

# Операции virt-sysprep

Команда `virt-sysprep` удаляет сведения, касающиеся конкретной системы и подготавливает образ ВМ для создания шаблона.

Часть из этих операций выполняется при запечатывании шаблона Linux.



Не нужно выполнять эти операции вручную.

Таблица 1. Список возможных операций над образом виртуальной машины

Операция	Описание	Операция выполняется по умолчанию?
<code>abrt-data</code>	Удалить данные о сбое, сгенерированные ABRT	Да
<code>backup-files</code>	Удалить файлы резервных копий редактора из гостевой ОС	Да
<code>bash-history</code>	Удалить файлы резервных копий редактора из гостевой ОС	Да
<code>blkid-tab</code>	Удалить данные blkid в гостевой ОС	Да
<code>ca-certificates</code>	Удалить сертификаты ЦС в гостевой ОС	Нет
<code>crash-data</code>	Удалить данные о сбое, сгенерированные kexec-tools	Да
<code>cron-spool</code>	Удалить данные о сбое, сгенерированные kexec-tools	Да
<code>customize</code>	Настроить гостевую ОС	Да
<code>dhcp-client-state</code>	Удалить аренду DHCP-клиента	Да
<code>dhcp-server-state</code>	Удалить аренду DHCP-сервера	Да
<code>dovecot-data</code>	Удалить данные Dovecot (почтового сервера)	Да
<code>firewall-rules</code>	Удалить правила брандмауэра	Нет
<code>flag-reconfiguration</code>	Пометить систему на перенастройку	Нет
<code>fs-uuids</code>	Изменить UUID файловой системы	Нет
<code>ipa-client</code>	Удалить файлы IPA	Да

<b>Операция</b>	<b>Описание</b>	<b>Операция выполняется по умолчанию?</b>
<b>kerberos-data</b>	Удалить данные Kerberos в гостевой ОС	Нет
<b>kerberos-hostkeytab</b>	Удалите файл keytab хоста Kerberos в гостевой ОС	Да
<b>logfiles</b>	Удалить файлы системы журналирования событий из гостевой ОС	Да
<b>lvm-uuids</b>	Изменить LVM2 PV и VG UUID	Да
<b>machine-id</b>	Удалить идентификатор локальной машины	Да
<b>mail-spool</b>	Удаление электронной почты из локального почтового каталога	Да
<b>net-hostname</b>	Удалить HOSTNAME и DHCP_HOSTNAME в конфигурации сетевого интерфейса	Да
<b>net-hwaddr</b>	Удалить конфигурацию HWADDR (жестко запрограммированный MAC-адрес)	Да
<b>pacct-log</b>	Удалить файлы журнала учета процессов	Да
<b>package-manager-cache</b>	Удалить кеш менеджера пакетов	Да
<b>pam-data</b>	Удалить данные PAM в гостевой	Да
<b>passwd-backups</b>	Удалить /etc/passwd- и подобные файлы резервных копий	Да
<b>puppet-data-log</b>	Удалить данные и файлы журналов puppet	Да
<b>rh-subscription-manager</b>	Удалите файлы менеджера подписки RH	Да
<b>rhn-systemid</b>	Удалить идентификатор системы RHN	Да
<b>rpm-db</b>	Удалить специфичные для хоста файлы базы данных RPM	Да
<b>samba-db-log</b>	Удалить базу данных и лог-файлы Samba	Да
<b>script</b>	Запускать произвольные скрипты в гостевой ОС	Да
<b>smolt-uuid</b>	Удалите аппаратный UUID Smolt	Да
<b>ssh-hostkeys</b>	Удалите ключи хоста SSH в гостевой	Да
<b>ssh-userdir</b>	Удалить каталоги ".ssh" в гостевой	Да
<b>sssd-db-log</b>	Удалите файлы базы данных и журналов sssd	Да

<b>Операция</b>	<b>Описание</b>	<b>Операция выполняется по умолчанию?</b>
<b>tmp-files</b>	Удалить временные файлы	Да
<b>udev-persistent-net</b>	Удалить постоянные сетевые правила udev	Да
<b>user-account</b>	Удалить учетные записи пользователей в гостевой ОС	Нет
<b>utmp</b>	Удалить файл utmp	Да
<b>yum-uuid</b>	Удалить UUID yum	Да

---

2025 orionsoft. Все права защищены.

# Создание виртуальных машин Windows

Для создания виртуальной машины Windows необходимо выполнить следующие действия:

1. Создайте виртуальную машину. Необходимо добавить виртуальный диск для хранилища и сетевой интерфейс для подключения виртуальной машины к сети.
2. Запустите виртуальную машину с использованием параметра "Однократный запуск (Run Once). Подключите ISO-диск гостевых инструментов Windows к виртуальной машине, чтобы установить драйверы VirtIO во время установки операционной системы.
3. Установите операционную систему Windows на виртуальной машине. Ознакомьтесь с инструкциями в документации по операционной системе.
4. Установите гостевые агенты и драйверы для обеспечения дополнительной функциональности виртуальной машины.

После выполнения всех действий новая виртуальная машина будет готова к работе и выполнению задач.

## 1. Создание виртуальной машины

При создании новой виртуальной машины задайте ее настройки. Можно изменить некоторые из этих настроек позднее, включая тип чипсета и BIOS.



Можно изменить длину имени виртуальной машины, заданную по умолчанию, с помощью инструмента `engine-config`. Запустите следующую команду на машине с Менеджером управления:

```
engine-config --set MaxVmNameLength=integer
```



### Порядок действий:

1. Нажмите **Ресурсы (Compute) > Виртуальные машины (Virtual Machines)**.
2. Нажмите **[ Создать (New) ]**. Откроется окно **Новая виртуальная машина (New Virtual Machine)**.
3. Выберите **Операционную систему (Operating System)** в выпадающем списке.
4. Укажите **Имя (Name)** виртуальной машины.
5. Добавьте виртуальный диск к виртуальной машине:
  - Для добавления существующего диска: нажмите **[ Прикрепить (Attach) ]** и выберите существующий виртуальный диск.

- Для создания нового диска: нажмите [ **Создать (Create)** ] и укажите **Размер (GiB) (Size(GiB))** и **Имя (Alias)** для нового виртуального диска. В остальных полях можно принять настройки по умолчанию или изменить их при необходимости. Дополнительные сведения о полях для всех типов дисков см. в разделе [Описание настроек в окнах "Новый диск \(New Virtual Disk\)" и "Изменить виртуальный диск \(Edit Virtual Disk\)".](#)
6. Подключите виртуальную машину к сети. Добавьте сетевой интерфейс, выбрав vNIC-профиль в выпадающем списке *nic1* внизу на вкладке **Общее (General)**.
7. Перейдите во вкладку **Система (System)**. Задайте для виртуальной машины объем оперативной памяти в поде **Оперативная память (разделяемая) (Memory Size)**.
8. На вкладке **Параметры загрузки (Boot Options)** выберите **Первое устройство (First Device)**, которое виртуальная машина будет использовать для начальной загрузки.
9. В остальных полях можно принять настройки по умолчанию или изменить их при необходимости. Дополнительные сведения о всех полях в окне **Новая виртуальная машина (New Virtual Machine)** см. в разделе [Описание настроек в окнах "Создать VM \(New Virtual Machine\)" и "Изменить VM \(Edit Virtual Machine\)".](#)
10. Нажмите [ **OK** ].

Новая виртуальная машина создается и отображается в списке виртуальных машин со статусом **Выключено (Down)**.



Перед использованием этой виртуальной машины обязательно установите операционную систему и драйверы VirtIO.

## 2. Запуск виртуальной машины с использованием параметра "Однократный запуск (Run Once)

### 2.1. Установка Windows на оборудовании VirtIO



Образ ISO с драйверами virtio для Windows, необходимый для выполнения процедур, описанных в этом разделе, можно загрузить в официальном [репозитории Orionsoft](#). Рекомендуем скачивать последнюю доступную версию образа.

Установите диск с драйверами VirtIO устройств во время установки Windows, подключив файл **virtio-win\_version.iso** к виртуальной машине. Драйверы VirtIO обеспечивают увеличение производительности на эмулируемых драйверах устройств.

