ОПЕРАЦИОННАЯ СИСТЕМА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ «ASTRA LINUX SPECIAL EDITION»

РУСБ.10015-01

Руководство по КСЗ. Часть 1

Оперативное обновление 1.7.5

Бюллетень № 2023-1023SE17

Листов 36

2023

АННОТАЦИЯ

В настоящем руководстве приводятся кумулятивные изменения в документ РУСБ.10015-01 97 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1» из комплектности изделия РУСБ.10015-01 «Операционная система специального назначения «Astra Linux Special Edition» (далее по тексту — ОС), которые необходимо учитывать при настройке и эксплуатации ОС с установленным оперативным обновлением согласно бюллетеню № 2023-1023SE17.

Руководство предназначено для администраторов безопасности.

СОДЕРЖАНИЕ

1. Общие сведения 4
2. Перечень изменений 5
2.1. Подраздел «4.3.1. Метка целостности» 5
2.2. Подраздел «4.7. PARSEC-привилегии» 7
2.3. Пункт «4.15.7. pdpl-group» 7
2.4. Пункт «4.17.1. Порядок применения мандатных правил управления доступом» . . 8
2.5. Подраздел «5.2. Мандатное управление доступом к виртуальной машине» 8
2.6. Пункт «5.6.4. Контроль целостности образов ВМ» 8
2.7. Пункт «5.14.1. Пример организации распределенного хранилища» 9
2.8. Пункт «5.14.2. Пример миграции виртуальных машин» 24
2.9. Подраздел «6.1. Правила регистрации событий» 27
2.10. Пункт «6.4.3. setfaud» 30
2.11. Пункт «6.4.7. Дополнительные параметры регистрации событий» 32
2.12. Подраздел «12.2. Настройка печати документа с ненулевой классификационной меткой» 33
2.13. Пункт «12.3.1. Файл описания переменных маркировки» 33
2.14. Пункт «16.4. Графический киоск в режиме «Мобильный»» 34
2.15. Пункт «16.5.34. Выключение и включение аудита файлов и процессов» 35
2.16. Пункт «16.5.35. Выключение и включение сетевого аудита» 36

1. ОБЩИЕ СВЕДЕНИЯ

В настоящем руководстве приведены кумулятивные изменения в документ РУСБ.10015-01 97 01-1: измененные разделы, подразделы и пункты документа, а также добавленные разделы, подразделы и пункты.

При администрировании комплекса средств защиты ОС с установленным оперативным обновлением согласно бюллетеню № 2023-1023SE17 рекомендуется руководствоваться документом РУСБ.10015-01 97 01-1 совместно с настоящим руководством.

2. ПЕРЕЧЕНЬ ИЗМЕНЕНИЙ

2.1. Подраздел «4.3.1. Метка целостности»

Пункт 4.3.1 изложить в редакции:

4.3.1. Метка целостности

Субъектам и сущностям задаются метки целостности — совокупность (декартово произведение) неиерархических уровней (категорий) целостности и иерархических (линейных) уровней целостности.

Метка целостности сущности отражает степень уверенности в целостности содержащейся в ней информации. Метка целостности субъекта соответствует его полномочиям по доступу к сущности в зависимости от ее метки целостности, а также отражает степень уверенности в корректности его функциональности.

Процесс при его непосредственном запуске наследует метку целостности процесса-родителя.

Субъект с определенной меткой целостности может получить доступ на запись к сущности, если его метка целостности не ниже метки целостности сущности.

В стандартной реализации иерархический (линейный) уровень целостности в ОС зарезервирован и не поддерживается его использование.

Неиерархический уровень целостности технически реализован как 32-битная маска, беззнаковая величина (`uint32_t`). В пользовательских интерфейсах представляется десятичным или шестнадцатеричным числом или наименованием.

В ОС по умолчанию выделены нулевой, четыре ненулевых и несравнимых между собой (далее — изолированных) неиерархических уровня целостности, а также максимальный неиерархический уровень целостности, который не меньше всех остальных в системе.

При установке ОС по умолчанию предлагается максимальным неиерархический уровень целостности `max_ilev`, равный 63 (битовая маска 00111111), а минимальный всегда 0.

Дополнительно для обозначения максимального уровня целостности в установленной ОС зарезервировано специальное наименование уровня целостности `Высокий` (`High`), для обозначения нулевого уровня целостности зарезервировано специальное наименование `Низкий` (`Low`).

Метка целостности может быть назначена пользователю или группе пользователей. Метка целостности пользователя указывается в `/etc/parsec/micdb` (для локальной учетной записи) или в базе FreeIPA (для доменной учетной записи). Метка целостности для группы пользователей (локальной или доменной) указывается в `/etc/parsec/micgrdb`.

ВНИМАНИЕ! Если пользователю назначена персональная метка целостности, то она имеет приоритет над меткой целостности группы, в которую входит данный пользователь (т.е. метка целостности группы не применяется).

Если пользователю не назначена персональная метка целостности, то при входе в сессию ему присваивается метка целостности группы, в которую он входит.

На одну группу может быть назначена только одна метка целостности, при этом пользователь может состоять в нескольких группах. В таком случае при входе пользователя в сессию вычисляется его эффективная метка целостности на основе меток целостности всех групп, в которых он состоит. Эффективная метка целостности представляет собой битовую маску, в которой каждый бит устанавливается равным 1, если он равен 1 в метке целостности хотя бы одной из групп (логическое «ИЛИ»). Если в метках целостности всех групп бит равен 0, то в эффективной метке он также устанавливается равным 0.

Непривилегированным пользователям, которым явно не назначена персональная метка целостности и у которых отсутствует метка целостности группы, неявно присваивается нулевая метка целостности.

Администратору, создаваемому при установке ОС, присваивается максимальный уровень целостности 63.

За системными службами, перечень и описание которых приведены в таблице 1, зарезервированы четыре изолированных уровня целостности.

Таблица 1

Уровень	Значение	Битовая маска	Описание
1	001	0000 0001	Уровень задействован для сетевых служб
2	002	0000 0010	Уровень задействован для виртуализации
3	004	0000 0100	Уровень задействован для специального ПО
4	008	0000 1000	Уровень задействован для графического сервера

Примечание. В текущей реализации, с учетом 32-битной маски, количество изолированных уровней целостности может быть увеличено до 32 при повышении максимального уровня целостности до 0xFFFF FFFF.

После установки ОС максимальный уровень целостности в системе может быть повышен. Максимальными уровнями целостности в системе могут быть числа, у которых битовая маска включает битовые маски всех остальных используемых уровней целостности в системе, например, 63 (0x3F, битовая маска 00111111), 127 (0x7F, битовая маска 01111111), 191 (0xBF, битовая маска 10111111) и т.д.

ВНИМАНИЕ! При повышении максимального уровня целостности в ОС выше значе-

ния 63, заданного при установке ОС, необходимо убедиться в повышении уровня целостности администратора ОС.

2.2. Подраздел «4.7. PARSEC-привилегии»

В таблице 8 уточнить описание привилегии `PARSEC_CAP_SETMAC`:

Таблица 8

Привилегия Атрибут Битовая маска	Описание
`PARSEC_CAP_SETMAC` `0x00004`	Позволяет процессу изменять собственную классификационную метку (уровень и категории конфиденциальности)

2.3. Пункт «4.15.7. pdpl-group»

После пункта 4.15.6 ввести новый пункт 4.15.7 с соответствующим изменением нумерации следующих пунктов:

4.15.7. pdpl-group

Инструмент командной строки `pdpl-group` позволяет устанавливать и просматривать уровень целостности группы пользователей ОС (см. 4.3.1).

Синтаксис инструмента `pdpl-group`:

```
pdpl-group [параметр] [группа]
```

Имя группы задается в текстовом формате. Описание параметров приведено в таблице 16.

Таблица 16

Параметр	Описание
`-d, --delete <группа>`	Удалить строку группы из файла
`-z, --zero <группа>`	Обнулить значение уровня целостности группы
`-i, --ilevel <уровень_целостности> <группа>`	Установить максимальный уровень целостности для группы
`-h, --help`	Вывести справку по использованию инструмента
`-v, --version`	Вывести информацию о версии инструмента

Пример

Присвоить группе `group1` уровень целостности, равный 63:

```
pdpl-group -i 63 group1
```

Результат выполнения команды:

```
minimal pdpl: Уровень_0:Низкий:Нет:0x0
0:0:0x0:0x0
maximal pdpl: Уровень_0:Высокий:Нет:0x0
0:63:0x0:0x0
```

2.4. Пункт «4.17.1. Порядок применения мандатных правил управления доступом»

В таблице 30 уточнить описание параметра `ac_ignore_server_maclabel` для настройки работы сервера СУБД в условиях мандатного управления доступом:

Таблица 30

Параметр	Описание
`ac_ignore_server_maclabel`	Определяет, будет ли сервер СУБД дополнительно использовать свою метку безопасности (метку безопасности процесса, например назначаемую параметром `PDPLabel`, см. 4.12) при определении прав пользователя на добавление, удаление и изменение данных или нет. Если этот параметр установлен в `FALSE`, то метка безопасности сервера используется для блокировки добавления в БД информации с меткой безопасности, превышающей метку безопасности сервера. Если этот параметр установлен в `TRUE`, то метка безопасности сервера не учитывается

2.5. Подраздел «5.2. Мандатное управление доступом к виртуальной машине»

В подразделе 5.2 исключить следующий абзац:

ВНИМАНИЕ! Существуют ограничения по конфигурированию виртуальной машины: в качестве сетевого адаптера не может быть выбрано устройство `virtio`.

