TERMIDESK

РУКОВОДСТВО АДМИНИСТРАТОРА

СЛЕТ.10001-01 90 05

Версия 6.0. Выпуск от июня 2025

Настройка компонента «Шлюз»

ОГЛАВЛЕНИЕ

1 . ОБЩИЕ СВЕДЕНИЯ 3
1.1 . О документе 3
1.2 . Назначение компонента «Шлюз» 3
1.3 . Требования к программному и аппаратному обеспечению 3
1.4 . Типографские соглашения 3
2 . УСТАНОВКА И УДАЛЕНИЕ КОМПОНЕНТА 5
2.1 . Получение пакетов установки в ОС Astra Linux Special Edition 5
2.2 . Установка Шлюза 7
2.3 . Установка в режиме замкнутой программной среды 8
2.4 . Удаление Шлюза 9
2.5 . Обновление Шлюза 9
2.6 . Постановка Шлюза на контроль целостности 9
3 . НАСТРОЙКА КОМПОНЕНТА 11
3.1 . Общие сведения по настройке и функционированию Шлюза 11
3.2 . Регистрация компонента в системе управления и мониторинга Termidesk 11
3.3 . Настройка защищенного подключения к Шлюзу 11
3.4 . Параметры конфигурирования компонента 12
3.5 . Статистика подключений 20
3.6 . Журналирование 21
4 . ПРИНЯТЫЕ ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ 22
5 . ПЕРЕЧЕНЬ СОКРАЩЕНИЙ 23

1 . ОБЩИЕ СВЕДЕНИЯ

1.1 . О документе

Настоящий документ является пятой частью руководства администратора на программный комплекс «Диспетчер подключений виртуальных рабочих мест Термидеск» (далее - Termidesk). Документ предназначен для администраторов системы и сети.

В этом руководстве приведено назначение, установка и настройка компонента «Шлюз». Для того, чтобы получить информацию о месте компонента в программном комплексе, необходимо обратиться ко второй части руководства администратора - СЛЕТ.10001-01 90 02 «Руководство администратора. Настройка программного комплекса».

1.2 . Назначение компонента «Шлюз»

Компонент «Шлюз» (далее - Шлюз) входит в состав Termidesk. Шлюз отвечает за туннелирование протоколов доставки, использующих транспортный протокол TCP, обеспечивая отделение инфраструктуры рабочих мест, находящихся во внутренней локальной сети, от внешних локальных сетей или глобальных сетей.

Шлюз является компонентом Termidesk и может устанавливаться как совместно с компонентами «Универсальный диспетчер», «Менеджер рабочих мест», так и отдельно при необходимости обеспечить распределенную конфигурацию.

1.3 . Требования к программному и аппаратному обеспечению

Для установки Шлюза минимальные аппаратные требования узла должны соответствовать следующим:

- процессор архитектуры x86-64 (AMD, Intel);
- оперативная память, не менее 4 ГБ;
- свободное дисковое пространство, не менее 1 ГБ;
- сетевое соединение, не менее 100 Мбит/с.

В среде функционирования Шлюза должна быть предварительно установлена операционная система (ОС):

- Astra Linux Special Edition 1.7 (минимальная версия - 1.7.1);
- Astra Linux Special Edition 1.8 (минимальная версия - 1.8.1).

1.4 . Типографские соглашения

Приняты следующие типографские соглашения:

- моноширинный шрифт – используется для выделения фрагментов текста программ, наименований файлов и папок (директорий), наименований пакетов, путей перемещения, строк комментариев, различных программных элементов (объект, класс, тип, переменная, команда, макрос и т. д.), а также вводимого и выводимого текста в режиме командной строки;
- «кавычки» – текст, заключенный в кавычки, используется для обозначения наименований документов, названий компонентов Termidesk, пунктов меню, наименований окон, вкладок, полей, других элементов графического интерфейса, а также вводимого и выводимого текста в режиме графического интерфейса;
- **[квадратные скобки]** – текст, заключенный в квадратные скобки, используется для наименования экранных кнопок;
- **<угловые скобки>** – текст, заключенный в угловые скобки, используется для наименования клавиш клавиатуры.

2 . УСТАНОВКА И УДАЛЕНИЕ КОМПОНЕНТА

2.1 . Получение пакетов установки в ОС Astra Linux Special Edition

Дистрибутив представлен бинарным файлом пакета ПО в deb-формате. Установка в ОС Astra Linux Special Edition производится из локального репозитория, распространяемого в формате iso-образа.

Получить iso-образ можно двумя способами:

- заполнив запрос через сайт Termidesk: https://termidesk.ru/support/#request-support;
- через личный кабинет: https://lk-new.astralinux.ru/.