Используйте **Запустить (Run) > Однократный запуск (Run Once)**, чтобы для разовой загрузки подключить файл **virtio-win\_version.iso**, который отличается от **Параметров**

**загрузки (Boot Options)**, определенных в окне **Новая виртуальная машина (New Virtual Machine)**.

Предварительные условия:

Следующие элементы должны быть добавлены в виртуальную машину:

- сетевой интерфейс **Red Hat VirtIO**.
- диск, использующий интерфейс **VirtIO**.

Можете загрузить **virtio-win\_version.iso** в домен хранения данных.

**i** Образы ISO следует загружать в домен данных через Портал администрирования или с помощью REST API.

Чтобы установить драйверы **virtio-win** при установке Windows, выполните следующие действия:

**Порядок действий:**

1. Нажмите **Ресурсы (Compute) > Виртуальные машины (Virtual Machines)** и выберите виртуальную машину.
2. Нажмите **Запустить (Run) > Однократный запуск (Run Once)**.
3. Разверните меню **Параметры загрузки (Boot Options)**.
4. Установите флажок **Прикрепить CD (Attach CD)** и выберите Windows ISO из выпадающего списка.
5. Установите флажок **Прикрепить CD с гостевыми дополнениями для Windows (Attach Windows guest tools CD)**.
6. Переместите **CD-ROM** в начало списка в поле **Предпочитаемая последовательность загрузки (Predefined Boot Sequence)**.
7. При необходимости настройте другие параметры в окне **Запустить VM (Run Virtual Machine(s))**. Для получения дополнительной информации см. раздел [Описание настроек в окне "Запустить VM \(Run Virtual Machine\(s\)\)"](#).
8. Нажмите **[ OK ]**. Состояние виртуальной машины меняется на **Включено (Up)**, и начинается установка операционной системы.

Откройте консоль виртуальной машины, если не открылась автоматически во время установки Windows.

9. Когда система предложит выбрать диск, на который нужно установить Windows, нажмите **[ Загрузить драйвер (Load driver) ]** и затем **[ OK ]**.
10. В разделе **Выбрать драйвер для установки (Select the driver to install)** выберите драйвер, подходящий для данной версии Windows. Например, для Windows Server 2019 выберите **Red Hat VirtIO SCSI controller (E:\amd64\2k19\viostor.inf)**.

11. Нажмите [Далее (Next)].

Остальная часть процесса установки проходит как обычно.

## 2.2. Открытие консоли виртуальной машины

Для подключения к виртуальной машине используйте инструмент удаленного просмотра (Remote Viewer).



Чтобы другие пользователи могли подключиться к ВМ, обязательно выключите и перезапустите виртуальную машину после завершения работы с консолью. Либо администратор может **Выключить строгую проверку пользователей (Disable strict user checking)**, чтобы исключить необходимость перезагрузки между сессиями пользователей. Для получения дополнительной информации см. [Описание настроек консоли виртуальной машины](#).

### Порядок действий:

1. Установите инструмент удаленного просмотра (Remote Viewer), если еще не установлен.  
См. [Установка компонентов консоли](#).
2. Нажмите **Ресурсы (Compute) > Виртуальные машины (Virtual Machines)** и выберите виртуальную машину.
3. Нажмите **[ Консоль (Console) ]**. По умолчанию браузер предложит загрузить файл с именем **console.vv**. При открытии файла откроется окно консоли виртуальной машины. Браузер можно настроить на автоматическое открытие этих файлов, чтобы нажатие на **[ Консоль (Console) ]** просто открывало консоль.



Открыть файл **console.vv** можно в течение 120 секунд после загрузки. Если прошло больше времени, нажмите **[ Консоль (Console) ]** еще раз.

### Дополнительные ресурсы

- [Настройка параметров консоли](#).

## 3. Установка гостевых агентов и драйверов

### 3.1. Гостевые агенты, инструменты и драйверы

В целях обеспечения надежной работы виртуальных машин, рекомендуем выполнить процедуру установки гостевых дополнений, следуя официальным руководствам от производителей соответствующих операционных систем.





Обратите внимание на следующие важные моменты:

- При поиске установочных пакетов, обязательно используйте официальный репозиторий производителя.
- Для успешной установки и функционирования гостевых дополнений, вам потребуются административные права.

Инструменты и агенты предоставляют информацию для виртуальных машин, включая:

- Использование ресурсов.
- IP-адреса.

Гостевые агенты, инструменты и драйверы распространяются в виде файла с расширением ISO, который можно подключить к виртуальным машинам.

Для включения дополнительных функций на виртуальной машине, установите на ней гостевые агенты и драйверы.

Таблица 1. Гостевые драйверы

Драйвер	Описание
virtio-net	Паравиртуализированный сетевой драйвер обеспечивает повышенную производительность по сравнению с эмулируемыми устройствами rtl.
virtio-block	Паравиртуализированный драйвер жесткого диска обеспечивает повышенную производительность ввода-вывода на эмулируемых устройствах IDE, оптимизируя координацию и связь между виртуальной машиной и гипервизором. Драйвер дополняет программную реализацию virtio-устройства, используемого хостом для выполнения роли аппаратного устройства.
virtio-scsi	Паравиртуализированный драйвер жесткого диска SCSI предлагает похожую функциональность, что и устройство virtio-block, с некоторыми дополнительными улучшениями. В частности, этот драйвер поддерживает добавление сотен устройств и именует устройства согласно стандартной схеме именования устройств SCSI.
virtio-serial	Драйвер virtio-serial обеспечивает поддержку нескольких последовательных портов. Повышенная производительность используется для организации быстрой связи между виртуальной машиной и хостом, на которую не влияют проблемы в сети. Быстрая связь требуется для гостевых агентов и других функций. Например: копирование и вставка через буфер обмена между виртуальной машиной и хостом; журналирование.
virtio-balloon	Драйвер virtio-balloon используется для управления размером памяти, фактически доступной виртуальной машине. Улучшает процесс избыточного выделения памяти.

Драйвер	Описание
qxl	Паравиртуализированный драйвер дисплея разгружает ЦП хоста и повышает производительность за счет снижения использования полосы пропускания сети при большинстве рабочих нагрузок.

Таблица 2. Гостевые агенты и инструменты

Гостевой агент/инструмент	Описание
qemu-guest-agent	Используется вместо ovirt-guest-agent-common на виртуальных машинах Red Hat Enterprise Linux 8. Устанавливается и включается по умолчанию.
spice-agent	<p>SPICE-агент поддерживает несколько мониторов и отвечает за поддержку режима "клиент-мышь", тем самым повышая удобство работы и скорость отклика по сравнению с эмуляцией QEMU. Захват курсора не требуется в режиме "клиент-мышь".</p> <p>SPICE-агент уменьшает использование полосы пропускания сети за счет:</p> <ul style="list-style-type: none"> <li>* снижения уровня отображения, глубины цвета; * отключения обоев; * сглаживания шрифтов и анимации.</li> </ul> <p>SPICE-агент включает дополнительные функции:</p> <ul style="list-style-type: none"> <li>* поддержка буфера обмена, что позволяет вырезать и вставлять текст и изображения между клиентом и виртуальной машиной; * автоматическая настройка гостевого дисплея в соответствии с настройками на стороне клиента.</li> </ul> <p>На виртуальных машинах на Windows, SPICE-агент состоит из vdservice и vdagent.</p>

## 3.2. Установка гостевых агентов, инструментов и драйверов в Windows



Образ ISO с драйверами virtio и гостевыми агентами для Windows, необходимый для выполнения процедур, описанных в этом разделе, можно загрузить в официальном [репозитории Orionsoft](#). Рекомендуем скачивать последнюю доступную версию образа.

Для установки гостевых агентов, инструментов и драйверов на виртуальную машину Windows, выполните следующие действия:

### Порядок действий:

1. Загрузите **virtio-win\_version.iso** в домен хранения данных.
2. Если виртуальная машина работает, то на Портале администрирования или Пользовательском портале кнопкой **Сменить CD (Change CD)** подключите файл **virtio-**

**win\_version.iso** к каждой виртуальной машине. Если виртуальная машина выключена, нажмите [ Однократный запуск (Run Once) ] и подключите файл ISO как CD.

3. Авторизуйтесь на виртуальной машине.
4. Выберите CD-привод, содержащий файл **virtio-win\_version.iso**. Установку можно выполнить из графического интерфейса или из командной строки.
5. Запустите установщик.