2.6. Пункт «5.6.4. Контроль целостности образов ВМ»

После пункта 5.6.3 ввести новый пункт 5.6.4:

5.6.4. Контроль целостности образов ВМ

Механизм контроля целостности образов ВМ позволяет блокировать запуск ВМ из образов, расположенных в файловых хранилищах системы виртуализации, целостность которых была нарушена. Применение механизма возможно только при включенном механизме контроля целостности с использованием алгоритма работы с контрольными суммами (см. 5.6.2).

Примечание. Механизм эффективно работает с небольшими дисками в формате QCOW2, при работе с дисками в формате RAW возможно увеличение затрат времени и ресурсов.

Для включения контроля целостности образов ВМ необходимо в файле /etc/libvirt/libvirtd.conf для параметра integrity_image_control задать значение 1:

```
integrity_image_control = 1
```

Для применения настроек требуется перезапустить службу libvirtd:

```
sudo systemctl restart libvirtd
```

После включения контроля целостности образов ВМ при первом запуске ВМ рассчитывается контрольная сумма (хеш) образа и сохраняется в каталоге /var/lib/libvirt/hash. При выключении ВМ рассчитывается контрольная сумма измененного образа ВМ и перезаписывается. При каждом последующем включении ВМ выполняется проверка контрольной суммы образа. Если вычисленная контрольная сумма не соответствует сохраненной в /var/lib/libvirt/hash, то запуск ВМ блокируется. Также выполняется перерасчет контрольной суммы при создании снимков, сохранении и перезагрузке ВМ.

При установке новой ВМ после включения механизма контроля целостности контрольная сумма образа вычисляется сразу при создании образа.

При включенном контроле целостности образов запуск ВМ выполняется дольше, поэтому для более быстрого вычисления рекомендуется использовать алгоритм xxhash128 вместо gost512. Для выбора алгоритма необходимо в /etc/libvirt/libvirtd.conf для параметра hash_type указать соответствующее значение:

```
hash_type = "xxhash128"
```

2.7. Пункт «5.14.1. Пример организации распределенного хранилища»

Пункт 5.14.1 изложить в редакции:

5.14.1. Пример организации распределенного хранилища

Организация распределенного хранилища для серверов виртуализации на примере двух серверов виртуализации и сервера хранения данных (СХД):

1) СХД — имя компьютера astra-storage, IP-адрес 172.16.1.20;

2) первый сервер виртуализации — имя компьютера astra1, IP-адрес 172.16.1.21;

3) второй сервер виртуализации — имя компьютера astra2, IP-адрес 172.16.1.22.

Компьютеры должны быть объединены в локальную сеть и доступны по имени компьютера (в /etc/hosts на каждом компьютере должны быть выполнены соответствующие настройки).

Для создания сетевого хранилища данных на СХД применяется технология SAN, позволяющая монтировать на компьютере сетевое блочное устройство как локальное

блочное устройство. Доступ к сетевому блочному устройству настраивается по протоколу iSCSI.

5.14.1.1. Создание и подключение сетевого хранилища

Для создания SAN-хранилища (сетевого блочного устройства) необходимо выполнить следующие действия на `astra-storage`:

1) вывести перечень имеющихся блочных устройств командой:

```
lsblk
```

Пример вывода команды:

```
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0 465,8G  0 disk
sdb      8:16   0 298,1G  0 disk
|__sdb1  8:17   0   512M  0 part /boot/efi
|__sdb2  8:18   0 296,7G  0 part /
|__sdb3  8:19   0   977M  0 part [SWAP]
sdc      8:32   0 465,8G  0 disk
|__sdc1  8:33   0 465,8G  0 part
sr0      11:0   1  1024M  0 rom
```

Из вывода команды определить блочное устройство, на основе которого будет создано сетевое блочное устройство (например, `sda`);

2) установить консоль управления Linux-IO Target командой:

```
sudo apt install targetcli-fb
```

3) запустить консоль управления Linux-IO Target командой:

```
sudo targetcli
```

4) в консоли управления Linux-IO Target вывести текущую конфигурацию командой:

```
ls
```

Пример вывода команды:

```
o- / ......................................................... [...]
o- backstores .............................................. [...]
| o- block ..................................... [Storage Objects: 0]
| o- fileio .................................... [Storage Objects: 0]
| o- pscsi ..................................... [Storage Objects: 0]
| o- ramdisk ................................... [Storage Objects: 0]
o- iscsi ................................................ [Targets: 0]
o- loopback ............................................. [Targets: 0]
o- vhost ................................................ [Targets: 0]
o- xen-pvscsi ........................................... [Targets: 0]
```

5) создать (зарегистрировать) сетевое блочное устройство в разделе `/backstores/block` командой:

```
/backstores/block create <имя_устройства> /dev/<блочное_устройство>
```

где <имя_устройства> — наименование, которое будет присвоено создаваемому сетевому блочному устройству;

<блочное_устройство> — выбранное блочное устройство для создания сетевого блочного устройства.

Пример команды для текущей конфигурации:

```
/backstores/block create san_storage /dev/sda
```

Результат выполнения команды:

```
Created block storage object san_storage using /dev/sda
```

6) проверить создание сетевого блочного устройства san_storage командой:

```
ls
```

Результат выполнения команды:

```
o- / ..................................................... [...]
o- backstores ........................................... [...]
| o- block ..................................... [Storage Objects: 1]
| | o- san_storage ...... [/dev/sda (465.8GiB) write-thru deactivated]
| |   o- alua ....................................... [ALUA Groups: 1]
| |     o- default_tg_pt_gp ........... [ALUA state: Active/optimized]
| o- fileio .................................... [Storage Objects: 0]
| o- pscsi ..................................... [Storage Objects: 0]
| o- ramdisk ................................... [Storage Objects: 0]
o- iscsi ..................................................... [Targets: 0]
o- loopback .................................................. [Targets: 0]
o- vhost ..................................................... [Targets: 0]
o- xen-pvscsi ................................................ [Targets: 0]
```

7) создать цель для сетевого блочного устройства (iSCSI-target) в разделе /iscsi командой:

```
/iscsi create
```

Результат выполнения команды:

```
Created target iqn.2003-01.org.linux-iscsi.astra-storage.x8664:
sn.727a2d3719d4.
Created TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260
```

Для сетевого блочного устройства была создана цель iqn.2003-01.org.linux-iscsi.astra-storage.x8664:sn.727a2d3719d4;

8) проверить создание цели командой:

```
ls
```

Результат выполнения команды:

```
o- / ..................................................... [...]
o- backstores ........................................... [...]
| o- block ..................................... [Storage Objects: 1]
| | o- san_storage ..... [/dev/sda (465.8GiB) write-thru deactivated]
| |   o- alua ....................................... [ALUA Groups: 1]
| |     o- default_tg_pt_gp ........... [ALUA state: Active/optimized]
| o- fileio .................................... [Storage Objects: 0]
| o- pscsi ..................................... [Storage Objects: 0]
| o- ramdisk ................................... [Storage Objects: 0]
o- iscsi ....................................... [Targets: 1]
| o- iqn.2003-01.org.linux-iscsi.astra-storage.x8664:sn.727a2d3719d4
[TPGs: 1]
|   o- tpg1 ................................ [no-gen-acls, no-auth]
|     o- acls ..................................... [ACLs: 0]
|     o- luns ..................................... [LUNs: 0]
|     o- portals ................................... [Portals: 1]
|       o- 0.0.0.0:3260 ..................................... [OK]
o- loopback .................................... [Targets: 0]
o- vhost ....................................... [Targets: 0]
o- xen-pvscsi .................................. [Targets: 0]
```

9) создать LUN (Logical Unit Number — номер логического устройства) на основе сетевого блочного устройства (зарегистрированного в разделе `/backstores/block`) командой:

```
/iscsi/<цель>/tpg1/luns/ create /backstores/block/<имя_устройства>
```

где <цель> — цель для сетевого блочного устройства, созданная согласно пункту перечисления 7);

<имя_устройства> — наименование сетевого блочного устройства, заданное при его создании согласно пункту перечисления 5).