Для подключения локального репозитория Termidesk на узле, где предполагается установка, нужно:

- скопировать в домашний каталог пользователя образ диска termidesk-<версия>.iso;
- подключить образ диска к файловой системе в каталог /mnt:

```
sudo mount -o loop termidesk-<версия>.iso /mnt
```

где:

-o loop - параметры для привязки петлевого устройства (/dev/loop) к файлу termidesk-<версия>.iso, устройство затем монтируется в указанный каталог /mnt;

- скопировать содержимое каталога repos подключенного образа диска в каталог /var локальной файловой системы:

```
sudo cp -Rp /mnt/repos /var
```

где:

-Rp - ключ для рекурсивного копирования подкаталогов и файлов с сохранением исходных свойств;

- отключить подключенный ранее образ диска от узла:

```
sudo umount /mnt
```

- установить пакет lsb-release:

```
sudo apt install -y lsb-release
```

где:

-y - ключ для пропуска подтверждения установки;

- добавить локальный репозиторий Termidesk (`/var/repos/astra`) в файл `/etc/apt/sources.list.d/termidesk_local.list` через командный интерпретатор `sh`:

```
sudo sh -c 'echo "deb file:/var/repos/astra $(lsb_release -cs) non-free" > /etc/apt/sources.list.d/termidesk_local.list'
```

где:

`-c` - ключ для чтения команд из вводимой строки (стандартный ввод);

`echo` - команда вывода текста, совместно с символом «>» используется для перенаправления строки `deb file:/var/repos/astra $(lsb_release -cs) non-free` в файл `/etc/apt/sources.list.d/termidesk_local.list`;

`deb file:/var/repos/astra $(lsb_release -cs) non-free` - добавляемый репозиторий, вложенная команда `$(lsb_release -cs)` подставляет версию - `1.8_x86-64`;

- выполнить поиск ключа репозитория Termidesk `GPG-KEY-PUBLIC` и добавить его:

```
sudo cat /var/repos/astra/GPG-KEY-PUBLIC | sudo gpg --dearmour -o /etc/apt/trusted.gpg.d/termidesk.gpg
```

- убедиться, что ключ `release@uveon.ru` был успешно добавлен:

```
apt-key list
```

В случае, если ключ не отображен в выводе команды, необходимо убедиться, что ключ `GPG-KEY-PUBLIC` существует:

```
cat /var/repos/astra/GPG-KEY-PUBLIC
```

Если ключ все же существует, необходимо проверить правильность выполнения шагов по добавлению репозитория Termidesk в файл `/etc/apt/sources.list.d/termidesk_local.list`.
При успешном выполнении всех шагов команда выведет содержимое ключа в формате Base64.
В зависимости от используемой версии ОС команды добавления ключа репозитория Termidesk могут отличаться. Если для ОС Astra Linux Special Edition 1.8 ключ репозитория был добавлен командой, применимой к другой версии ОС, то для исправления нужно:

- узнать идентификатор добавленного ключа репозитория:

```
apt-key list
```

- удалить ключ:

```
apt-key --keyring /etc/apt/trusted.gpg del 4E1632CC
```

где `4E1632CC` - последние 8 символов идентификатора ключа;

- вновь добавить ключ репозитория.

- обновить данные пакетного менеджера:

```
sudo apt update
```

Данную команду (`sudo apt update`) необходимо выполнять при каждом изменении списка источников пакетов или при изменении содержимого этих источников.

2.2 . Установка Шлюза

Для установки Шлюза из подключенного репозитория Termidesk (см. подраздел **Получение пакетов установки в ОС Astra Linux Special Edition**) используется команда:

```
sudo apt install termidesk-gateway
```

> Для обслуживания API-запросов по состоянию Шлюза необходимо после установки Шлюза задать параметр `urlCheckToken`.
> Для регистрации Шлюза в системе управления и мониторинга Termidesk и отслеживания его актуального статуса из портала администратора Termidesk, нужно после установки Шлюза задать параметры
> `coordinatorUrl`,
> `coordinatorUser`,
> `coordinatorPass` (см. подраздел **Параметры конфигурирования компонента**).

Зависимости пакета `termidesk-gateway`:

- `libc6 (>= 2.14)`;
- `libgcc1 (>= 1:3.3.1)`;
- `libssl1.1 (>= 1.1.0)`;
- `libstdc++6 (>= 5.2)`.

Проверка состояния службы `termidesk-gateway` выполняется командой:

```
sudo systemctl status termidesk-gateway
```

Строка «`Active`» отображает состояние сервиса, где статус «`active (running)`» свидетельствует об успешном запуске `termidesk-gateway`.

Для просмотра установленной версии Шлюза `termidesk-gateway` выполнить:

```
termidesk-gateway -v
```

Начиная с Termidesk версии 5.0 изменился API-запрос валидации токена Шлюза `termidesk-gateway` для подключения к компоненту «Универсальный диспетчер».
При обновлении компонента «Универсальный диспетчер» и/или Шлюза `termidesk-gateway` на версию из состава программного комплекса 5.X нужно вручную обновить параметр `urlCheckToken` (см. подраздел **Параметры конфигурирования компонента**) для корректной работы.

2.3 . Установка в режиме замкнутой программной среды

Замкнутая программная среда (ЗПС) является средством повышения безопасности ОС путем контроля целостности (неизменности) файлов. Механизм контроля реализован в виде невыгружаемого модуля ядра ОС Astra Linux Special Edition (модуль `digsig_verif`), выполняющего проверку электронной цифровой подписи файлов (ЭЦП).

Перед установкой компонента Termidesk необходимо установить пакет `termidesk-digsig-keys`, выполнив следующее:

- подключить репозиторий Termidesk или получить отдельный пакет `termidesk-digsig-keys` из репозитория;
- выполнить установку с использованием репозитория:

```
sudo apt -y install termidesk-digsig-keys
```

- либо выполнить установку из deb-пакета:

```
sudo apt install -y /home/user/termidesk-digsig-keys-XXXXXX_amd64.deb
```

где:
`-y` - ключ для пропуска подтверждения установки;
`/home/user/termidesk-digsig-keys-XXXXXX_amd64.deb` - расположение пакета `termidesk-digsig-keys-XXXXXX_amd64.deb`.