Установку гостевых дополнений для Windows Server 2022 и Windows 11 производите с помощью установщика, расположенного в каталоге **win2022-win11**.

**Для установки из графического интерфейса пользователя выполните следующие действия:**

- a. Дважды нажмите на **virtio-win-gt-x64.msi** или **virtio-win-gt-x86.msi**.
- b. Нажмите [ **Далее (Next)** ] на экране приветствия.
- c. Следуйте подсказкам мастера установки.
- d. Когда установка будет завершена, выберите **Да, я хочу перезагрузить компьютер сейчас (Yes, I want to restart my computer now)** и нажмите **Завершить (Finish)**, чтобы применить изменения.
- e. После перезагрузки виртуальной машины откройте CD-ROM, перейдите в каталог **guest-agent** и дважды нажмите на **qemu-ga-x86\_64.msi** или **qemu-ga-i386.msi**, чтобы установить **qemu-ga**, гостевой агент Qemu.

**Для автоматической установки из командной строки выполните следующие действия:**

- a. Откройте командную строку от имени Администратора.
- b. Введите команду **msiexec** :

```
D:\ msiexec /i "PATH_TO_MSI" /qn [/l*v "PATH_TO_LOG"] [/norestart]  
ADDLOCAL=ALL
```

Другие возможные значения для **ADDLOCAL** перечислены ниже.

Например, чтобы запустить установку, когда **virtio-win-gt-x64.msi** находится на диске **D:\**, без сохранения действий в журнале событий, а затем сразу же перезапустить виртуальную машину, введите следующую команду:

```
D:\ msiexec /i "virtio-win-gt-x64.msi" /qn ADDLOCAL=ALL
```

После завершения установки гостевые агенты и драйверы передают информацию об использовании Менеджера управления и позволяют получить доступ к USB-устройствам и

другим функциям.

### 3.2.1. Драйверы и гостевой агент для Windows Server

Для установки драйверов и гостевого агента используйте образ [virtio-win-1.1.1.2](#)

Драйверы устанавливаются вручную из диспетчера устройств из папки **z173** на диске с образом.

Гостевой агент устанавливается из папки **z173\guest-agent**.

Если на ВМ выбран чипсет (тип BIOS) Q35 , то могут быть установлены не все драйверы. В такой ситуации смените чипсет (тип BIOS) на I440FX .

## 3.3. Значения для ADDLOCAL, позволяющие настроить процесс установки virtio-win из командной строки

При установке **virtio-win-gt-x64.msi** или **virtio-win-gt-x32.msi** из командной строки можно установить любой из драйверов или любое сочетание драйверов.

Кроме того, можно установить определенные агенты, но при этом для каждого агента нужно установить соответствующие драйверы.

Параметр ADDLOCAL команды msieexec позволяет указать драйверы или агенты для установки. ADDLOCAL=ALL устанавливает все драйверы и агенты. Другие значения перечислены в следующих таблицах.

Таблица 3. Значения, которые можно указать для ADDLOCAL при установке драйверов

Значение для ADDLOCAL	Имя драйвера	Описание
FE_network_driver	virtio-net	Паравиртуализированный сетевой драйвер обеспечивает повышенную производительность по сравнению с эмулируемыми устройствами rtl.
FE_balloon_driver	virtio-balloon	Управляет объемом памяти, к которой фактически обращается виртуальная машина. Улучшает процесс избыточного выделения памяти (облегчает работу планировщика выделения памяти гипервизору).
_PVPANIC_DRIVER```   □	PVPANIC```   □	Драйвер устройства QEMU pvpanic.
!UFWCFG_DRIVER```   □	QEMUFWCFG```   □	Драйвер устройства QEMU FWCfg.
CISERIAL_DRIVER```   □	QEMUPCISERIAL```   □	Драйвер последовательного PCI-устройства QEMU.

Значение для ADDLOCAL	Имя драйвера	Описание
FE_SPICE_DRIVER```	SPICE DRIVER```	Паравиртуализированный драйвер дисплея, разгружает ЦП хоста и повышает производительность путем сужения полосы пропускания сети на большинстве рабочих нагрузок.
VIOINPUT_DRIVER```	VIOINPUT```	Драйвер ввода VirtIO.
VIORNG_DRIVER```	VIORNG```	Драйвер устройства VirtIO RNG.
VIOSCSI_DRIVER```	VIOSCSI```	Контроллер сквозного доступа VirtIO SCSI.
IOSERIAL_DRIVER```	VIOSERIAL```	Драйвер устройства VirtIO Serial.
VIOSTOR_DRIVER```	VIOSTOR```	Драйвер VirtIO Block.

Таблица 4. Значения, которые можно указать для ADDLOCAL при установке агентов и соответствующих необходимых драйверов

Агент	Описание	Соответствующий драйвер (драйверы)	Значение для ADDLOCAL
Spice Agent	Поддерживает несколько мониторов, отвечает за поддержку режима "клиент-мышь", снижает использование полосы пропускания, обеспечивает поддержку буфера обмена между клиентом и виртуальной машиной, повышает удобство работы и скорость отклика.	VIOSERIAL```    ДРАЙВЕР SPICE	FE_spice_Agent FE_vioserial_driver FE_spice_driver

#### Пример 1. Использование msisexec

Следующая команда устанавливает только контроллер сквозного доступа VirtIO SCSI, драйвер устройства VirtIO Serial и драйвер VirtIO Block:

```
D:\ msisexec /i "virtio-win-gt-x64.msi" /qn
ADDLOCAL=`FE_vioscsi_driver,FE_vioserial_driver,FE_viostor_driver
```

Следующая команда устанавливает только Spice Agent и необходимые драйверы:

```
D:\ msisexec /i "virtio-win-gt-x64.msi" /qn ADDLOCAL =
FE_spice_Agent,FE_vioserial_driver,FE_spice_driver
```

## **Дополнительные ресурсы**

- [Обновление гостевых агентов и драйверов Windows с помощью командной строки.](#)

Веб-сайт разработчика Microsoft:

- [Установщик Windows.](#)
- [Параметры командной строки для установщика Windows.](#)
- [Свойства для установщика Windows.](#)



# Развертывание SDS на базе ПСХД MIND uStor

## 1. Введение

Гиперконвергентный вариант развертывания среды виртуализации объединяет вычислительные возможности, хранилища, сети хостов и предоставляет управление инфраструктурой в веб-интерфейсе.



Перед началом развертывания обязательно ознакомьтесь со следующими документами:

- [Архитектура](#)
- [Рекомендации](#)
- [Системные требования](#)

## 2. Алгоритм развертывания

- Подготовьте серверы.
- [Разверните Менеджер управления](#) в желаемой архитектуре.



В случае развертывания в архитектуре HostedEngine не размещайте диск VM HostedEngine на ресурсах ПСХД MIND uStor.

### 2.1. Подготовка серверов

- Серверы должны быть полностью укомплектованы всеми необходимыми аппаратными компонентами.
- Серверы должны быть скоммутированы в сеть управления и сеть кластерного взаимодействия.
- На серверах должны быть выполнены базовые проверки работоспособности аппаратных компонентов и подтверждена их исправность.
- На серверах должны быть выполнены работы по обновлению микрокодов аппаратных компонентов до рекомендуемого производителем уровня.
- При наличии в составе серверов аппаратного RAID-контроллера и использования его ресурсов для установки ОС этот контроллер должен быть предварительно

сконфигурирован для использования его логических томов в процессе установки поддерживаемой операционной системы.

- На серверах должны быть выполнены следующие настройки BIOS/UEFI:
  - Включён режим загрузки UEFI;
  - Отключён режим Secure Boot;
  - Отключён режим P-state Control;
  - Отключён (или в состоянии C0/C1) режим C-state Control;
  - Настройка Workload Configuration = I/O sensitive;
  - Настройка Power Performance Tuning = BIOS Controls EPB;
  - Настройка ENERGY\_PERF\_BIAS\_CFG mode = Performance.
- На серверах должна быть установлена среда исполнения zVirt Node.
- На всех серверах должны быть настроены репозитории.

### 3. Развёртывание гиперконвергентного решения



Следующие операции должны выполняться на всех серверах, которые будут играть роль узлов кластера ПСХД.