Пример команды для текущей конфигурации:

```
/iscsi/iqn.2003-01.org.linux-iscsi.astra-storage.x8664:sn.727a2d3719d4/
tpg1/luns/ create /backstores/block/san_storage
```

Результат выполнения команды:

```
Created LUN 0
```

10) проверить результат создания LUN командой:

```
ls
```

Результат выполнения команды:

```
o- / ..................................................... [...]
```

```
o- backstores ..................................................... [...]
| o- block ......................................... [Storage Objects: 1]
| | o- san_storage ........ [/dev/sda (465.8GiB) write-thru activated]
| |   o- alua ....................................... [ALUA Groups: 1]
| |     o- default_tg_pt_gp ........... [ALUA state: Active/optimized]
| o- fileio ....................................... [Storage Objects: 0]
| o- pscsi ........................................ [Storage Objects: 0]
| o- ramdisk ...................................... [Storage Objects: 0]
o- iscsi ................................................... [Targets: 1]
| o- iqn.2003-01.org.linux-iscsi.astra-storage.x8664:sn.727a2d3719d4
TPGs: 1]
|   o- tpg1 ................................... [no-gen-acls, no-auth]
|     o- acls ............................................... [ACLs: 0]
|     o- luns ............................................... [LUNs: 1]
|     | o- lun0 .... [block/san_storage (/dev/sda) (default_tg_pt_gp)]
|     o- portals ......................................... [Portals: 1]
|       o- 0.0.0.0:3260 ....................................... [OK]
o- loopback ................................................ [Targets: 0]
o- vhost ................................................... [Targets: 0]
o- xen-pvscsi .............................................. [Targets: 0]
```

11) в данном примере не используется авторизация, поэтому для отключения контроля доступа к цели необходимо последовательно выполнить команды:

```
cd /iscsi/<цель>/tpg1
set attribute generate_node_acls=1
set attribute demo_mode_write_protect=0
```

Примеры команд для текущей конфигурации:

а) `cd /iscsi/iqn.2003-01.org.linux-iscsi.astra-storage.x8664:sn.727a2d3719d4/tpg1`

б) `set attribute generate_node_acls=1`

Результат выполнения команды:

```
Parameter generate_node_acls is now '1'
```

в) `set attribute demo_mode_write_protect=0`

Результат выполнения команды:

```
Parameter demo_mode_write_protect is now '0'
```

12) сохранить конфигурацию сетевого блочного устройства командой:

```
/ saveconfig
```

Результат выполнения команды:

```
Configuration saved to /etc/rtslib-fb-target/saveconfig.json
```

Конфигурация сетевого блочного устройства сохранена в файл /etc/rtslib-fb-target/saveconfig.json;

13) выйти из консоли управления Linux-IO Target командой:

```
exit
```

Результат выполнения команды:

```
Global pref auto_save_on_exit=true
Last 10 configs saved in /etc/rtslib-fb-target/backup/.
Configuration saved to /etc/rtslib-fb-target/saveconfig.json
```

5.14.1.2. Подключение на серверах виртуализации сетевого блочного устройства

Созданное на astra-storage сетевое блочное устройство (iSCSI-target) необходимо подключить в качестве локального блочного устройства на серверах виртуализации (iSCSI-initiator).

Данные настройки должны быть выполнены на всех компьютерах, на которых настраивается доступ к сетевому блочному устройству. В данном примере настройки выполняются на серверах виртуализации astra1 и astra2. Для подключения сетевого блочного устройства к серверу виртуализации необходимо:

1) установить пакет open-iscsi командой:

```
sudo apt install open-iscsi
```

2) настроить автоматическое подключение LUN при перезагрузке сервера виртуализации. Для этого в конфигурационном файле /etc/iscsi/iscsid.conf для параметра node.startup установить значение automatic:

```
node.startup = automatic
```

3) запустить службу iscsi командой:

```
sudo systemctl start iscsi
```

После первого запуска службы iscsi будет сгенерирован уникальный идентификатор инициатора (iSCSI-initiator), который можно просмотреть в файле /etc/iscsi/initiatorname.iscsi, выполнив команду:

```
sudo cat /etc/iscsi/initiatorname.iscsi
```

Результат выполнения команды на astra1:

```
## DO NOT EDIT OR REMOVE THIS FILE!
## If you remove this file, the iSCSI daemon will not start.
## If you change the InitiatorName, existing access control lists
## may reject this initiator.  The InitiatorName must be unique
## for each iSCSI initiator.  Do NOT duplicate iSCSI InitiatorNames.
InitiatorName=iqn.1993-08.org.debian:01:694dac4a9eba
```

Результат выполнения команды на astra2:

```
## DO NOT EDIT OR REMOVE THIS FILE!
## If you remove this file, the iSCSI daemon will not start.
```

```
## If you change the InitiatorName, existing access control lists
## may reject this initiator.  The InitiatorName must be unique
## for each iSCSI initiator.  Do NOT duplicate iSCSI InitiatorNames.
InitiatorName=iqn.1993-08.org.debian:01:7c60f380d294
```

4) выполнить поиск доступных целей (сетевых блочных устройств, iSCSI-target):

```
sudo iscsiadm -m discovery -t st -p <IP-адрес>
```

где <IP-адрес> — IP-адрес компьютера, на котором зарегистрировано сетевое блочное устройство.

Пример команды для текущей конфигурации:

```
sudo iscsiadm -m discovery -t st -p 172.16.1.20
```

Результат выполнения команды:

```
172.16.1.20:3260,1 iqn.2003-01.org.linux-iscsi.astra-storage.x8664:
sn.727a2d3719d4
```

5) автоматически подключить все найденные цели:

```
sudo iscsiadm -m node -l
```

Результат выполнения команды:

```
Logging in to [iface: default, target:
iqn.2003-01.org.linux-iscsi.astra-storage.x8664:sn.727a2d3719d4,
portal: 172.16.1.20,3260]
Login to [iface: default, target:
iqn.2003-01.org.linux-iscsi.astra-storage.x8664:sn.727a2d3719d4,
portal: 172.16.1.20,3260] successful.
```

6) проверить, что на сервере виртуализации было добавлено новое блочное устройство:

```
lsblk
```

Результат выполнения команды на astra1:

```
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0 931,5G  0 disk
|_sda1   8:1    0   512M  0 part /boot/efi
|_sda2   8:2    0    28G  0 part /
|_sda3   8:3    0  18,6G  0 part [SWAP]
|_sda4   8:4    0 884,5G  0 part /home
sdb      8:16   0 465,8G  0 disk
sr0     11:0    1  1024M  0 rom
```

На astra1 добавлено новое блочное устройство /dev/sdb.

Результат выполнения команды на astra2:

```
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda           8:0    0 465,8G  0 disk
```

```
nvme0n1     259:0    0 238,5G  0 disk
|_nvme0n1p1 259:1    0   512M  0 part /boot/efi
|_nvme0n1p2 259:2    0    28G  0 part /
|_nvme0n1p3 259:3    0   7,5G  0 part [SWAP]
|_nvme0n1p4 259:4    0 202,6G  0 part /home
```

На astra2 добавлено новое блочное устройство /dev/sda.

5.14.1.3. Настройка кластера

Для использования сетевого хранилища серверами виртуализации требуется создать кластер и разметить на нем кластерную файловую систему OCFS2.

Создание кластера

Для создания кластера необходимо на одном из узлов кластера (например, astra1) выполнить следующие действия:

1) установить пакет ocfs2-tools командой:

```
sudo apt install ocfs2-tools
```

2) создать ocfs2-кластер командой:

```
sudo o2cb add-cluster <имя_кластера>
```

Например:

```
sudo o2cb add-cluster ocfs2cluster
```

В результате выполнения команды будет создан файл конфигурации кластера /etc/ocfs2/cluster.conf;

3) добавить описание узлов кластера, выполнив команду для каждого узла кластера:

```
sudo o2cb add-node <имя_кластера> <сетевое_имя_узла>
--ip <IP-адрес_узла>
```

где <имя_кластера> — имя созданного кластера;

<сетевое_имя_узла> — имя компьютера, добавляемого в качестве узла кластера;

<IP-адрес_узла> — IP-адрес компьютера, добавляемого в качестве узла кластера.

Имя узла кластера должно соответствовать имени компьютера, указанному в /etc/hostname, также имена компьютеров и их IP-адреса должны быть указаны в /etc/hosts на других узлах кластера.

Пример команд для текущей конфигурации:

```
sudo o2cb add-node ocfs2cluster astra1 --ip 172.16.1.21
sudo o2cb add-node ocfs2cluster astra2 --ip 172.16.1.22
```

Данные о добавленных узлах кластера записываются в /etc/ocfs2/cluster.conf.

Проверить содержимое файла конфигурации кластера возможно с помощью команды:

```
cat /etc/ocfs2/cluster.conf
```

Результат выполнения команды:

```
cluster:
heartbeat_mode = local
node_count = 2
name = ocfs2cluster

node:
number = 0
cluster = ocfs2cluster
ip_port = 7777
ip_address = 172.16.1.21
name = astra1

node:
number = 1
cluster = ocfs2cluster
ip_port = 7777
ip_address = 172.16.1.22
name = astra2
```

На всех остальных узлах кластера требуется выполнить следующие действия для настройки кластера:

1) установить пакет `ocfs2-tools` командой:

```
sudo apt install ocfs2-tools
```

2) скопировать в локальный каталог `/etc/ocfs2/` файл конфигурации кластера:

```
sudo scp
     <локальный_администратор>@<IP-адрес_узла>:/etc/ocfs2/cluster.conf
     /etc/ocfs2/
```

где `<IP-адрес_узла>` — IP-адрес узла кластера, на котором была выполнена настройка кластера и с которого копируется файл конфигурации;

`<локальный_администратор>` — имя локального администратора узла кластера, на котором была выполнена настройка кластера и с которого копируется файл конфигурации.