- перезагрузить ОС:

```
sudo reboot
```

- выполнить установку компонента Termidesk.

Для ЗПС может быть активирован режим проверки встроенной ЭЦП в расширенных атрибутах (`DIGSIG_XATTR_MODE`). В этом случае потребуется подписать файлы, которые будут проходить проверку, имеющимися в организации ключами. Информация о процессе подписи и активации механизма проверки встроенной ЭЦП в расширенных атрибутах приведена в справочном центре Astra Linux: https://wiki.astralinux.ru/pages/viewpage.action?pageId=41190634.

2.4 . Удаление Шлюза

Для удаления Шлюза `termidesk-gateway` нужно:

- удалить без подтверждения `termidesk-gateway`:

```
sudo aptitude purge -y termidesk-gateway
```

- очистить оставшиеся зависимости и конфигурации:

```
sudo aptitude purge ~c -y
```

2.5 . Обновление Шлюза

Обновление Шлюза `termidesk-gateway` выполняется процедурой установки новой версии. При обновлении `termidesk-gateway` файл `/lib/systemd/system/termidesk-gateway.service` будет перезаписан. Конфигурационный файл Шлюза `/etc/termidesk/termidesk-gateway.conf` будет заменен файлом `/etc/termidesk/gateway.yaml`.

> Начиная с Termidesk версии 5.0 изменился API-запрос валидации токена Шлюза `termidesk-gateway` для подключения к компоненту «Универсальный диспетчер».
> При обновлении компонента «Универсальный диспетчер» и/или Шлюза `termidesk-gateway` на версию из состава программного комплекса 5.X нужно вручную обновить параметр `urlCheckToken` (см. подраздел **Параметры конфигурирования компонента**).

При обновлении распределенной конфигурации необходимо учесть, что если ранее на узлах со Шлюзом был установлен `termidesk-gateway`, необходимо сначала обновить эти узлы, и только потом - узлы компонента «Универсальный диспетчер» и узлы компонента «Менеджер рабочих мест».

2.6 . Постановка Шлюза на контроль целостности

После установки и настройки Шлюза необходимо поставить его на контроль целостности.
Для контроля целостности используются встроенные в ОС Astra Linux Special Edition программные средства на основе `Another File Integrity Checker`, представленного пакетом `afick`.
Настройка регламентного контроля целостности выполняется в конфигурационном файле `/etc/afick.conf`.

> Настройку следует производить только после окончательного внедрения компонента, поскольку с файлов конфигурации и исполняемых файлов будут сняты эталонные контрольные суммы.

Для постановки на контроль целостности Шлюза необходимо добавить в конфигурационный файл `/etc/afick.conf` следующие строки:

```
/etc/systemd/system/multi-user.target.wants/termidesk-gateway.service PARSEC
/etc/termidesk PARSEC
/usr/bin/termidesk-gateway PARSEC
/usr/lib/systemd/system/termidesk-gateway.service PARSEC
/usr/share/doc/termidesk-gateway PARSEC
/var/lib/systemd/deb-systemd-helper-enabled/multi-user.target.wants/termidesk-gateway.service PARSEC
/var/lib/systemd/deb-systemd-helper-enabled/termidesk-gateway.service.dsh-also PARSEC
/etc/systemd/system/multi-user.target.wants/termidesk-gateway.service PARSEC


#Репозиторий Termidesk
/var/repos/ PARSEC
```

Для снятия эталонных значений контрольных сумм используется команда:

```
afick -i
```

Для проверки соответствия контрольных сумм эталонным значениям автоматически создаются задания в системном планировщике заданий `cron`.

Для ручной проверки соответствия контрольных сумм используется команда:

```
afick -k
```

> ⚠ В случае отсутствия по какой-либо причине исполняемых фалов `*.pyc`, они будут повторно сгенерированы при перезапуске сервера Python. В этом случае нужно повторно проинициализировать средство регламентного контроля целостности `afick`.

3 . НАСТРОЙКА КОМПОНЕНТА

3.1 . Общие сведения по настройке и функционированию Шлюза

Для проверки состояния службы Шлюза используется команда:

```
sudo systemctl status termidesk-gateway
```

Шлюз может быть вынесен в демилитаризованную зону сетевой инфраструктуры организации.
Для подключения пользователей к рабочим местам (РМ) через Шлюз необходимо обеспечить доступность сетевых портов, заданных в конфигурационном файле `gateway.yaml` (см. подраздел **Параметры конфигурирования компонента**).
Шлюз `termidesk-gateway` принимает запросы на подключение и направляет их на поставщик ресурсов или в виртуальную машину (ВМ), исключая взаимодействие с веб-сервером.