#### Порядок действий:

1. Подключитесь по SSH к серверу с учетной записью **root**.
2. Подключите репозиторий zvirt-ustor:

```
tee -a /etc/yum.repos.d/zvirt-ustor.repo <<EOF
[zvirt-extras-ustor]
username=$zuser
password=$zpasswd
name=zVirt Extras uStor repository
baseurl=https://repo-zvirt.orionsoft.ru/extras-ustor/
enabled = 1
skip_if_unavailable = 1
gpgcheck = 0
EOF
```

BASH | □

3. Настройте доступ в репозиторий:

```
zvirt-credentials.py -u USERNAME -p PASSWORD
```

BASH | □

При этом доступ к репозиторию будет настроен на использование учетных данных, значения которых сохраняются в файлах **/etc/yum/vars/zuser** и **/etc/yum/vars/zpasswd**.

В противном случае потребуется ручное указание параметров `username` и `password` в файле `zvirt-stable.repo`.

4. Если развертывание SDS выполняется в кластере с развернутой BM HostedEngine, активируйте режим глобального обслуживания, выполнив следующую команду на хосте с ролью HostedEngine:

```
hosted-engine --set-maintenance --mode=global
```

BASH | □

5. Выполните обновление qemu и libvirt с поддержкой uStor из zvирт-устор:

- а. Очистите кэш:

```
dnf clean all
```

BASH | □

- б. Очистите блокировку версий

```
dnf versionlock clear
```

BASH | □

- в. Переустановите пакеты:

```
dnf reinstall -y qemu-kvm* libvirt-*
```

BASH | □

6. Установите UStor и зависимости:

```
dnf install -y ustор-*
```

BASH | □

7. Установите `etcd`:

```
dnf install -y etcd
```

BASH | □

8. Создайте файл конфигурации `ustор.conf`:

```
tee -a /etc/ustор/ustор.conf <<EOF
{
  "etcd_address": ["<host1>:2379", "<host2>:2379", "<host3>:2379"], ①
  "osd_network": "<network>" ②
}
EOF
```

BASH | □

① В параметре `etcd_address` необходимо указать актуальный список узлов кластера в формате `<fqdn|ip>:<port>`.

② В параметре `osd_network` необходимо указать CIDR внутренней сети для межузлового взаимодействия. Например, `10.252.12.0/24`

9. Инициализируйте сервисы uStor:

```
/usr/lib/ustor/mon/scripts/make-etcd
```

BASH | ↗

```
systemctl enable --now etcd ustor-mon
```

BASH | ↗

10. Разрешите сервисы **etcd-client**, **etcd-server** и порты **56999-60999/TCP** на Межсетевом экране:

```
firewall-cmd --permanent --add-service=etcd-client  
firewall-cmd --permanent --add-service=etcd-server  
firewall-cmd --permanent --add-port=56999-60999/tcp  
firewall-cmd --reload
```

BASH | ↗

11. Повторите настройку на остальных узлах.

# Руководство по управлению SDS на базе ПСХД MIND uStor

## Аннотация

Гиперконвергентная среда zVirt представляет собой архитектуру вычислительных ресурсов и хранилищ, объединенных в единую платформу. Она предоставляет возможность объединять несколько серверов с локальными устройствами хранения (дисками) в единое хранилище данных, которое можно масштабировать по мере необходимости.

В этом документе объясняется, как выполнять задачи обслуживания, характерные для гиперконвергентной среды zVirt на базе ПСХД MIND uStor.

## 1. Настройка локального хранилища uStor

### 1.1. Создание OSD

Чтобы использовать локальное хранилище серверов, выполняющих роль узлов хранения в кластере ПСХД MIND UStor, на каждом из них необходимо создать или изменить конфигурационный файл с параметрами подключения, а затем выполнить процедуру подготовки локальных накопителей для запуска экземпляров службы OSD.



Перед использованием диска, его wwid необходимо добавить в blacklist multipath:

```
tee -a /etc/multipath/conf.d/ustor_blacklist.conf <<EOF
blacklist {
    wwid "<UUID>" ①
}
EOF
```

BASH | □

① Необходимо указать корректный UUID нужного диска.

UUID диска можно получить командой:

```
ls -l /dev/disk/by-id | grep "sdb$" | awk '{print $9}' | sed 's/scsi-//'
```

BASH | □



Описанные ниже действия необходимо выполнить на всех узлах кластера ПСХД.

## Порядок действий:

- Подключитесь к узлу кластера по SSH.
- Выполните подготовку выбранных локальных накопителей для запуска экземпляров службы OSD:
  - Получите список накопителей с указанием логического пути до каждого с помощью команды `lsblk`:

```
lsblk -d -o NAME,SIZE,TYPE,MODEL,ROTA
```

BASH | □

Пример вывода:

NAME	SIZE	TYPE	MODEL	ROTA
sda	150G	disk	QEMU HARDDISK	1
sdb	200G	disk	QEMU HARDDISK	1
sdc	200G	disk	QEMU HARDDISK	1
sdd	100G	disk	iSCSI Disk	0
sr0	1024M	rom	QEMU DVD-ROM	1

BASH | □



Твердотельные накопители должны иметь свойство `rotat=0`. При использовании виртуальных дисков ядро по умолчанию ставит для них значение `rotat=1`. Для изменения этого значения:

- Создайте/измените файл `/etc/udev/rules.d/50-custom-scheduler.rules`, добавив в него следующую строку:

```
ACTION=="add|change", KERNEL=="sd[a-f]", ATTR{queue/rotational}="0",  
ATTR{queue/scheduler}="none" ①
```

① В `KERNEL=="sd[a-f]"` укажите нужные диски.

- Перезагрузите правила udev:

```
udevadm control --reload-rules  
udevadm trigger
```

BASH | □

- Убедитесь, что изменения применились:

```
lsblk -d -o NAME,SIZE,TYPE,MODEL,ROTA
```

BASH | □

NAME	SIZE	TYPE	MODEL	ROTA
sda	150G	disk	QEMU HARDDISK	0
sdb	200G	disk	QEMU HARDDISK	0
sdd	100G	disk	iSCSI Disk	0
sr0	1024M	rom	QEMU DVD-ROM	1

- Определите по полученному списку, какие из накопителей будут использоваться для организации общего дискового хранилища.

- c. Подготовьте локальные накопители в автоматическом режиме с помощью команды `ustor-disk prepare`:

```
ustor-disk prepare <DEVICE|DEVICE_LIST> ①
```

BASH | ↗

① <DEVICE|DEVICE\_LIST> — список, полученный на предыдущем шаге с помощью команды `lsblk`.

Например:

```
ustor-disk prepare --osd_per_disk 2 /dev/sdb
```

BASH | ↗

- d. Дождитесь окончания работы команды.

- e. Обновите параметры суперблоков OSD и привяжите необходимые порты:

```
ustor-disk update-sb <device> --bind_port <port>
```

BASH | ↗

Например:

```
ustor-disk update-sb /dev/sdb1 --bind_port 57000  
ustor-disk update-sb /dev/sdb2 --bind_port 57001
```

BASH | ↗



Эту операцию необходимо выполнить на всех узлах кластера ПСХД для всех OSD.

- f. Перезапустите таргет `ustor` на всех узлах.

```
systemctl restart ustor.target
```

BASH | ↗

3. Проверьте, что все накопители успешно подготовлены и доступны в кластере, с помощью команды:

```
ustor-cli ls-osd -l
```

BASH | ↗

Пример вывода:

ID	STATE	HOST	TIME	SIZE	USED%	BLOCK_SIZE	REWEIGHT	TAGS
DATA	META	JOURNAL						
1	up	g1.vlab.local	6	99.9 G	0%	128 K	1	ssd,
		200G /dev/sdb1						
2	up	g1.vlab.local	6	99.9 G	0%	128 K	1	ssd,
		200G /dev/sdb2						
3	up	g2.vlab.local	3	99.9 G	0%	128 K	1	ssd,
		200G /dev/sdb1						
4	up	g2.vlab.local	3	99.9 G	0%	128 K	1	ssd,
		200G /dev/sdb2						

```
5 up g3.vlab.local 5 99.9 G 0% 128 K 1 ssd,  
200G /dev/sda1  
6 up g3.vlab.local 5 99.9 G 0% 128 K 1 ssd,  
200G /dev/sda2
```

## 1.2. Удаление OSD

Для удаления OSD используется команда `ustor-disk purge`. Команду необходимо выполнять на том узле кластера ПСХД, на котором размещен удаляемый OSD.