В ходе выполнения команды требуется:

- на запрос пароля для команды sudo — ввести пароль локального администратора узла кластера, на который копируется файл;
- на запрос установки соединения ответить «yes» («Да»);
- ввести пароль локального администратора узла кластера, с которого копируется файл конфигурации.

Пример команды для текущей конфигурации, выполняется на astra2:

```
sudo scp admin1@172.16.1.21:/etc/ocfs2/cluster.conf /etc/ocfs2
```

Результат выполнения команды:

```
[sudo] пароль для admin2:
The authenticity of host '172.16.1.21 (172.16.1.21)' can't be established.
ECDSA key fingerprint is SHA256:mONkRKlGRAc1ZEywS/sYRVdZINN2PAHJIKRwLFzMGSM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.1.21' (ECDSA) to the list of known hosts.
admin1@172.16.1.21's password:
cluster.conf 100% 371 197.3KB/s 00:00
```

На узле astra2 проверить полученный файл конфигурации:

```
cat /etc/ocfs2/cluster.conf
```

Файл конфигурации кластера на узле astra2 /etc/ocfs2/cluster.conf должен быть идентичен файлу конфигурации кластера на узле astra1.

На каждом узле настроить кластер OCFS2:

1) запустить мастер настройки кластера OCFS2 командой:

```
sudo dpkg-reconfigure ocfs2-tools
```

2) в мастере настройки кластера OCFS2:
   - разрешить запускать кластер OCFS2 (O2CB) во время загрузки, нажав кнопку **[Да]**;
   - задать имя кластера (например, ocfs2cluster) и нажать кнопку **[Ok]**;
   - для остальных параметров выбрать значения, установленные по умолчанию;
3) перезапустить службу o2cb командой:

```
sudo systemctl restart o2cb
```

4) проверить успешный запуск службы o2cb. Для этого просмотреть в журнале регистрации события, относящиеся к службе:

```
sudo journalctl -u o2cb
```

Пример вывода команды на узле astra1:

```
-- Logs begin at Thu 2023-06-01 13:49:19 MSK, end at Thu 2023-06-01
14:12:52 MSK. --
июн 01 14:10:38 astra1 systemd[1]: Starting Load o2cb Modules...
июн 01 14:10:38 astra1 o2cb[3293]: checking debugfs...
июн 01 14:10:39 astra1 o2cb[3293]: Loading stack plugin "o2cb": OK
```

```
июн 01 14:10:39 astra1 o2cb[3293]: Loading filesystem "ocfs2_dlmfs": OK
июн 01 14:10:39 astra1 o2cb[3293]: Mounting ocfs2_dlmfs filesystem at /dlm: OK
июн 01 14:10:39 astra1 o2cb[3293]: Setting cluster stack "o2cb": OK
июн 01 14:10:39 astra1 o2cb[3293]: Registering O2CB cluster "ocfs2cluster": OK
июн 01 14:10:39 astra1 o2cb[3293]: Setting O2CB cluster timeouts : OK
июн 01 14:10:39 astra1 o2hbmonitor[3342]: Starting
июн 01 14:10:39 astra1 systemd[1]: Started Load o2cb Modules.
июн 01 14:12:34 astra1 systemd[1]: Stopping Load o2cb Modules...
июн 01 14:12:34 astra1 o2cb[3418]: Clean userdlm domains: OK
июн 01 14:12:34 astra1 o2cb[3418]: Stopping O2CB cluster ocfs2cluster:
Unregistering O2
июн 01 14:12:34 astra1 o2cb[3418]: Unmounting ocfs2_dlmfs filesystem: OK
июн 01 14:12:34 astra1 o2cb[3418]: Unloading module "ocfs2_dlmfs": OK
июн 01 14:12:34 astra1 o2cb[3418]: Unloading module "ocfs2_stack_o2cb": OK
июн 01 14:12:34 astra1 systemd[1]: o2cb.service: Succeeded.
июн 01 14:12:34 astra1 systemd[1]: Stopped Load o2cb Modules.
июн 01 14:12:34 astra1 systemd[1]: o2cb.service: Consumed 185ms CPU time.
июн 01 14:12:34 astra1 systemd[1]: Starting Load o2cb Modules...
июн 01 14:12:34 astra1 o2cb[3485]: checking debugfs...
июн 01 14:12:34 astra1 o2cb[3485]: Loading stack plugin "o2cb": OK
июн 01 14:12:34 astra1 o2cb[3485]: Loading filesystem "ocfs2_dlmfs": OK
июн 01 14:12:34 astra1 o2cb[3485]: Mounting ocfs2_dlmfs filesystem at /dlm: OK
июн 01 14:12:34 astra1 o2cb[3485]: Setting cluster stack "o2cb": OK
июн 01 14:12:34 astra1 o2cb[3485]: Registering O2CB cluster "ocfs2cluster": OK
июн 01 14:12:34 astra1 o2cb[3485]: Setting O2CB cluster timeouts : OK
июн 01 14:12:34 astra1 o2hbmonitor[3531]: Starting
июн 01 14:12:34 astra1 systemd[1]: Started Load o2cb Modules.
```

В журнале должны отсутствовать сообщения об ошибках запуска службы. Для выхода из просмотра журнала нажать клавишу **<q>**.

Разметка кластерной файловой системы

Разметка подключенного сетевого блочного устройства для использования кластерной файловой системы OCFS2 выполняется на одном из узлов кластера.

Для разметки необходимо выполнить команду:

```
sudo mkfs.ocfs2 -T vmstore <блочное_устройство>
```

где -T — запустить автоматическую тонкую настройку параметров файловой системы;

vmstore — производить тонкую настройку для обеспечения максимальной производительности при размещении на блочном устройстве файлов образов виртуальных машин;

<блочное_устройство> — сетевое блочное устройство, подключенное согласно 2.7.

Для текущей конфигурации выполнить разметку сетевого блочного устройства на узле astra1 командой:

```
sudo mkfs.ocfs2 -T vmstore /dev/sdb
```

Вывод команды:

```
[sudo] пароль для admin1:
mkfs.ocfs2 1.8.5
Cluster stack: classic o2cb
Filesystem Type of vmstore
Label:
Features: sparse extended-slotmap backup-super unwritten inline-data
strict-journal-super xattr indexed-dirs refcount discontig-bg append-dio
Block size: 4096 (12 bits)
Cluster size: 1048576 (20 bits)
Volume size: 500107837440 (476940 clusters) (122096640 blocks)
Cluster groups: 15 (tail covers 25356 clusters, rest cover 32256 clusters)
Extent allocator size: 188743680 (45 groups)
Journal size: 134217728
Node slots: 8
Creating bitmaps: done
Initializing superblock: done
Writing system files: done
Writing superblock: done
Writing backup superblock: 5 block(s)
Formatting Journals: done
Growing extent allocator: done
Formatting slot map: done
Formatting quota files: done
Writing lost+found: done
mkfs.ocfs2 successful
```

В результате на блочном устройстве /dev/sdb (сетевое блочное устройство san_storage) была создана файловая система OCFS2.

Для просмотра доступных сетевых блочных устройств с кластерной файловой системой OCFS2 выполнить команду на любом из узлов кластера:

```
sudo mounted.ocfs2 -d
```

Результат выполнения команды на узле astra1 текущей конфигурации:

```
Device    Stack  Cluster  F  UUID                                Label
/dev/sdb  o2cb             E42202DFA47B42DB9B7DE6778555B4A4
```

Размеченное сетевое блочное устройство необходимо примонтировать на каждом узле кластера, для этого требуется:

1) создать точку монтирования:

```
sudo mkdir -p <точка_монтирования>
```

2) примонтировать сетевое блочное устройство:

```
sudo mount <сетевое_блочное_устройство> <точка_монтирования>
```

Пример команд на узле astra1 для текущей конфигурации:

1) создать точку монтирования:

```
sudo mkdir -p /mnt/ocfs2-storage
```

2) примонтировать сетевое блочное устройство /dev/sdb:

```
sudo mount /dev/sdb /mnt/ocfs2-storage/
```

3) для проверки монтирования сетевого блочного устройства выполнить команду:

```
lsblk
```

Результат выполнения команды:

```
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0 931,5G  0 disk
|_sda1   8:1    0   512M  0 part /boot/efi
|_sda2   8:2    0    28G  0 part /
|_sda3   8:3    0  18,6G  0 part [SWAP]
|_sda4   8:4    0 884,5G  0 part /home
sdb      8:16   0 465,8G  0 disk /mnt/ocfs2-storage
sr0     11:0    1  1024M  0 rom
```

Для просмотра примонтированных сетевых блочных устройств с кластерной файловой системой OCFS2 необходимо выполнить команду на любом из узлов кластера:

```
sudo mounted.ocfs2 -f
```

Результат выполнения команды на узле astra1 текущей конфигурации:

```
Device    Stack  Cluster  F  Nodes
/dev/sdb  o2cb             astra1, astra2
```

В выводе команды отображается сетевое блочное устройство и список узлов кластера, на которых оно примонтировано.