3.2 . Регистрация компонента в системе управления и мониторинга Termidesk

Для централизованного управления объектами инфраструктуры в веб-интерфейсе Termidesk реализована система управления и мониторинга состояния компонентов.
Регистрация компонента в системе происходит через подключение к серверу RabbitMQ, который хранит информацию об узле компонента и передает ее компоненту «Менеджер рабочих мест». Для регистрации Шлюза в системе необходимо задать обязательные параметры `coordinatorUrl`, `coordinatorUser`, `coordinatorPass` (см. подраздел **Параметры конфигурирования компонента**).
Если параметр `coordinatorUrl` задан, то после запуска Шлюз инициирует подключение к серверу RabbitMQ. Все запросы к этому серверу ожидают ответа в течение времени, заданного в параметре `coordinatorTimeout`, по истечении которого фиксируется ошибка подключения. При успешном подключении Шлюз передает свой статус с URL проверки состояния (`healthcheck`) в формате `JSON` и периодически обновляет его. Таймаут обновления статуса задается в параметре `coordinatorRefreshTime`.

3.3 . Настройка защищенного подключения к Шлюзу

Для настройки защищенного подключения к Шлюзу нужно:

- скопировать закрытый ключ формата `.key` и сертификат формата `.pem` в каталог `/etc/termidesk`:

```
1 sudo cp /etc/ssl/private/ssl-cert-snakeoil.key /etc/termidesk/
2 sudo cp /etc/ssl/certs/ssl-cert-snakeoil.pem /etc/termidesk/
```

- назначить права по использованию ключа и сертификата пользователю `termidesk-gateway`:

```
sudo chown termidesk-gateway:termidesk-gateway /etc/termidesk/ssl-cert-snakeoil.*
```

- предоставить права на чтение ключа и сертификата:

```
sudo chmod 640 /etc/termidesk/ssl-cert-snakeoil.key
sudo chmod 644 /etc/termidesk/ssl-cert-snakeoil.pem
```

- отредактировать файл `/etc/termidesk/gateway.yaml`, указав пути к файлам ключа и сертификата в параметрах `key` и `cert`:

```
_val0: &sslprof0
  key: ${sslKey:/etc/termidesk/ssl-cert-snakeoil.key}
  cert: ${sslCert:/etc/termidesk/ssl-cert-snakeoil.pem}
```

Подробное описание файла `/etc/termidesk/gateway.yaml` содержится в подразделе **Параметры конфигурирования компонента**.

- выполнить перезагрузку службы:

```
sudo systemctl restart termidesk-gateway.service
```

3.4 . Параметры конфигурирования компонента

Параметры конфигурирования Шлюза задаются в конфигурационном файле `/etc/termidesk/gateway.yaml`.

При обновлении компонента «Универсальный диспетчер» и/или Шлюза `termidesk-gateway` на версию из состава программного комплекса 5.X нужно обновить значение для параметра `urlCheckToken`.

Реализована обратная совместимость с прошлым файлом конфигурации: файл `/etc/termidesk/gateway.yaml` реализует подстановку значений параметров из переменных окружения, которые были заданы в `/etc/termidesk/termidesk-gateway.conf`.

Для задания параметров конфигурирования Шлюза из файла нужно отредактировать файл `/etc/termidesk/gateway.yaml`, указав необходимые значения для параметров.

В примере файла значения параметров `user` и `pass` преобразованы с помощью утилиты `gwpasswd`, которая входит в состав компонента.

Пример файла `/etc/termidesk/gateway.yaml`:

```
_val0: &sslprof0
  key: ${sslKey:/etc/termidesk/ssl-cert-snakeoil.key}
  cert: ${sslCert:/etc/termidesk/ssl-cert-snakeoil.pem}
  ca: ${sslCa}
  passphrase: ${sslPassPhrase}
  dhparams: ${sslDhParams}
```

```
  ciphers: ${sslCiphers}

gwservers:
- listen: ${wsServerIP:0.0.0.0}:${wsServerPort:5099}
  websocket:
    pingtimeout: ${wsIdleTimeout:30}
  checktoken:
    url: ${urlCheckToken:https://disp.termidesk.local/api/wsproxy/v5.1/verify}
  tcp_downstream:
    reconnect: ${gwTCPReconnect:0}

- listen: ${wssServerIP:0.0.0.0}:${wssServerPort:10000}
  websocket:
    pingtimeout: ${wsIdleTimeout:30}
  checktoken:
    url: ${urlCheckToken:https://disp.termidesk.local/api/wsproxy/v5.1/verify}
  tcp_downstream:
    reconnect: ${gwTCPReconnect:0}
  ssl: *sslprof0

mgtserver:
  listen: ${mgtServerIP:0.0.0.0}:${mgtServerPort:8102}
  path: ${healthCheckURL:/api/health}
  token: ${healthCheckAccessKey}
  ssl: *sslprof0
  metrics:
    path: ${metricsCheckURL:/api/health/metrics}
    token: ${metricsAccessKey}

rabbitmq:
  url: ${coordinatorUrl:amqp://USER:PASS@disp.termidesk.local:5672/termidesk}
  user: ${coordinatorUser:AES256:F406D42BBA34B79A02AF73FFD1713A82}
  pass: ${coordinatorPass:AES256:9E0A2FBFE5B03E31C8BC717C437D82E8}
  timeout: ${coordinatorTimeout}
  refreshtime: ${coordinatorRefreshTime:60}
  single: ${coordinatorSingle:true}
  exchange: ${coordinatorExchange}
  routingkey: ${coordinatorRoutingKey}

loglevel:
  info: ${logInfo:true}
  debug: ${logDebug:false}
```

Чтобы получить преобразованное значение параметра с помощью утилиты `gwpasswd`, нужно:

- выполнить команду:

```
sudo gwpasswd -e <значение>
```

где:

`-e` - ключ для запуска преобразования переданной строки;

`<значение>` - строка, значение которой будет преобразовано;

Существует также второй способ передачи строки на вход утилите gpasswd:

```
echo <значение> | gwpasswd -e -
```

- преобразованное значение будет отображено в интерфейсе командной строки. Полученное значение необходимо внести в соответствующий параметр конфигурационного файла с приставкой AES256:. Значение допускается вносить как в ранее существующий файл /etc/termidesk/termidesk-gateway.conf, так и в /etc/termidesk/gateway.yaml. В приведенном выше примере значение было подставлено в параметр ${coordinatorUser}:

```
1 rabbitmq:
2   url: ${coordinatorUrl:amqp://USER:PASS@disp.termidesk.local:5672/termidesk}
3   user: ${coordinatorUser:AES256:F406D42BBA34B79A02AF73FFD1713A82}
```

- выполнить перезагрузку службы:

```
sudo systemctl restart termidesk-gateway.service
```

Параметры, используемые в файле /etc/termidesk/gateway.yaml после установки, приведены в таблице (см. Таблица 1).

Задание параметров может производиться и переменными окружения в формате: ${ENV} или ${ENV:DEFAULT}, где DEFAULT - значение для подстановки, если указанная переменная окружения не найдена.

Если переменная окружения существует с пустым значением, то пустое значение будет подставлено в /etc/termidesk/gateway.yaml вместо переменной, а не значение по умолчанию.

Например, если нужно задать через переменные окружения адрес IP_ADDR и порт прослушивания службы PORT, то в конфигурационном файле /etc/termidesk/ gateway.yaml можно изменить значение следующим образом:

```
1 listen: ${IP_ADDR}:${PORT:8000}
```

Либо эти значения можно задать напрямую в файле, без использования переменных окружения, тогда запись изменится к виду:

```
1 listen: "127.0.0.1:8181"
```

Таблица 1 – Доступные переменные Шлюза

Параметр	Описание
Секция val0: &sslprof0	

Параметр	Описание

`_val0: &sslprof0`	Переменная `&sslprof0` преобразует значения параметров защищенного подключения в шаблон, который затем может быть использован в других частях файла `gateway.yaml`.
	Для использования шаблона в соответствующей секции файла следует создать параметр `ssl:*sslprof0`.
	Значение `*sslprof0` является ссылкой на шаблон и подставляет в соответствующей секции параметры защищенного подключения, заданные в шаблоне переменной `&sslprof0`.
	Пример:
	`_val0: &sslprof0`
	`  key: ${sslKey:/etc/termidesk/ssl-cert-snakeoil.key}`
	`  cert: ${sslCert:/etc/termidesk/ssl-cert-snakeoil.pem}`
	
	`mgtserver:`
	`  ssl: *sslprof0`
	`  listen: ${mgtServerIP:0.0.0.0}:${mgtServerPort:8102}`
	`  path: ${healthCheckURL:/api/health}`
	`  token: ${healthCheckAccessKey}`

`key: ${sslKey:/etc/termidesk/ssl-cert-snakeoil.key}`	Путь к файлу ключа для соединения SSL. Параметр не имеет значения по умолчанию.
	При использовании сертификатов и ключей на файлы `.key` и `.pem` необходимо выдать права на чтение командой:
	`sudo chmod 640 /etc/termidesk/ssl-cert-snakeoil.key`
	`sudo chmod 644 /etc/termidesk/ssl-cert-snakeoil.pem`

`cert: ${sslCert:/etc/termidesk/ssl-cert-snakeoil.pem}`	Путь к файлу сертификата для соединения SSL. Параметр не имеет значения по умолчанию
`ca: ${sslCa}`	Путь к корневому сертификату, на котором выпущен сертификат, указанный в параметре `cert`. Параметр не имеет значения по умолчанию. Используется для проверки подлинности указанного в параметре `cert` сертификата
`passphrase: ${sslPassPhrase}`	Пароль для пары «сертификат + ключ». Параметр не имеет значения по умолчанию
`dhparams: ${sslDhParams}`	Путь к файлу с ключами Диффи-Хеллмана для соединения SSL. Параметр не имеет значения по умолчанию
`ciphers: ${sslCiphers}`	Используемый алгоритм преобразований для соединения SSL. Параметр не имеет значения по умолчанию. Список поддерживаемых алгоритмов определяется SSL-библиотекой, установленной в ОС
Секция `gwservers:`	

Параметр	Описание

`listen: ${wsServerIP:0.0.0.0}:${wsServerPort:5099}`	Назначение параметров узла Шлюза:
	▪ `wsServerIP` - адрес прослушивания входящих подключений;
	▪ `wsServerPort` - порт прослушивания службы для входящих подключений.
	Значения по умолчанию:
	▪ `wsServerIP:0.0.0.0`;
	▪ `wsServerPort:5099`.
	Для протокола SSL нужно использовать параметры `wssServerIP` и `wssServerPort` и значение 10000 для номера порта:
	`listen:${wssServerIP:0.0.0.0}:${wssServerPort:10000}`
	ⓘ При использовании протокола SSL обязательно должна быть настроена секция `_val0: &sslprof0`.
	Порт прослушивания службы может быть назначен любой, здесь приведены стандартные значения. Однако в случае, если нужно назначить порт из диапазона общеизвестных TCP-портов 1 - 1023, следует установить исполняемому файлу `/usr/bin/termidesk-gateway` разрешение на их использование:
	`sudo setcap 'cap_net_bind_service=+ep' /usr/bin/termidesk-gateway`