Синтаксис команды:

```
ustor-disk purge <device|device_list>
```

BASH | ↗

Обязательные аргументы:

- `<device|device_list>` - устройство OSD, которое необходимо отключить (отмонтировать). Необходимо указать полный путь до блочного устройства в формате `/dev/XX`. Можно указать список устройств через пробел.

Пример:

```
ustor-disk purge /dev/sdb
```

BASH | ↗

## 2. Проверка статуса кластера

После успешного завершения всех процедур по настройке кластера, для проверки его состояния выполните команду `ustor-cli status` на любом узле кластера ПСХД.

В выводе данной команды будет отображено:

- Суммарное количество установленных экземпляров системных служб: etcd, Monitor, OSD;
- Суммарное количество запущенных на текущий момент экземпляров системных служб;
- Текущий размер конфигурационной базы данных;
- Объём неразмеченного пространства хранения кластера.

Пример вывода команды `ustor-cli status`:

```
cluster:  
etcd: 3 / 3 up, leader g3.vlab.local, 6.3 M database size  
mon: 3 up, master g1.vlab.local  
osd: 6 / 6 up
```

BASH | ↗

```
data:
  raw:    0 B used, 599.6 G / 599.6 G available
  state: 0 B clean
  pools: 0 / 0 active
  pgs:

io:
  client: 0 B/s rd, 0 op/s rd, 0 B/s wr, 0 op/s wr
```

В штатном режиме работы суммарное количество и количество запущенных экземпляров на текущий момент должны быть равны.

## 3. Управление пулами хранения

Для управления пулами хранения используется соответствующая команда утилиты **ustor-cli**. Утилиту можно запустить на любом узле кластера ПСХД.

### 3.1. Создание пула хранения

Для создания пула хранения необходимо воспользоваться командой `ustor-cli create-pool`.

Синтаксис команды:

```
ustor-cli create-pool [общие параметры] <name>
```

BASH | ↗

Обязательные аргументы:

- `<name>` - имя создаваемого пула хранения
- `--scheme <scheme>` - схема защиты данных создаваемого пула хранения.

Может принимать значение `replicated` или `ec`.

Основные параметры:

- `--ec <N>+<K>` - параметр для создания пула со схемой защиты данных Erasure Coding (`ec`).

Аргумент **N+K** даёт возможность передать значения параметров `pg_size` и `parity_chunks`. Сумма значений аргумента равна `pg_size`, а значение **K** из аргумента равно `parity_chunks`.

- `--pg_size <size>` - общее количество дисков для PG этого пула, т. е. количество копий для реплицированных пулов и количество дисков с данными и чётностью для

пулов "ec/xor".

- `--pg_count <count>` - количество PG для создаваемого пула хранения. Для максимальной эффективности распределения пространства хранения на одном OSD должно быть 64-128 PG.

Формула для расчёта оптимального количества PG: общее число OSD пула \* 100 / pg\_size. Полученное значение можно округлить до ближайшей степени 2.

Примеры:

Создание пула хранения с именем "Pool\_Replicated" и схемой защиты "replicated" в кластере из трех узлов:

```
ustor-cli create-pool Pool_Replicated --scheme replicated --pg_size 3 --  
pg_count 64
```

BASH | ↗

Создание пула хранения с именем "Pool\_EC" и схемой защиты "Erasure Coding" в кластере из трех узлов:

```
ustor-cli create-pool Pool_EC --ec 2+1 --pg_count 64
```

BASH | ↗

Проверить состояние пула хранения можно с помощью команды `ustor-cli status`.

Отобразить информацию о созданных пулах хранения можно с помощью команды `ustor-cli ls-pool`.

## 3.2. Изменение пула хранения

Для изменения параметров пула хранения используется команда `ustor-cli modify-pool`.

Синтаксис команды:

```
ustor-cli modify-pool <id|name> [--name <new_name>] [параметры]
```

BASH | ↗

Обязательные аргументы:

- `<id|name>` - идентификатор или имя пула.

Основные параметры:

- `--pg_size <size>` - общее количество дисков для PG этого пула, т. е. количество копий для реплицированных пулов и количество дисков с данными и чётностью для пулов "ec/xor".

- `--pg_count <count>` - количество PG для создаваемого пула хранения. Для максимальной эффективности распределения пространства хранения на одном OSD должно быть 64-128 PG.

Формула для расчёта оптимального количества PG: общее число OSD пула \* 100 / pg\_size. Полученное значение можно округлить до ближайшей степени 2.

- `--parity_chunks <number>` - количество блоков чётности (K) для пулов со схемой защиты данных "ec/xor".

Параметр игнорируется для пулов со схемой защиты данных "replicated".

- `--f | --force` - принудительное создание пула, без проверки параметров пула/OSD.

Примеры:

#### Переименование пула хранения TestPool1

```
ustor-cli modify-pool TestPool1 --name TestPool2
```

BASH | ↗

#### Изменение pg\_size и pg\_count пула хранения TestPool1

```
ustor-cli modify-pool TestPool1 --pg_size 4 --pg_count 128
```

BASH | ↗

## 3.3. Получение информации о пуле хранения

Для получения информации о пуле хранения используется команда `ustor-cli get-pool`.

Синтаксис команды:

```
ustor-cli ls-pool <pool_name> [параметры]
```

BASH | ↗

Обязательные аргументы:

- `<pool_name>` - имя пула.

Основные параметры:

- `--detail` - получить подробную информацию о пуле хранения.
- `--json` - использовать формат JSON в выводе результата выполненной команды.

Пример вывода:

```
ustor-cli ls-pool
```

BASH | ↗

NAME	SCHEME	STATUS	PGS	TOTAL	USED	AVAILABLE	USED%	EFFICIENCY
Pool_Replica	EC 2+1	active	16	399.8 G	0 B	399.8 G	0%	100%



Обратите внимание, что для работы пул должен иметь статус `active`.

## 3.4. Удаление пула хранения

Для удаления пула хранения используется команда `ustor-cli delete-pool`.

Синтаксис команды:

```
ustor-cli rm-pool <pool_name>
```

BASH | ↗

Обязательные аргументы:

- `<pool_name>` - имя удаляемого пула.

## 4. Управление ресурсами хранения

Для управления ресурсами хранения используется соответствующая команда утилиты `ustor-cli`. Утилиту можно запустить на любом узле кластера ПСХД.

### 4.1. Создание ресурса хранения

Для создания ресурса хранения необходимо воспользоваться командой `ustor-cli create-image`.

Синтаксис команды:

```
ustor-cli create-image <name> --pool <pool> --size <size>
```

BASH | ↗

Обязательные аргументы:

- `<name>` - имя ресурса хранения.
- `--pool <pool>` - имя пула хранения.
- `--size <size>` - размер создаваемого ресурса хранения. По умолчанию используются байты, но возможно указание размерности (M, G, T).

Примеры:

Создание ресурса хранения с именем Image\_Test и размером 20 ГБ, в пуле хранения Pool\_Replicated:

```
BASH | utoor-cli create-image --size 20G Image_Test --pool Pool_Replicated
```

Отобразить информацию о созданных ресурсах хранения можно с помощью команды utoor-cli ls-image.

## 4.2. Изменение ресурса хранения

Команда utoor-cli modify-image даёт возможность изменить ряд параметров существующего ресурса хранения. Например, можно изменить размер ресурса (увеличить или уменьшить), переименовать ресурс, изменить режим использования ресурса хранения (включить или выключить режим только для чтения).

Синтаксис команды:

```
BASH | utoor-cli modify-image <name> --new-name <image_name> --size <size> [параметры]
```

Обязательные аргументы:

- <name> - имя ресурса хранения.
- --new-name <image\_name> - новое имя в случае переименования ресурса.
- --size <size> - новый размер в случае увеличения/уменьшения размера ресурса. По умолчанию используются байты, но возможно указание размерности (M, G, T).

Если новый размер меньше старого, лишние данные будут удалены.