5.14.1.4. Автоматическое монтирование

Для обеспечения автоматического монтирования сетевого блочного устройства при загрузке узла необходимо:

1) определить UUID блочного устройства, выполнив команду на любом из узлов кластера:

```
sudo blkid
```

Результат выполнения команды на узле astra1 текущей конфигурации:

```
/dev/sda1: UUID="503E-AC24" TYPE="vfat"
PARTUUID="8dabfe8a-031e-4da6-b5be-33a8cba97322"
/dev/sda2: UUID="68de36df-e90e-4fb9-8bba-47dd0b5f4dca"
TYPE="ext4" PARTUUID="e2aa5324-2f15-4962-98f2-744ca4e8f844"
/dev/sda3: UUID="b2f367e4-3e3f-4c58-a159-0e222ebdc422"
TYPE="swap" PARTUUID="f2e75fdf-db3e-40ee-8a9f-ab634363c6fe"
/dev/sda4: UUID="abc173bd-a889-499e-9424-96ce8f8c2b9b"
TYPE="ext4" PARTUUID="f5de7620-7636-4bb0-afe2-1c4192a5c1aa"
/dev/sdb: UUID="e42202df-a47b-42db-9b7d-e6778555b4a4" TYPE="ocfs2"
```

2) на каждом узле кластера отредактировать файл /etc/fstab, добавив строку вида:

```
UUID=e42202df-a47b-42db-9b7d-e6778555b4a4       /mnt/ocfs2-storage
ocfs2   _netdev,x-systemd.requires=o2cb.service   0       0
```

где e42202df-a47b-42db-9b7d-e6778555b4a4 — UUID сетевого блочного устройства;

/mnt/ocfs2-storage — точка монтирования;

ocfs2 — тип файловой системы OCFS2;

_netdev,x-systemd.requires=o2cb.service — параметры монтирования, задающие монтирование после загрузки сетевых служб и при запущенной службе o2cb.

5.14.1.5. Централизованное управление в среде виртуализации

Для создания централизованного хранилища образов (pool, пул) необходимо на каждом сервере виртуализации, объединенном в кластер (astra1, astra2), выполнить добавление пула:

1) на узле astra1 войти в ОС под учетной записью администратора, входящего в группу libvirt-admin. Если в системе включено ролевое управление доступом в среде виртуализации (см. 5.3), то необходимо выполнить настройки в соответствии с 5.3.1;

2) запустить virt-manager, в главном окне программы выбрать строку подключения «QEMU/KVM» и в меню выбрать «Правка — Свойства подключения»;

3) в окне «QEMU/KVM — сведения о подключении» перейти во вкладку «Пространство данных» и нажать кнопку **[+]** (**[Добавить пул]**);

4) в окне «Добавление пространства»:

а) в поле «Название» задать наименование пула, например «ocfs2-pool» (название пула на всех узлах может быть одинаковым для удобства);

б) в поле «Путь к цели» указать точку монтирования сетевого блочного устройства (на узле `astra1` для текущей конфигурации `/mnt/ocfs2-storage`);

в) нажать **[Готово]**;

5) проверить наличие подключенного хранилища — пул должен отображаться во вкладке «Пространство данных» в соответствии с рис. 5;

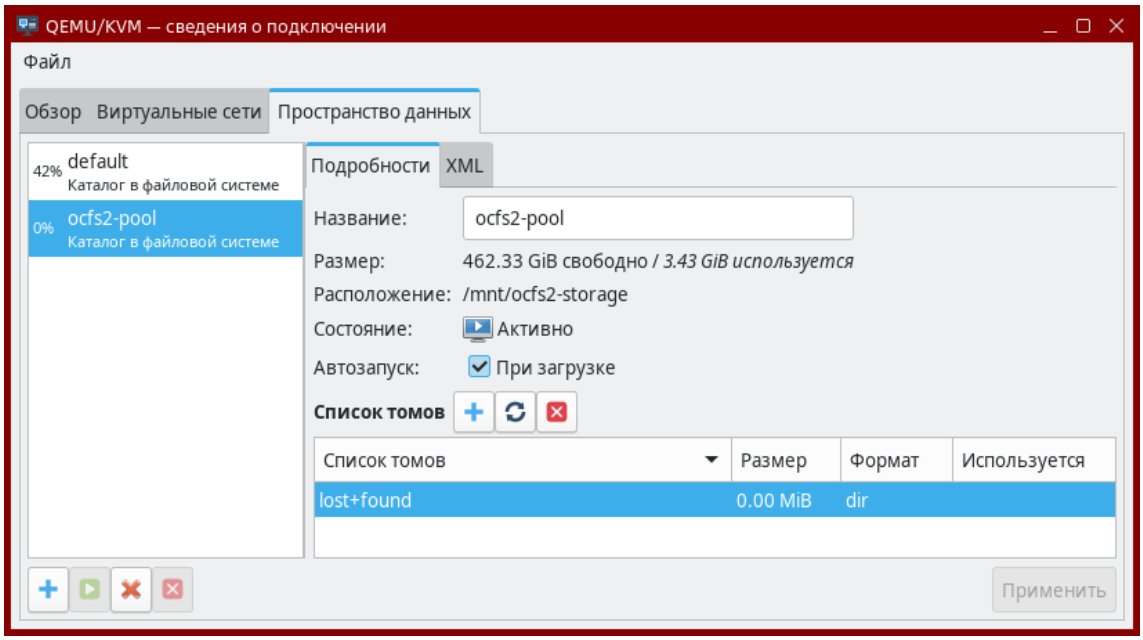

Рис. 5

Для проверки совместного использования пула необходимо:

1) на одном из узлов кластера (например, `astra2`) выполнить копирование образа ВМ в хранилище командой:

```
sudo cp <путь_к_образу>
     <точка_монтирования_сетевого_блочного_устройства>
```

Пример:

```
sudo cp ~/alse-vanilla-1.7.2-qemu-max-mg8.0.0.qcow2 /mnt/ocfs2-storage
```

2) на каждом из узлов кластера проверить отображение образа во вкладке «Пространство данных» в списке томов (предварительно требуется нажать кнопку обновления списка томов) в соответствии с рис. 6.

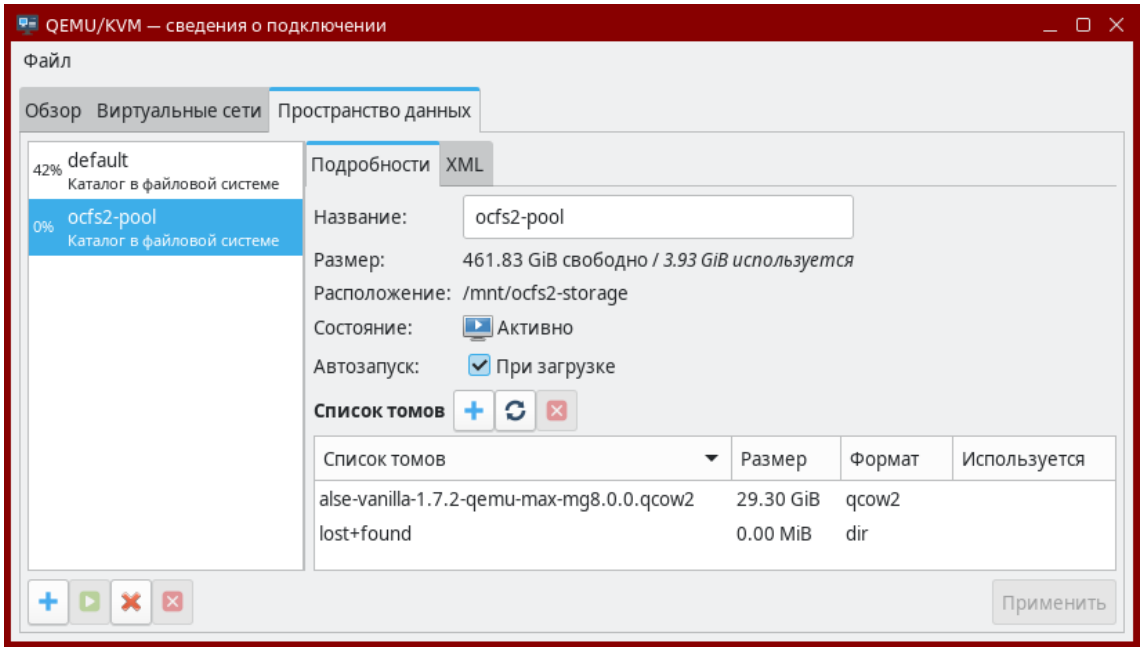

Рис. 6

2.8. Пункт «5.14.2. Пример миграции виртуальных машин»

Пункт 5.14.2 изложить в редакции:

5.14.2. Пример организации распределенного хранилища

Миграция запущенной ВМ с одного сервера виртуализации на другой сервер виртуализации возможна при использовании серверами виртуализации распределенного хранилища.

Миграция выполняется по SSH, поэтому предварительно на каждом сервере виртуализации должна быть настроена служба `ssh` (описание настройки `ssh` см. в РУСБ.10015-01 95 01-1) и на одном из серверов виртуализации должна быть создана ВМ.