`websocket:` `  pingtimeout: ${wsIdleTimeout:30}`	Таймаут соединения (в секундах) к компоненту «Клиент» Termidesk. Может принимать значения в интервале с 8 до 960, а также 0 (отключен). Значение по умолчанию: «30»

Параметр	Описание
`checktoken:` `url: ${urlCheckToken:https://disp.termidesk.local/api/wsproxy/v5.1/verify}`	Назначение IP-адреса или FQDN узла компонента «Универсальный диспетчер» с ролью «Портал пользователя» для обслуживания API-запросов по валидации подключений, запрашиваемых «Универсальным диспетчером» через Шлюз. При указании FQDN стоит учитывать, что ОС должна иметь возможность разрешить его в IP-адрес. При обновлении компонента «Универсальный диспетчер» и/или Шлюза `termidesk-gateway` на версию из состава программного комплекса 5.X следует использовать значения версий API: ▪ `v1.1`; ▪ или `v5.1`. Значение добавлено в Termidesk версии 5.1. Версия API v1.1 будет удалена в последующих версиях Termidesk. Требуется заранее перейти на использование наиболее новой версии. Пример: `${urlCheckToken:https://disp.termidesk.local/api/wsproxy/v5.1/verify}`, где: `disp.termidesk.local` - FQDN узла компонента «Универсальный диспетчер» с ролью «Портал пользователя». При комплексной установке Termidesk значение `disp.termidesk.local` может быть заменено на `127.0.0.1`. Если программный комплекс Termidesk установлен в распределенном варианте, следует изменить значение `disp.termidesk.local` на внешний IP-адрес или FQDN узла балансировщика нагрузки
`tcp_downstream:` `reconnect: ${gwTCPReconnect:0}`	Количество переподключений Шлюза к ВМ. Может принимать значения в интервале с 0 до 10. Значение по умолчанию:«0»
Секция `mgtserver:`	
`listen: ${mgtServerIP:0.0.0.0}:${mgtServerPort:8102}`	Задание параметров для подключения к API Шлюза: ▪ `mgtServerIP` - IP-адрес или FQDN доступа к API Шлюза. Код 200 в ответе на API-запрос свидетельствует о работоспособности Шлюза. При распределенной установке следует указать значение `0.0.0.0` или явно задать IP-адрес, настроенный на одном из сетевых интерфейсов, для активации приема запросов с внешних систем; ▪ `mgtServerPort` - порт доступа к API Шлюза. Значения по умолчанию: ▪ `mgtServerIP:127.0.0.1`; ▪ `mgtServerPort:8102`
`path: ${healthCheckURL:/api/health}`	Адрес для запросов проверки состояния Шлюза. Значение по умолчанию: `${healthCheckURL:/api/health}`

Параметр	Описание
`token: ${healthCheckAccessKey}`	Параметр для доступа к API Шлюза. Параметр не имеет значения по умолчанию. Значение может быть получено из параметра `HEALTH_CHECK_ACCSESS_KEY`, определенного в конфигурационном файле `/etc/opt/termidesk-vdi/termidesk.conf` узла «Универсального диспетчера». Пример: `${healthCheckAccessKey:270c1e6a4cd013a3824982458a26ec4dcac17f60f80a74098a62994f775351e2}`, где: `270c1e6a4cd013a3824982458a26ec4dcac17f60f80a74098a62994f775351e2` - значение, указанное в параметре `HEALTH_CHECK_ACCSESS_KEY` конфигурационного файла «Универсального диспетчера»
`ssl: *sslprof0`	Настройка SSL
`metrics:` `  path: ${metricsCheckURL:/api/health/metrics}`	Адрес API для запроса метрик узла Шлюза. Значение по умолчанию: `${metricsCheckURL:/api/health/metrics}`
`  token: ${metricsAccessKey}`	Ключ для доступа к API-запроса метрик узла Шлюза. Параметр не имеет значения по умолчанию. При задании значения ключа следует руководствоваться правилом, что: ▪ размер ключа должен составлять от 0 до 64 символа; ▪ должны использоваться символы в шестнадцатеричной системе (0-9, a-f). Пример: `${metricsAccessKey:270c1e6a4cd013a382498}`
Секция `rabbitmq:`	
`url: ${coordinatorUrl:amqp://USER:PASS@disp.termidesk.local:5672/termidesk}`	URL, содержащий IP-адрес (или FQDN) и порт (по умолчанию `5672`) для подключения к серверу RabbitMQ, а также очередь - `termidesk`. Параметр не имеет значения по умолчанию. Используемый формат: `amqp(s)://USER:PASS@<IP-адрес>:<порт>/termidesk`. `USER` и `PASS` необязательные параметры, они могут быть заданы в параметрах `${coordinatorUser}` и `${coordinatorPass}`. Пример: `${coordinatorUrl:amqp://USER:PASS@disp.termidesk.local:5672/termidesk}` . На данный момент поддерживается задание только одного URL для подключения к серверу RabbitMQ. Подключение может использоваться как незащищенное (в этом случае указывается `amqp`), так и защищенное (`amqps`). В случае, если RabbitMQ установлен на том же узле (локально), нужно указать: `${coordinatorUrl: amqp://127.0.0.1:5672/termidesk}`
`user: ${coordinatorUser}`	Имя пользователя для подключения к серверу RabbitMQ. Параметр не имеет значения по умолчанию. Если параметр задан, то он меняет значение `USER` в параметре `${coordinatorUrl}`. При стандартной установке компонента «Универсальный диспетчер» строго по документации в части настроек файла `/etc/rabbitmq/definitions.json` используется `${coordinatorUser:termidesk}`