Перед уменьшением размера ресурса хранения следует изменить размер файловой системы, если она есть.

Примеры:

Переименование ресурса хранения Image\_Test1

```
BASH | utoor-cli modify-image Image_Test1 --newname Image_Test2
```

Увеличение размера ресурса хранения Image\_Test1

```
BASH | utoor-cli modify-image Image_Test1 --size 20G
```

## 4.3. Получение информации о ресурсе хранения

Для получения информации о ресурсе хранения используется команда `ustor-cli get-image`.

Синтаксис команды:

```
ustor-cli ls-image <image_name> [параметры]
```

BASH | ↗

Обязательные аргументы:

- `<image_name>` - имя пула.

Основные параметры:

- `--json` - использовать формат JSON в выводе результата выполненной команды.

Пример вывода:

```
ustor-cli ls-image
```

BASH | ↗

NAME	POOL	SIZE	FLAGS	PARENT
Image_Test	Pool_Replica	20 G	-	

## 4.4. Удаление ресурса хранения

Для удаления ресурса хранения используется команда `ustor-cli delete-image`.

Синтаксис команды:

```
ustor-cli rm-image <image_name>
```

BASH | ↗

Обязательные аргументы:

- `<image_name>` - имя удаляемого ресурса.

## 5. Управление сопоставлением ресурсов с блочными устройствами

В MIND UStor реализована возможность подключения к ресурсам хранения СХД с помощью технологии NBD — «сетевого блочного устройства», исполняемого в пространстве пользователя. Эта технология даёт возможность монтировать блочные устройства через стандартное сетевое соединение.

Подключение и управление устройствами NBD осуществляется с помощью утилиты **ustor-nbd**.

## 5.1. Создание локального устройства NBD

Команда `ustor-nbd map` даёт возможность создать локальное устройства NBD из ресурса хранения.

Синтаксис команды:

```
ustor-nbd map --image <IMAGE_NAME> [параметры]
```

BASH | ↗

Обязательные аргументы:

- `--image <IMAGE_NAME>` - имя ресурса хранения, который будет использоваться для подключения в качестве устройства NBD.

Основные параметры:

- `--nbd_timeout <number>` - максимальное время выполнения любой операции чтения/записи в секундах, при превышении которого ядро остановит NBD-устройство.

Допустимо установить опцию в 0, чтобы выключить ограничение времени, но в этом случае станет невозможным выключение NBD-устройства при нештатном завершении процесса `ustor-nbd`.

- `--logfile <PATH>` - путь к файлу, в который будет осуществляться запись сообщений в процессе работы службы.
- `--dev_num <NUMBER>` - номер NBD-устройства после запуска службы `/dev/nbd<NUMBER>`.

По умолчанию используется автоматический подбор номера.

Примеры:

```
Создание NBD с параметрами по умолчанию из ресурса Image_Test
```

```
ustor-nbd map --image Image_Test
```

BASH | ↗

## 5.2. Отключение локального устройства NBD

Команда `ustor-nbd unmap` даёт возможность отключить (отмонтировать) ранее подключённое устройство NBD.

Синтаксис команды:

```
ustor-nbd unmap <DEVICE>
```

BASH | ↗

Обязательные аргументы:

- <DEVICE> - устройство NBD, которое необходимо отключить (отмонтировать). Необходимо указать полный путь до блочного устройства в формате `/dev/nbdX`.

Примеры:

Удаление устройства nbd0

```
ustor-nbd unmap /dev/nbd0
```

BASH | ↗

## 5.3. Просмотр списка подключенных устройств NBD

Команда `ustor-nbd ls` даёт возможность вывести информацию обо всех подключённых устройствах NBD и обслуживающих их процессах

Синтаксис команды:

```
ustor-nbd ls [параметры]
```

BASH | ↗

Основные параметры:

- `--json` - использовать формат JSON в выводе результата выполненной команды.

Пример вывода:

```
ustor-nbd ls
```

BASH | ↗

```
/dev/nbd0
image: Image_Test
interface: nbd
pid: 486921
```

# 6. Управление подключением устройств nbd к виртуальным машинам

## 6.1. Подключение nbd к виртуальной машине





В текущей реализации присутствуют следующие ограничения:

- NBD-устройства могут быть подключены только к работающим виртуальным машинам.
- NBD-устройства, подключенные к виртуальным машинам, не управляются и не отслеживаются Менеджером управления. Управление осуществляется исключительно средствами libvirt.
- Для виртуальных машин с подключенными NBD-устройствами недоступно резервное копирование в домен экспорта.

### Предварительные требования:

- На нужном хосте создано устройство NBD.
- Виртуальная машина работает на хосте с подключаемым устройством NBD.

### Порядок действий:

1. Подключитесь с учетной записью **root** по SSH к хосту, на котором создано нужное устройство NBD.
2. Убедитесь, что NBD устройство присутствует в системе.

```
ustor-nbd ls  
  
/dev/nbd0  
image: vm-disk1  
interface: nbd  
pid: 486921
```

BASH | □

3. Создайте xml-представление добавляемого диска:

```
tee -a /root/TestDisk1.xml <<EOF  
<disk type='network' device='disk'>  
    <source protocol='ustor' name='vm-disk1'/>  
    <target dev='sdb' bus='scsi' /> ① ②  
</disk>  
EOF
```

BASH | □

- ① В качестве `dev` укажите свободное имя блочного устройства на виртуальной машине.  
② В параметре `bus` также можно использовать значение `virtio`.

Для проверки свободных имен устройств виртуальной машины используйте следующую команду:

```
virsh -c qemu:///system?authfile=/etc/ovirt-hosted-engine/virsh_auth.conf  
domblklist <vm-name>
```

BASH | □

`<vm-name>` - имя виртуальной машины

Нужные значения будут отображаться в столбце "Назначение".

- Используйте **virsh** для подключения диска к нужной виртуальной машине.

```
BASH | 
virsh -c qemu:///system?authfile=/etc/ovirt-hosted-engine/virsh_auth.conf
attach-device --live --config testVm ~/TestDisk1.xml
```

Ожидаемый вывод выполнения команды:

```
BASH | 
Устройство успешно подключено
```

- В гостевой ОС убедитесь, что устройство подключено.

## 6.2. Отключение nbd от виртуальной машины

**Порядок действий:**

- Подключитесь с учетной записью **root** по SSH к хосту, на котором работает нужная виртуальная машина с подключенным NBD.
- Определите нужное имя целевого устройства виртуальной машины:

```
BASH | 
virsh -c qemu:///system?authfile=/etc/ovirt-hosted-engine/virsh_auth.conf
domblklist <vm-name> ①
```

① <vm-name> - имя виртуальной машины.

Пример вывода:

Назначение	Источник
sdb	/run/vdsm/payload/560047d1-20cd-43cd-994e-b04e54b8d765.img
sdc	-
vda	/rhev/data-center/mnt/blockSD/.../images/...
sde	vm-disk2
sdd	vm-disk1

Нужные значения отображаются в столбце "Назначение".

- Отключите диск от ВМ:

```
BASH | 
virsh -c qemu:///system?authfile=/etc/ovirt-hosted-engine/virsh_auth.conf
detach-disk <vm-name> <target-name> ① ②
```

① <vm-name> - имя виртуальной машины.

② <target-name> - имя целевого устройства, которое необходимо отключить.



# Системные требования для решения на базе ПСХД MIND uStor

## 1. Версии zVirt

Поддерживаемая гиперконвергентная инфраструктура должна быть реализована на базе zVirt Node версии 4.3 и выше.



Гиперконвергентное решение на базе ПСХД MIND uStor является экспериментальным (технический предрелиз). Настоятельно не рекомендуем использовать его в производственной среде. Кроме того, не будет поддерживаться обновление до следующей версии среды zVirt с развернутым решением ПСХД MIND uStor.

## 2. Узлы кластера ПСХД MIND uStor

Для развертывания полнофункционального решения SDS на базе ПСХД MIND uStor требуется не менее трех узлов. Также поддерживается масштабирование до 24 с шагом в 1 узел.

