара математически однозначно связанных ключей, определяющих взаимосвязанные защитные преобразования.

Открытый ключ — ключ из ключевой пары, который может быть сделан общедоступным.

Сертификат открытого ключа — артефакт, содержащий открытый ключ, информацию о владельце ключа и подтверждающий принадлежность открытого ключа владельцу, защищенный с применением закрытого ключа.

Хеш — строка бит, являющаяся выходным результатом функции хеширования.

Центр аутентификации — программный компонент, реализующий возможность подтверждения подлинности ключей с помощью сертификатов.

Цифровая подпись — результат преобразования хеша для его защиты от несанкционированного доступа с использованием закрытого ключа (не предназначена для криптографической защиты информации).

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

БД — база данных
ВМ — виртуальная машина
ЕПП — единое пространство пользователей
КСЗ — комплекс средств защиты
ЛВС — локальная вычислительная сеть
МКЦ — мандатный контроль целостности
НСД — несанкционированный доступ
ОС — операционная система специального назначения «Astra Linux Special Edition»
ПО — программное обеспечение
СЗИ — средства защиты информации
СЗФС — сетевая защищенная файловая система
СУБД — система управления базами данных
ФС — файловая система
ЦА — центр аутентификации

AD — Active Directory (служба каталогов)
ACL — Access Control List (список контроля доступа)
API — Application Programming Interface (программный интерфейс приложения)
ARP — Address Resolution Protocol (протокол разрешения адресов)
BIOS — Basic Input-Output system (базовая система ввода-вывода)
BIND — Berkeley Internet Name Domain (пакет программного обеспечения для поддержки DNS, разработанный в Калифорнийском университете, г. Беркли)
BOOTP — Bootstrap Protocol (простой протокол динамической конфигурации хоста)
BSD — Berkeley Software Distribution (программное изделие Калифорнийского университета)
CA — Certification Authority (центр аутентификации)
CephFS — Ceph File System (файловая система Ceph)
CIFS — Common Internet File System (общий протокол доступа к файлам Интернет)
DC — Domain Controller (контроллер домена)
DHCP — Dynamic Host Configuration Protocol (протокол динамической конфигурации хоста)
DIB — Directory Information Base (информационная база каталога)
DIT — Directory Information Tree (информационное дерево каталога)
DN — Distinguished Name (уникальное имя)
DNS — Domain Name System (система доменных имен)
FTP — File Transfer Protocol (протокол передачи файлов)
FQDN — Fully Qualified Domain Name (полностью определенное имя домена)
GID — Group Identifier (идентификатор группы)
HTTP — HyperText Transfer Protocol (протокол передачи гипертекста)
IDE — Integrated Drive Electronics (встроенный интерфейс накопителей)

IMAP — Internet Message Access Protocol (протокол доступа к сообщениям в сети Интернет)

IP — Internet Protocol (межсетевой протокол)

IPA — Identity, Policy, and Audit (система по управлению идентификацией пользователей, задания политик доступа и аудита для сетей на базе Linux и Unix)

IPC — InterProcess Communication (межпроцессное взаимодействие)

KDC — Key Distribution Center (центр распределения ключей)

KRA — Key Recovery Authority (служба восстановления ключей)

KVM — Kernel-based Virtual Machine (программное решение, обеспечивающее виртуализацию в среде Linux на платформе, которая поддерживает аппаратную виртуализацию на базе Intel VT (Virtualization Technology) либо AMD SVM (Secure Virtual Machine))

LDAP — Lightweight Directory Access Protocol (легковесный протокол доступа к службам каталогов)

LPR — Line Printer Remote (удаленный линейный принтер)

LVM — Logical Volume Manager (менеджер логических томов)

MAC — Mandatory Access Control (мандатное управление доступом)

MDA — Mail Delivery Agent (агент доставки электронной почты)

MDS — Metadata Server (сервер метаданных)

MIT — Massachusetts Institute of Technology (Массачусетский Технологический Институт)

MON — Monitor (монитор)

MTA — Mail Transfer Agent (агент пересылки сообщений)

MTU — Maximum Transfer Unit (максимальная единица передачи)