Параметр	Описание
pass: ${coordinatorPass}	Пароль для подключения серверу RabbitMQ. Параметр не имеет значения по умолчанию. Если параметр задан, то он меняет значение PASS в параметре ${coordinatorUrl}. При стандартной установке компонента «Универсальный диспетчер» строго по документации в части настроек файла /etc/rabbitmq/definitions.json используется ${coordinatorPass:ksedimret}
timeout: ${coordinatorTimeout}	Интервал (в секундах) ожидания ответа от сервера RabbitMQ. Параметр не имеет значения по умолчанию Пример: ${coordinatorTimeout:10}
refreshtime: ${coordinatorRefreshTime:60}	Интервал (в секундах) обновления (переопубликации) регистрационной информации (URL и другие данные) Шлюза. Параметр не имеет значения по умолчанию. Значение «0» отключает автоматическую отправку состояния Шлюза на сервер RabbitMQ. Значение по умолчанию: «60»
single: ${coordinatorSingle:true}	Способ передачи данных. Параметр принимает значения: ▪ true - данные передаются до первого подтверждения; ▪ false - данные передаются по циклу (бесконечно). Значение по умолчанию: true
exchange: ${coordinatorExchange}	Координатор маршрутизации сообщений, определенный в RabbitMQ. Отвечает за маршрутизацию сообщений в разные очереди. Параметр не имеет значения по умолчанию
routingkey: ${coordinatorRoutingKey}	Ключ маршрутизации RabbitMQ, используется для маршрутизации задачи в очереди. Параметр не имеет значения по умолчанию
Секция loglevel:	
info: ${logInfo:true}	Активация режима журналирования уровня «INFO»
debug: ${logDebug:false}	Активация подробного режима журналирования уровня «DEBUG»

Для проверки состояния Шлюза termidesk-gateway через утилиту curl нужно:

- получить значение переменной HEALTH_CHECK_ACCSESS_KEY, определенной в конфигурационном файле /etc/opt/termidesk-vdi/termidesk.conf узла «Универсального диспетчера»;
- на узле Шлюза отредактировать конфигурационный файл, присвоив параметру token: ${healthCheckAccessKey} полученное значение, например:

```
token: ${healthCheckAccessKey:270c1e6a4cd013a3824982458a26ec4dcac17f60f80a74098a62994f775351e2}
```

- выполнить команду проверки состояния Шлюза:

```
curl -v -s -X 'GET' "${HOSTNAME}:8102/api/health" -H 'accept: application/json' -H
"Authorization: Token ${healthCheckAccessKey}" --fail -w "\n%{http_code}\n"
```

Для получения информации по доступным аргументам командной строки нужно выполнить:

```
termidesk-gateway --help
```

3.5 . Статистика подключений

Шлюз позволяет получать статистику использования через запрос `/info` с разными флагами. Статистика может быть выведена в кратком виде, содержащем только общую информацию, и в полном виде, в котором дополнительно отражены подключения «Клиентов».

Доступны следующие флаги, которые можно передать в самом запросе:

- `json` - отобразить статистику в JSON-формате, без использования флага будет выведен простой текст. Если в заголовке запроса присутствует поле «accept: application/json», то это равносильно установке флага `json`;
- `pretty` - отформатировать вывод в удобочитаемый вид для JSON-формата;
- `full` - отобразить полную статистику с информацией по подключенным «Клиентам».

Примеры запросов:

- `https://<IP-адрес Шлюза>:<порт Шлюза>/info` - краткая статистика в текстовом виде;
- `https://<IP-адрес Шлюза>:<порт Шлюза>/info?full` - полная статистика в текстовом виде;
- `https://<IP-адрес Шлюза>:<порт Шлюза>/info?json` - краткая статистика в JSON-формате;
- `https://<IP-адрес Шлюза>:<порт Шлюза>/info?json&pretty&full` - полная статистика в удобочитаемом JSON-формате.

> ⓘ Вся статистика объединена в одну общую структуру и не имеет значения, через какой протокол она вызывается: HTTP или HTTPS.