Каждый узел кластера должен соответствовать следующим требованиям:

- ЦПУ: не менее 12 ядер на сокет при наличии двух и более сокетов;
- ОЗУ: 32 ГБ и дополнительно 1,5 ГБ на каждый терабайт неразмеченного объема хранения;
- Хранилище:
  - Для zVirt Node и необходимого ПО: не менее 120 Гб.
  - Для организации хранения:
    - SSD, подключаемые по протоколу NVMe или SAS/SATA;
    - HDD, подключаемые по протоколу SAS/SATA.

В зависимости от целей и задач, которые будет решать создаваемый кластер СХД, тип, количество и ёмкость накопителей может варьироваться.



При подключении дисков через RAID-контроллер необходимо использовать режимы НВА или JBOD. При отсутствии данных режимов допускается создание RAID0 на каждом диске и отключение кэширования на чтение и запись.

- Не менее двух NIC (сетевых интерфейсных карт) на узел для разделения трафика данных и управления:
  - Для управления: Ethernet 2 x 1 Гбит/с и более;
    - рекомендуется агрегация сетевых интерфейсов (bonding или teaming).
  - Для организации хранения: 2 x Ethernet 10 Гбит/с+ и более;
    - требуется агрегация сетевых интерфейсов (bonding или teaming) для обеспечения отказоустойчивости подключения узла в кластере;
    - поддерживается RDMA с технологией On-Demand-Paging(ODP) и без;
    - В гиперконвергентных (HCI) инсталляциях трафик ПСХ MIND UStor может быть совмещен с трафиком виртуальных машин. В подобной конфигурации важно учитывать возможное пиковое потребление полосы пропускания разными типами трафика, чтобы исключить взаимно негативное влияние.

## 3. Менеджер управления

Минимальные требования к вычислительным ресурсам Менеджера управления, описанные здесь, основаны на типичной установке малого и среднего размера. Точные требования варьируются в зависимости от размера и нагрузки. Подробнее см в разделе [Рекомендации по развертыванию Менеджера управления](#) лучших практик.

Таблица 1. Требования к аппаратному обеспечению менеджера управления zVirt

Ресурс	Минимальные требования	Рекомендуемые требования
Процессор	4 ядра/vCPU	8 ядер/vCPU
Память	4 ГБ	32 ГБ
Жёсткий диск	<ul style="list-style-type: none"><li>• При развертывании в архитектуре HostedEngine: 51 ГБ</li><li>• При развертывании в архитектуре Standalone: 120 ГБ</li></ul>	120 ГБ
Сетевой интерфейс	1 сетевой интерфейс (NIC) 1 Гбит/с	

## 4. Сети

Для всех узлов кластера в гиперконвергентной среде и для Менеджера управления необходимо использовать полностью определенные доменные имена (FQDN), которые могут быть разрешены DNS как в прямом, так и в обратном направлении.

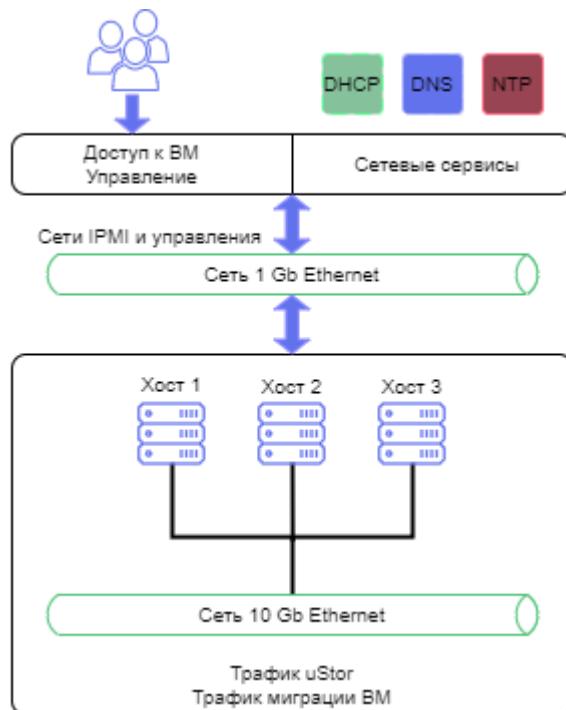
Если внешний DNS недоступен, например, в изолированной среде, убедитесь, что файл **/etc/hosts** на каждом узле кластера содержит сопоставления имен и адресов всех узлов и Менеджера управления.

! Рекомендуется использовать DNS вместо **/etc/hosts**.

IPv6 поддерживается только в средах с адресами IPv6 (включая адреса DNS и шлюзов). Dual-stack (IPv4 и IPv6) среды не поддерживаются.

Рекомендуется использовать выделенные сети:

- Внешняя сеть для управления и трафика виртуальных машин.
- Внутренняя сеть для трафика Хранилища и миграции виртуальных машин.



Рекомендуется, чтобы каждый узел кластера имел два порта Ethernet, по одному для каждой сети. Это обеспечивает оптимальную производительность.

Для обеспечения высокой доступности рекомендуется разместить каждую сеть на отдельном сетевом коммутаторе. Для повышения отказоустойчивости обеспечьте отдельный источник питания для каждого коммутатора.

### Внешняя сеть

- Используется виртуализацией zVirt для трафика управления и трафика виртуальных машин.
- Требуется не менее одного подключения Ethernet со скоростью 1 Гбит/с.
- IP-адреса, используемые в этой сети, должны использовать подсети, отличные от внутренней сети.

## **Внутренняя сеть**

- Используется для межузлового взаимодействия кластера СХД.
- Требуется не менее одного подключения Ethernet 10 Гбит/с+ или RDMA (RoCEv2) 25 Гбит/с+ с поддержкой технологии ODP.

Устройства сетевого ограждения, использующие IPMI, должны находиться в выделенной сети.

Перед началом процесса развертывания определите следующие детали:

- IP-адрес шлюза для узлов кластера. Этот адрес должен отвечать на запросы ping.



Шлюз необходимо назначать только на том интерфейсе, который расположен во внешней сети.

- IP-адрес сети управления (ovirtmgmt).
- Полное доменное имя (FQDN) для Менеджера управления.

## **5. Виртуальные машины**

Количество виртуальных машин, которые можно запустить в гиперконвергентном решении, сильно зависит от функций этих виртуальных машин и уровня их загрузки.

Ограничение максимального количества виртуальных машин и виртуальных процессоров см. в статье [Максимальные показатели \(Ограничения\) zVirt 4.X](#).

## **6. Масштабирование**

Количество узлов в начальном развертывании кластера СХД должно быть не менее 3.

После начального развертывания кластер можно масштабировать до 24 узлов с шагом 1.

Также масштабирование развертывания возможно путем добавления дисков и расширения пулов и ресурсов хранения ПСХД MIND uStor.

# Утилита AAA JDBC

## 1. Управление пользователями

### 1.1. Создание пользователя

```
ovirt-aaa-jdbc-tool user add jdoe \
--attribute=firstName=John \
--attribute=lastName=Doe \
--attribute=email=jdoe@unknown.com
```

Только имя пользователя (`username`) является обязательным, остальные атрибуты необязательны. Для пользователя могут быть указаны следующие атрибуты:

- `department`
- `description`
- `displayName`
- `email`
- `firstName`
- `lastName`
- `title`



Вновь созданные пользователи не могут войти в систему, пока не будет выполнена команда сброса пароля (`password-reset`).