MUA — Mail User Agent (клиент электронной почты)

NAT — Network Address Translation (преобразование сетевых адресов)

NFS — Network File System (сетевая файловая система)

NIS — Network Information Service (сетевая информационная служба)

NSS — Name Service Switch (диспетчер службы имен)

NTP — Network Time Protocol (протокол сетевого времени)

OCI — Open Container Initiative (проект, который разрабатывает открытые стандарты для сред контейнеризации)

OSD — Object Storage Device (устройство хранения объектов)

PAM — Pluggable Authentication Modules (подключаемые модули аутентификации)

PID — Process Identifier (идентификатор процесса)

PKI — Public Key Infrastructure (инфраструктура открытых ключей)

PTP — Precision Time Protocol (протокол точного времени)

POP3 — Post Office Protocol Version 3 (почтовый протокол, версия 3)

QEMU — Quick Emulator (средства эмуляции аппаратного обеспечения)

RADOS — Reliable Autonomic Distributed Object Store (безотказное автономное распределенное хранилище объектов)

RBD — RADOS block device (блочное устройство)

RFC — Request For Comments (общее название технических стандартов сети Интернет)

RPC — Remote Procedure Call (удаленный вызов процедур)
RTS — Real Time Clock (время, установленное в аппаратных часах компьютера)
SASL — Simple Authentication and Security Layer (простая аутентификация и слой безопасности)
SATA — Serial ATA (последовательный интерфейс обмена данными с накопителями информации, является развитием интерфейса IDE)
SCSI — Small Computer System Interface (системный интерфейс малых компьютеров)
SMB — Server Message Block (блок сообщений сервера)
SPICE — Simple Protocol for Independent Computing Environments (простой протокол для независимой вычислительной среды)
SQL — Structured Query Language (язык структурированных запросов)
SSH — Secure Shell Protocol (протокол защищенной передачи информации)
SSL — Secure Sockets Layer (протокол защищенных сокетов)
SSSD — System Security Services Daemon (системная служба, управляющая доступом к удаленным каталогам и механизмам аутентификации)
TCP — Transmission Control Protocol (протокол управления передачей данных)
TLS — Transport Layer Security (протокол защиты транспортного уровня)
TTL — Time To Live (время жизни IP-пакета)
UDP — User Datagram Protocol (протокол пользовательских дейтаграмм)
UEFI — Unified Extensible Firmware Interface (унифицированный расширяемый микропрограммный интерфейс)
UID — User Identifier (идентификатор пользователя)
URI — Uniform Resource Identifier (унифицированный идентификатор ресурса)
UTC — Universal Time Coordinated (универсальное скоординированное время)
UUID — Universally Unique IDentifier (всемирно уникальный идентификатор)
VFS — Virtual File System (виртуальная файловая система)
VIP — Virtual IP-adress (виртуальный IP-адрес)
VNC — Virtual Network Computing (система удаленного доступа к рабочему столу компьютера)
VPN — Virtual Private Network (виртуальная частная сеть)
VRRP — Virtual Redundancy Routing Protocol (сетевой протокол виртуального резервирования маршрутизаторов, предназначенный для увеличения доступности)
XCA — X window system Certification Authority (графический инструмент создания и управления центром аутентификации)

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Входящий номер сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					
1		Все				РУСБ.14.02-16 от 17.02.16			
2		Все				РУСБ.14.03-17 от 05.04.17			
3		Все				РУСБ.14.02-18 от 11.04.18			
4		Все				РУСБ.14.03-18 от 19.04.18			
5		Все				РУСБ.08-20 от 18.02.20			
6		Все				РУСБ.20-20 от 23.07.20			
7		Все				РУСБ.06-21 от 01.04.21			
8		Все				РУСБ.12-21 от 30.07.21			
9		Все				РУСБ.10-22 от 03.11.22			
10		Все				РУСБ.03-23 от 26.04.23			
11		Все				РУСБ.14-23 от 26.12.23			
12		Все				РУСБ.02-24 от 25.03.24			
13		Все				РУСБ.04-24 от 01.07.24			