Пример передачи флага `json` через HTTP-заголовок:

```
curl -X 'GET' -H "Accept: application/json" https://<IP-адрес Шлюза>:<порт Шлюза>/info
```

Что также равносильно запросу:

```
curl -X 'GET' https://<IP-адрес Шлюза>:<порт Шлюза>/info?json
```

Пример вывода полной статистики в удобочитаемом JSON-формате при одном подключенном «Клиенте»:

```
1  {
2    "timeStart": "2024-02-09 16:56:26.509 +0800",// Время запуска прокси-сервера
```

```
3     "allDevice": {                                  // Общая статистика
4       "webUp": {                                  // Веб-сокеты
5           "recvBytes": 867863000,                 // Кол-во принятых байт
6           "sendBytes": 867834000,                 // Кол-во переданных байт
7           "recvFrames": 867863,                   // Кол-во принятых фреймов
8           "sendFrames": 39008                     // Кол-во переданных фреймов
9       },
10      "tcpDown": {                                // TCP-сокеты
11          "recvBytes": 867863000,                 // Кол-во принятых байт
12          "sendBytes": 867863000                  // Кол-во переданных байт
13      },
14      "udpDown": {                                // UDP-сокеты
15          "recvBytes": 0,
16          "sendBytes": 0,
17          "recvFrames": 0,
18          "sendFrames": 0
19      }
20  },
21  "numClients": 1,                                // Кол-во подключенных «Клиентов»
22  "clients":
    [                                   // Инфо о подключенных «Клиентах»
23
    {                                           // Инфо по каждому «Клиенту»
24          "timeConnect": "2024-02-09 16:58:12.578 +0800", // Время подключения
25          "ipUp": "127.0.0.1:36778",                  // IP-адрес и порт
26          "ipDown": "127.0.0.1:9080",                 // IP-адрес и порт
27          "webUp":
    {                                   // Веб-сокет
28              "recvBytes": 867863000,                 // Кол-во принятых байт
29              "sendBytes": 867834000,                 // Кол-во переданных байт
30              "recvFrames": 867863,                   // Кол-во принятых фреймов
31              "sendFrames": 39008                     // Кол-во переданных фреймов
32          },
33          "tcpDown": {                                // TCP Сокет
34              "recvBytes": 867863000,                 // Кол-во принятых байт
35              "sendBytes": 867863000                  // Кол-во переданных байт
36          }
37      }
38   ]
39  }
```

3.6 . Журналирование

Для просмотра журнала Шлюза `termidesk-gateway` можно выполнить:

```
sudo journalctl -f -u termidesk-gateway.service
```

или:

```
sudo less /var/log/syslog
```

4 . ПРИНЯТЫЕ ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Термин	Определение
Балансировщик нагрузки	Самостоятельный компонент, отвечающий за распределение нагрузки на множество «Универсальных диспетчеров» и «Шлюзов»
Гостевая ОС	ОС, функционирующая на ВМ
Компонент «Менеджер рабочих мест»	Компонент Termidesk. Отделяемый компонент программного комплекса, отвечающий за взаимодействие с поставщиком ресурсов и управления жизненным циклом РМ, включая создание, настройку, запуск, отключение и удаление. Является обработчиком фоновых задач. Устанавливается из пакета `termidesk-vdi`. Наименование служб после установки: `termidesk-celery-beat.service` и `termidesk-celery-worker.service`
Поставщик ресурсов	ОС, платформа виртуализации или терминальный сервер (MS RDS/STAL), предоставляющие вычислительные мощности, ресурсы хранения данных, а также сетевые ресурсы для размещения фондов РМ
Протокол доставки	Поддерживаемый в Termidesk протокол удаленного доступа к РМ
Компонент «Универсальный диспетчер»	Компонент Termidesk. Отделяемый компонент программного комплекса, отвечающий за идентификацию пользователей, назначение им РМ и контроль доставки РМ. Устанавливается из пакета `termidesk-vdi`. Наименование службы после установки: `termidesk-vdi.service`
Компонент «Шлюз»	Компонент Termidesk. Самостоятельный компонент, отвечающий за туннелирование протоколов доставки, использующих транспортный протокол TCP. Устанавливается из пакета `termidesk-gateway`. Наименование службы после установки: `termidesk-gateway.service`.
Ключ	Применяется в контексте файла, не опции в команде. Последовательность псевдослучайных чисел, сгенерированная особым образом
Сертификат	Артефакт, содержащий информацию о владельце ключа и подтверждающий принадлежность ключа владельцу

5 . ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

Сокращение	Пояснение
ВМ	Виртуальная машина
ЗПС	Замкнутая программная среда
ОС	Операционная система
РМ	Рабочее место
ЭЦП	Электронная цифровая подпись
API	Application Programming Interface (интерфейс прикладного программирования)
FQDN	Fully Qualified Domain Name (полностью определенное имя домена)
HTTP	HyperText Transfer Protocol (протокол передачи гипертекста)
HTTPS	Hypertext Transfer Protocol Secure (расширение протокола HTTP для поддержки шифрования)
IP	Internet Protocol (межсетевой протокол)
JSON	JavaScript Object Notation (стандартный текстовый формат для структурированных данных)
MS RDS	Microsoft Remote Desktop Services (службы удаленного рабочего стола Microsoft)
SSL	Secure Sockets Layer (криптографический протокол)
STAL	Terminal Server Astra Linux (сервер терминалов Astra Linux)
TCP	Transmission Control Protocol (протокол управления передачей данных)
UDP	User Datagram Protocol (протокол пользовательских датаграмм)
URL	Uniform Resource Locator (унифицированный указатель ресурса)

UVEON
Unified Virtualization
Experience, Opportunities, Now

© ООО «УВЕОН»

119571, г. Москва, Ленинский проспект,
д. 119А, помещ. 9Н
https://termidesk.ru/
Телефон: +7 (495) 975-1-975

Общий e-mail: info@uveon.ru
Отдел продаж: sales@uveon.ru
Техническая поддержка: support@uveon.ru