На сервере виртуализации, с которого будет выполняться миграция запущенной ВМ (например, `astra1`), необходимо подключить по SSH второй сервер виртуализации (например, `astra2`). Для этого:

1) войти в ОС на узле `astra1` и запустить `virt-manager`;

2) в меню выбрать «Файл — Добавить соединение» и добавить подключение по SSH к узлу `astra2` в соответствии с рис. 7, указав в поле «Имя пользователя» учетную запись администратора узла `astra2`;

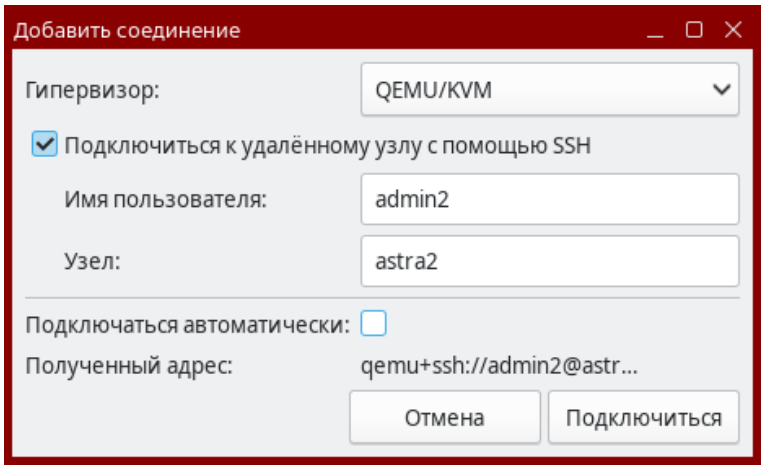

Рис. 7

3) в окне подтверждения соединения нажать **[Да]**;

4) в окне авторизации ввести пароль от учетной записи администратора на узле `astra2`;

5) в главном окне `virt-manager` в списке соединений будет добавлено подключение к узлу `astra2` в соответствии с рис. 8.

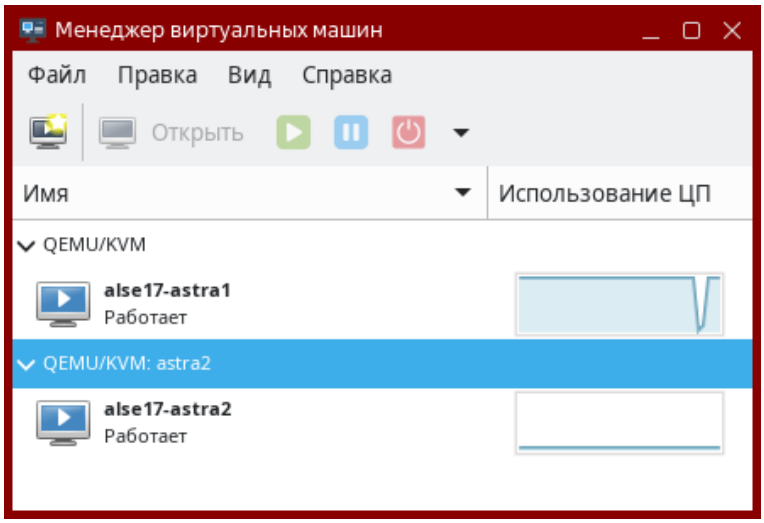

Рис. 8

Для создания ВМ на сервере виртуализации `astra1`, например путем импорта образа диска ВМ, добавленного в хранилище согласно 5.14.1.5, необходимо:

1) запустить `virt-manager`;

2) выбрать подключение «QEMU/KVM» и в меню выбрать «Файл — Создать виртуальную машину»;

3) в открывшемся окне «Новая виртуальная машина» выбрать «Импорт образа диска» и нажать **[Вперед]**;

4) на следующем шаге:

   а) в поле «Укажите путь к пространству хранения» указать в качестве источника образ ВМ в хранилище (например, `ocfs2-pool`);

б) в поле «Выберите операционную систему для установки» выбрать название операционной системы импортируемой ВМ в соответствии с рис. 9;

в) для перехода к следующему шагу нажать **[Вперед]**;

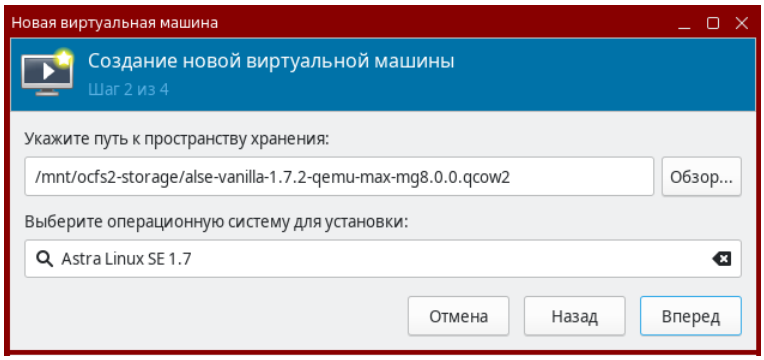

Рис. 9

5) на следующем шаге, при необходимости, задать значения памяти и количества процессоров, затем нажать **[Вперед]**;

6) на следующем шаге в поле «Название» указать имя ВМ, например `alse-astra1`. При необходимости проверить или изменить конфигурацию ВМ — установить флаг «Проверить конфигурацию перед установкой». Затем для окончания настройки, создания и запуска импортированной ВМ нажать **[Готово]**.

Для выполнения миграции запущенной ВМ с одного сервера виртуализации на другой сервер виртуализации необходимо в главном окне `virt-manager` открыть контекстное меню ВМ и выбрать «Миграция». В открывшемся окне из раскрывающегося списка «Новый узел» выбрать сервер виртуализации, на который необходимо переместить ВМ, и нажать **[Миграция]** в соответствии с рис. 10.

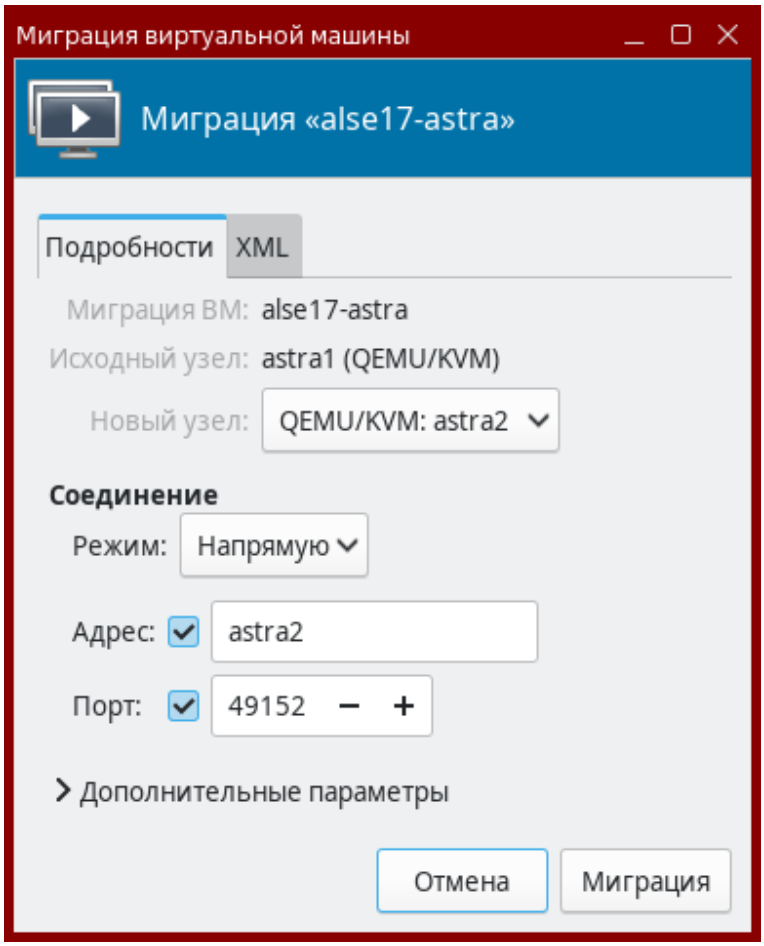

Рис. 10

В результате выполнения миграции виртуальная машина `alse-astra1` мигрирует с узла `astra1` на `astra2` (см. рис. 11).

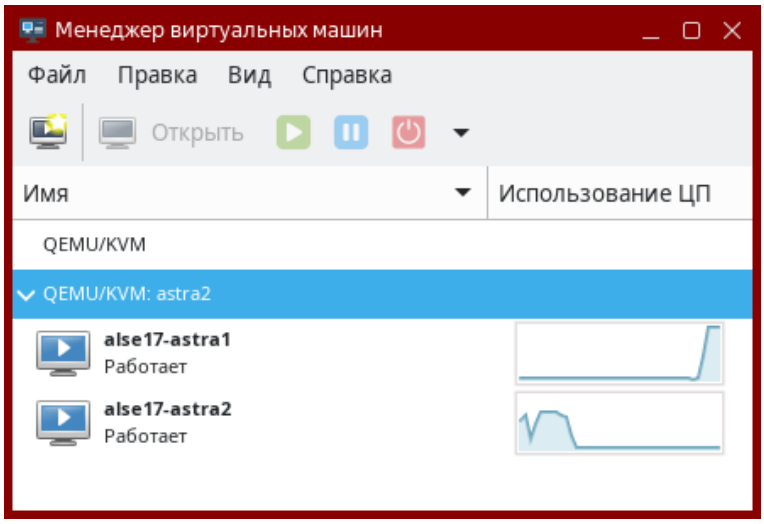

Рис. 11

2.9. Подраздел «6.1. Правила регистрации событий»

Подраздел 6.1 изложить в следующей редакции:

6.1. Правила регистрации событий

Регистрация событий осуществляется в соответствии с правилами аудита, правила делятся на два типа:

1) временные — действуют до перезагрузки системы. Данные правила относятся к подсистеме ядерного аудита: они задаются посредством инструмента `auditctl` и начинают выполняться при запуске службы `auditd`;

2) постоянные — действуют всегда, даже после перезагрузки системы. Такие правила задаются в файлах формата `*.rules`, располагающихся в каталоге `/etc/audit/rules.d/`.

Подробное описание правил аудита, а также синтаксис использования инструмента `auditctl` приведены в справочной странице `man auditctl`.

Примеры:

1. Регистрировать все системные вызовы от процесса с идентификатором (PID) 1005:

```
auditctl -a exit,always -S all -F pid=1005
```

2. Регистрировать все файлы, открытые пользователем с идентификатором auid 510:

```
auditctl -a exit,always -S open -F auid=510
```

При добавлении постоянных правил аудита в файлах используется синтаксис инструмента `auditctl` без указания имени инструмента.

Пример

Регистрировать все системные вызовы от процесса с идентификатором (PID) 1005: `-a exit,always -S all -F pid=1005`

Подсистема безопасности PARSEC предоставляет дополнительный способ выбора событий для регистрации — PARSEC-аудит. Он использует правила регистрации событий, назначаемые на процессы и файлы с помощью инструментов командной строки `setfaud` (см. 6.4.3), `useraud` (см. 6.4.4), `psaud` (см. 6.4.5).

Для работы PARSEC-аудита в файлах `/etc/audit/rules.d/10-parsec.rules` (аудит процессов и файлов) и `/etc/audit/rules.d/10-parsec-nw.rules` (сетевой аудит) заданы следующие постоянные правила:

1) правила аудита процессов (необходимы для работы `useraud` и `psaud`):

```
-a always,exit -F subj_type=psaud -F arch=<архитектура> -S
<системные_вызовы> -k parsec-p
```

где `-a always,exit` — регистрировать события в журнале, добавить правило в список `exit` (события, происходящие при выходе из системного вызова);

`-F subj_type=psaud` — обрабатывать правила, заданные с помощью `useraud` и `psaud`;

`-F arch=<архитектура>` — аппаратная платформа;

`-S <системные_вызовы>` — перехватывать события при вызовах, указанных в `<системные_вызовы>` (список вызовов, разделенных запятой);

`-k parsec-p` — присвоить ключ фильтрации `parsec-p` событиям по данному правилу;

2) правила аудита файлов (необходимы для работы `getfaud` и `setfaud`):

```
-a always,exit -F obj_type=faud -F arch=<архитектура> -S
<системные_вызовы> -k parsec-f
```

где `-a always,exit` — регистрировать события в журнале, добавить правило в список `exit` (события, происходящие при выходе из системного вызова);

`-F obj_type=faud` — обрабатывать правила, заданные с помощью `setfaud`;

`-F arch=<архитектура>` —- аппаратная платформа;

`-S <системные_вызовы>` — перехватывать события при вызовах, указанных в `<системные_вызовы>` (список вызовов, разделенных запятой);

`-k parsec-f` — присвоить ключ фильтрации `parsec-f` событиям по данному правилу;

3) правила сетевого аудита (анализируют работу системных вызовов, связанных со взаимодействием по сети):

```
-a always,exit -F subj_type=psaud -F arch=<архитектура> -S
<сетевые_системные_вызовы> -k parsec-p
```

где `-a always,exit` — регистрировать события в журнале, добавить правило в список `exit` (события, происходящие при выходе из системного вызова);

`-F subj_type=psaud` — обрабатывать правила, заданные с помощью `useraud` и `psaud`;

`-F arch=<архитектура>` — аппаратная платформа;

`-S <сетевые_системные_вызовы>` — перехватывать события при вызовах, указанных в `<сетевые_системные_вызовы>` (список вызовов, разделенных запятой);

`-k parsec-p` — присвоить ключ фильтрации `parsec-p` событиям по данному правилу.

Включение и выключение правил PARSEC-аудита процессов и файлов выполняется с помощью инструмента командной строки `astra-audit-control` (описание приведе-

но в 16.5.34). Включение и выключение правил сетевого PARSEC-аудита выполняется с помощью инструмента командной строки `astra-audit-network-control` (описание приведено в 16.5.35).

В дополнение к постоянным правилам PARSEC-аудита (файлов, процессов и сетевого) можно задавать собственные постоянные правила аудита. Правила рекомендуется добавлять в файл `/etc/audit/rules.d/audit.rules`. При необходимости возможно создать в каталоге `/etc/audit/rules.d/` новый файл с произвольным именем и расширением `*.rules` и задать в нем необходимые правила. Файл `/etc/audit/rules.d/audit.rules` можно редактировать вручную или с помощью графической утилиты графической утилиты `system-config-audit` («Конфигурация аудита», описание утилиты приведено в электронной справке). Другие файлы правил можно редактировать только вручную.

2.10. Пункт «6.4.3. setfaud»

Пункт 6.4.3 изложить в редакции:

6.4.3. setfaud

Команда `setfaud` устанавливает на файлы и каталоги списки правил регистрации событий. Правила задаются или в командной строке (параметры `-s`, `-m`), или в файлах (параметры `-S`, `-M`, `-B`). При этом файлы могут быть сформированы с помощью перенаправления вывода команды `getfaud` (см. 6.4.2).

Только администратор может изменять списки правил регистрации событий у файлов и каталогов.

Синтаксис команды:

```
setfaud [параметры] [правила_регистрации] [объект]
```

Правила регистрации задаются в виде:

```
[u:<пользователь>:<флаги_регистрации>]
     [,g:<группа>:<флаги_регистрации>][,o:<флаги_регистрации>]
```

где `<пользователь>` и `<группа>` — символические или численные идентификаторы пользователя и группы;

`u:` — правило для пользователя;

`g:` — правило для группы;

`o:` — правило для остальных пользователей (для которых правила не заданы явно).

Флаги регистрации задаются в виде:

```
<флаги_успешных_операций>[[:<флаги_неуспешных_операций>], ...]
```

При этом флаги операций могут иметь вид:

1) `<+|-><имя_регистрируемого_события>,...`

Пример

Установка списка правил регистрации успешного события выполнения файла, регистрации неуспешного удаления файла и удаление правила регистрации неуспешного открытия файла:

```
+exec:+delete-open
```

2) `[+|-]<число>`

3) `<сокращенное_имя_регистрируемого_события>,...`

Пример

Регистрация успешных событий открытия и удаления файла (и никаких других):

```
ou
```

Для каталогов, наряду с обычным списком правил регистрации событий, возможно установить список правил регистрации событий по умолчанию. Если на каталог установлен список правил регистрации по умолчанию, то на все файлы и дочерние каталоги, которые будут созданы в данном каталоге, будут установлены списки регистрации событий из списка по умолчанию.

Для внесения изменений в список по умолчанию используется параметр `-d`. Данный параметр взаимоисключаем с параметром `-R`. При использовании параметра `-R` изменяются рекурсивно списки правил регистрации событий для каталога и всех уже существующих файлов и дочерних каталогов в нем. При использовании нескольких параметров параметр `-d` должен быть первым (`-ds` — правильно, `-sd` — не будет работать).

Примеры:

1. Регистрация успешных и неуспешных операций удаления файлов и дочерних каталогов в каталоге `/opt/test`, выполняемых пользователем `username`. Правила регистрации будут назначаться всем объектам, создаваемым в `/opt/test`. Правила для уже существующих объектов изменены не будут.

```
setfaud -ds u:username:+delete:+delete /opt/test
```

2. Регистрация успешных и неуспешных операций запуска файлов на исполнение в каталоге `/opt/test`, выполняемых пользователем `username`. Правила регистрации будут назначены данному каталогу и уже существующим объектам в данном каталоге. Создаваемые в каталоге `/opt/test` объекты не будут иметь назначенных правил регистрации.

```
setfaud -Rs u:username:+exec:+exec /opt/test
```

Описание параметров `setfaud` приведено в таблице 31.

Таблица 31

Параметр	Описание
-s, --set	Установить список регистрируемых событий из командной строки
-b, --remove	Удалить все элементы списка регистрируемых событий
-m, --modify	Изменить или добавить элементы списка из командной строки
-d, --default	Используется для каталогов. Работать со списком регистрируемых событий по умолчанию (установка списка регистрируемых событий на создаваемые в каталоге объекты). Параметр всегда должен быть первым. Не может использоваться с параметром -R
-S, --set-file	Установить список регистрируемых событий из файла
-X, --remove-all	Удалить все списки регистрируемых событий
-M, --modify-file	Изменить или добавить элементы списка регистрируемых событий из файла
-B, --restore	Восстановить атрибуты из файла
-R, --recursive	Используется для каталогов. Назначить список регистрации событий рекурсивно
-L, --logical	Следовать по символическим ссылкам
-P, --physical	Не следовать по символическим ссылкам
-h, --help	Вывести справку и выйти
-v, --version	Вывести информацию о версии и выйти

Список регистрируемых событий, а также описание команды приведены в справочной странице `man setfaud`.

Пример

Аудит всех операций (кроме изменения файла) — как успешных, так и не успешных — с файлом `filename`, выполняемых пользователями, для которых правила не заданы явно:

```
setfaud -m o:oxudnarmc:oxudnarmc filename
```

2.11. Пункт «6.4.7. Дополнительные параметры регистрации событий»

Изменить заголовок пункта 6.4.7 и пункт изложить в редакции:

6.4.7. Параметры регистрации событий

При необходимости возможно отключить регистрацию событий в журнал, но правила PARSEC-аудита все равно продолжат обрабатываться и загружать ресурсы ОС.

Отключение регистрации набора системных вызовов возможно выполнить одним из следующих способов:

1) отключение регистрации системных вызовов, не используемых для мандатного управления доступом — выполнить команду:

```
echo 1 > /parsecfs/disable-non-mac-audit
```

Для проверки состояния регистрации выполнить команду:

```
cat /parsecfs/disable-non-mac-audit
```

Если вывод команды равен 1, то регистрация системных вызовов, не используемых для мандатного управления доступом, отключена;

2) отключение регистрации всех системных вызовов — выполнить команду:

```
echo 1 > /parsecfs/disable-all-audit
```

Для проверки состояния регистрации выполнить команду:

```
cat /parsecfs/disable-all-audit
```

Если вывод команды равен 1, то регистрация всех системных вызовов отключена;

3) отключение регистрации запретов доступа (на основе мандатного управления доступом) — выполнить команду:

```
echo 1 > /parsecfs/disable-denied-audit
```

Для проверки состояния регистрации выполнить команду:

```
cat /parsecfs/disable-denied-audit
```

Если вывод команды равен 1, то регистрация запретов доступа отключена.

Для оптимизации работы и уменьшения нагрузки на ОС рекомендуется выполнять отключение регистрации событий с помощью инструмента командной строки `astra-audit-control` (описание приведено в 16.5.34). В таком случае будет отключена как регистрация событий в журнал, так и сама обработка правил PARSEC-аудита.

2.12. Подраздел «12.2. Настройка печати документа с ненулевой классификационной меткой»

В таблице 60 добавить параметры `MacAudit`, `MacEnableFonarik` и `MacSelectFont`:

Таблица 60

Параметр	Значение по умолчанию	Описание
`MacAudit`	`on`	Включает регистрацию событий с помощью библиотеки `libastraeventse`
`MacEnableFonarik`	`on`	Создавать задание для печати регистрационной информации документа («фонарика» на обороте последнего листа документа)
`MacSelectFont`	`off`	Запрашивать у пользователя шрифт для маркировки

2.13. Пункт «12.3.1. Файл описания переменных маркировки»

В таблице 61 добавить атрибут `mac-marker-font`:

Таблица 61

Атрибут	Описание
mac-marker-font	Шрифт для маркировки

В таблице 63 добавить атрибут mac-job-marking-required:

Таблица 63

Атрибут	Описание
mac-job-marking-required	Определяет, требуется ли маркировка задания

2.14. Пункт «16.4. Графический киоск в режиме «Мобильный»»

Подраздел 16.4 изложить в редакции:

16.4. Графический киоск в режиме «Мобильный»

Графический киоск в режиме «Мобильный» ограничивает доступ пользователя к функциям системы и возможность запуска графических приложений. Настройка графического киоска выполняется администратором в конфигурационном файле /etc/mobile-kiosk/mobile-kiosk.conf (если указанные каталог и конфигурационный файл отсутствуют, их необходимо создать). В конфигурационном файле необходимо указать имя учетной записи пользователя, для которого настраивается графический киоск, и в качестве значения параметра Applications перечислить desktop-файлы приложений, которые можно будет запускать указанному пользователю:

```
[<имя_пользователя>]
Applications=<приложение1>,<приложение2>,...
```

Пример

Настройка киоска для пользователя user, разрешающая запуск браузеров «Хромиум» и «Хромиум ГОСТ»:

```
[user]
Applications=chromium,chromium-gost
```

Киоск для указанного пользователя будет включаться автоматически в начале каждой последующей сессии данного пользователя.

В графическом киоска пользователю будут доступны только указанные приложения, а также следующие функции системы:

1) просмотр и удаление уведомлений;
2) строка поиска;
3) изменение громкости звука и яркости экрана;
4) завершение работы.

Остальные приложения и функции системы (в том числе добавление, перемещение и удаление значков, изменение обоев, добавление виджетов) пользователю будут недоступны.

Если в конфигурационном файле /etc/mobile-kiosk/mobile-kiosk.conf в качестве значения параметра Applications указано одно приложение и дополнительно указан параметр SingleMode=true, киоск будет работать в одиночном режиме — указанное приложение будет запускаться автоматически при входе пользователя в сессию и пользователь сможет работать только в данном приложении. При завершении работы приложения будет автоматически завершена пользовательская сессия. Из функций системы пользователю будет доступно только завершение работы.

Пример

Настройка киоска для пользователя user, разрешающая работать в одиночном режиме в приложении «Калькулятор»:

```
[user]
Applications=org.kde.kalk
SingleMode=true
```

2.15. Пункт «16.5.34. Выключение и включение аудита файлов и процессов»

После пункта 16.5.33 ввести новый пункт 16.5.34:

16.5.34. Выключение и включение аудита файлов и процессов

Подсистема аудита может оказывать существенное влияние на производительность ОС. При необходимости возможно отключить PARSEC-аудит файлов и процессов для оптимизации нагрузки на ОС, а также для повышения производительности в целом.

Инструмент командной строки astra-audit-control позволяет выключать и включать правила PARSEC-аудита, располагающиеся в файле /etc/audit/rules.d/10-parsec.rules (описание приведено в 6.1).

Параметры вызова, используемые данным инструментом, приведены в таблице 66.

При выключении PARSEC-аудита файл /etc/audit/rules.d/10-parsec.rules удаляется из каталога /etc/audit/rules.d/, в результате чего данные правила аудита перестают выполняться. Значение в файлах /parsecfs/disable-all-audit, /parsecfs/disable-denied-audit и /parsecfs/disable-non-mac-audit изменяются на 1 (описание параметров приведено в 6.4.7).

При включении PARSEC-аудита файл с правилами 10-parsec.rules, шаблон которого хранится в /usr/lib/parsec/audit/rules.d/, копируется в каталог /etc/audit/rules.d/ и правила начинают выполняться. Значения в файлах /parsecfs/disable-all-audit, /parsecfs/disable-denied-audit и /parsecfs/disable-non-mac-audit изменяются на 0.

Изменения вступают в действие немедленно.

Параметр вызова `is-enabled` выводит информацию о наличии файла `/etc/audit/rules.d/10-parsec.rules`, для которого возможны следующие состояния:

1) `ВКЛЮЧЕНО` — файл с заданными правилами аудита присутствует в каталоге;
2) `ВЫКЛЮЧЕНО` — файл с заданными правилами аудита отсутствует в каталоге.

2.16. Пункт «16.5.35. Выключение и включение сетевого аудита»

После вновь введенного пункта 16.5.34 ввести новый пункт 16.5.35:

16.5.35. Выключение и включение сетевого аудита

Сетевой аудит, как и подсистема аудита в целом, могут оказывать существенное влияние на производительность ОС. При необходимости возможно отключить сетевой PARSEC-аудит, что значительно уменьшит объем журналов регистрации сетевых событий.

Инструмент командной строки `astra-audit-network-control` позволяет выключать и включать постоянные правила сетевого PARSEC-аудита, располагающиеся в файле `/etc/audit/rules.d/10-parsec-nw.rules` (описание приведено в 6.1).

Параметры вызова, используемые данным инструментом, приведены в таблице 66.

При выключении сетевого PARSEC-аудита файл `/etc/audit/rules.d/10-parsec-nw.rules` удаляется из каталога `/etc/audit/rules.d/`, в результате чего правила аудита перестают выполняться.

При включении сетевого PARSEC-аудита файл с правилами `10-parsec-nw.rules`, шаблон которого хранится в `/usr/lib/parsec/audit/rules.d/`, копируется в каталог `/etc/audit/rules.d/` и правила начинают выполняться.

Изменения вступают в действие немедленно.

Параметр вызова `is-enabled` выводит информацию о наличии файла `/etc/audit/rules.d/10-parsec-nw.rules`, для которого возможны следующие состояния:

1) `ВКЛЮЧЕНО` — файл с заданными правилами аудита присутствует в каталоге;
2) `ВЫКЛЮЧЕНО` — файл с заданными правилами аудита отсутствует в каталоге.