### 1.2. Отображение подробной информации о существующем пользователе

```
ovirt-aaa-jdbc-tool user show jdoe
```

### 1.3. Обновление существующего пользователя

```
ovirt-aaa-jdbc-tool user edit jdoe \
--attribute="displayName=John Doe"
```

## 1.4. Удаление существующего пользователя

```
ovirt-aaa-jdbc-tool user delete jdoe
```

## 1.5. Включение/отключение пользователя

Пользователь может быть отключен для предотвращения входа в систему:

```
ovirt-aaa-jdbc-tool user edit jdoe --flag=+disabled
```

И следующая команда снова включает пользователя:

```
ovirt-aaa-jdbc-tool user edit jdoe --flag=-disabled
```

## 1.6. Разблокировка заблокированного пользователя

Пользователь может быть заблокирован, например, при большом количестве неудачных входов в систему. Следующая команда разблокирует пользователя:

```
ovirt-aaa-jdbc-tool user unlock jdoe
```

## 1.7. Изменение срока действия учетной записи

```
ovirt-aaa-jdbc-tool user edit jdoe --account-valid-to="2025-08-15 10:30:00Z"
```

## 1.8. Управление паролями

Следующая команда устанавливает новый пароль для пользователя **jdoe** в интерактивном режиме:

```
ovirt-aaa-jdbc-tool user password-reset jdoe
```

При использовании опции **password** поддерживаются следующие типы ввода пароля:

- **interactive** - инструмент запрашивает ввод пароля
- **pass:STRING** - пароль задается в командной строке
- **env:KEY** - пароль задается в переменной окружения ENV
- **pass:[\*\*\*\*]** - пароль считывается из указанного файла

- none: - задает пустой пароль



По умолчанию срок действия пароля истекает в тот же момент, когда он задан. Следующая команда устанавливает новый пароль и дату истечения пароля на **15 августа 2025 10:30:00 UTC**:

```
ovirt-aaa-jdbc-tool user password-reset jdoe --password-valid-to="2025-08-15  
10:30:00Z"
```

## 2. Управление группами

### 2.1. Создание группы

```
ovirt-aaa-jdbc-tool group add group1 \  
--attribute="description=First group"
```

Обязательным является только имя группы, остальные атрибуты необязательны. Для группы могут быть указаны следующие атрибуты:

- description
- displayName

### 2.2. Отображение подробной информации о существующей группе

```
ovirt-aaa-jdbc-tool group show group1
```

### 2.3. Обновление существующей группы

Следующая команда обновляет отображаемое имя:

```
ovirt-aaa-jdbc-tool group edit group1 \  
--attribute="displayName=Group 1"
```

### 2.4. Удаление существующей группы

```
ovirt-aaa-jdbc-tool group delete group1
```

## 3. Управление членством в группах

---

### 3.1. Добавление пользователя в группу

Следующая команда добавляет пользователя **jdoe** в группу **group1**:

```
ovirt-aaa-jdbc-tool group-manage useradd group1 --user=jdoe
```

### 3.2. Показывать членов группы

```
ovirt-aaa-jdbc-tool group-manage show group1
```

### 3.3. Удаление пользователя из группы

Следующая команда удаляет пользователя **jdoe** из группы **group1**:

```
ovirt-aaa-jdbc-tool group-manage userdel group1 --user=jdoe
```

### 3.4. Добавление группы в группу

Следующая команда добавляет группу **group2** в группу **group1**:

```
ovirt-aaa-jdbc-tool group-manage groupadd group1 --group=group2
```

### 3.5. Удаление группы из группы

Следующая команда удаляет группу **group2** из группы **group1**:

```
ovirt-aaa-jdbc-tool group-manage groupdel group1 --group=group2
```

## 4. Поиск пользователей/групп

---

### 4.1. Поиск пользователей

Следующая команда отображает всех существующих пользователей:

```
ovirt-aaa-jdbc-tool query --what=user
```



Для сужения результатов можно использовать следующие атрибуты пользователя:

- department
- description
- displayName
- email
- firstName
- id
- lastName
- name
- title

Например, следующая команда ищет пользователей, имя пользователя которых начинается с **j**:

```
ovirt-aaa-jdbc-tool query --what=user --pattern="name=j*"
```



## 4.2. Поиск групп

Следующая команда отображает все существующие группы:

```
ovirt-aaa-jdbc-tool query --what=group
```



Для сужения результатов можно использовать следующие групповые атрибуты:

- description
- displayName
- name

Например, следующая команда ищет группы, название которых начинается с **gr**:

```
ovirt-aaa-jdbc-tool query --what=group --pattern="name=gr*"
```



## 5. Настройки

## 5.1. Расширения AAA-JDBC

Настройки расширения **AAA-JDBC** можно отобразить с помощью:

```
ovirt-aaa-jdbc-tool settings show
```

## 5.2. Обновление параметров

Следующая команда обновляет параметр **PASSWORD\_EXPIRATION\_DAYS**:

```
ovirt-aaa-jdbc-tool settings set --name=PASSWORD_EXPIRATION_DAYS --value=365
```

## 5.3. Конфигурация дополнительных доменов



Наиболее актуальную документацию можно найти в **README.admin** внутри пакета **ovirt-engine-extension-aaa-jdbc**



После всех нижеперечисленных настроек, управление ролями пользователей нового домена неполноценны.

Чтобы настроить новый профиль, использующий расширение **aaa-jdbc**, выполните следующие шаги:

1. Настройте базу данных для расширения **aaa-jdbc**

Замените **DB\_NAME**, **DB\_USER** и **DB\_PASSWORD** на реальные значения:

```
su - postgres -c "psql -d template1 << __EOF__  
create user DB_USER password 'DB_PASSWORD';  
create database DB_NAME owner DB_USER template template0  
encoding 'UTF8' lc_collate 'en_US.UTF-8' lc_ctype 'en_US.UTF-8';  
__EOF__  
"
```

2. Настройте PostgreSQL

Добавьте в файл **/var/lib/pgsql/data/pg\_hba.conf** следующие строки (замените **DB\_NAME** и **DB\_USER** реальными значениями):

```
host      DB_NAME      DB_USER      0.0.0.0/0          md5  
host      DB_NAME      DB_USER      ::0/0          md5
```

Эти строки должны располагаться перед следующими строками:

```
host    all      all      127.0.0.1/32      ident
host    all      all      ::1/128        ident
```

### 3. Заполните базу данных для расширения **aaa-jdbc**

Замените **DB\_HOST**, **DB\_NAME**, **DB\_USER** и **DB\_PASSWORD** реальными значениями:

```
PGPASSWORD="DB_PASSWORD" \
/usr/share/ovirt-engine-extension-aaa-jdbc/dbscripts/schema.sh \
-s DB_HOST \
-p DB_PORT \
-d DB_NAME \
-u DB_USER \
-c apply
```

### 4. Настроить профиль **AAA**

Выберите имя своего профиля (оно будет видно пользователям при входе в систему) и скопируйте примеры файлов конфигурации, переименуйте в соответствии с вашим профилем (измените **PROFILE** выбранным значением):

```
cp /usr/share/ovirt-engine-extension-aaa-
jdbc/examples/extension.d/authn.properties \
/etc/ovirt-engine/extensions.d/PROFILE-authn.properties
cp /usr/share/ovirt-engine-extension-aaa-
jdbc/examples/extension.d/authz.properties \
/etc/ovirt-engine/extensions.d/PROFILE-authz.properties
cp /usr/share/ovirt-engine-extension-aaa-
jdbc/examples/aaa/profile.properties \
/etc/ovirt-engine/aaa/PROFILE.properties
```

Отредактируйте созданные файлы **PROFILE.properties\*** и замените переменные, окруженные @, реальными значениями. Измените права доступа, выполнив следующие команды:

```
cd /etc/ovirt-engine/extensions.d/
chown ovirt:ovirt PROFILE-authn.properties PROFILE-authz.properties
chmod 600 PROFILE-authn.properties PROFILE-authz.properties
chown ovirt:ovirt ../aaa/PROFILE.properties
chmod 600 ../aaa/PROFILE.properties
```

### 5. Перезапустите службу **ovirt-engine**:

```
systemctl restart ovirt-engine
```

Проверьте **/var/log/ovirt-engine/engine.log**, чтобы убедиться, что расширение успешно инициализировано в соответствии с **PROFILE**.

## 6. Настройка пользователей и групп

Настройте своих пользователей и группы с помощью утилиты **ovirt-aaa-jdbc-tool** и укажите конфигурацию базы данных с помощью параметра командной строки **--db-config**:

```
ovirt-aaa-jdbc-tool \
--db-config=/etc/ovirt-engine/aaa/PROFILE.properties \
user add test
ovirt-aaa-jdbc-tool \
--db-config=/etc/ovirt-engine/aaa/PROFILE.properties \
user password-reset test --password-valid-to="2025-08-15 10:30:00Z"
```

7. Войдите в веб-интерфейс, используя существующую учетную запись администратора, и назначьте нужные разрешения пользователям/группам, указанным в вашем **PROFILE**. Для назначения прав перейдите в раздел **Управление > Пользователи > test (Имя пользователя) > Разрешения > Добавить системные разрешения**.

# Утилиты БД

Каталог `/usr/share/ovirt-engine/setup/dbutils` содержит несколько скриптов для работы с базой данных.

## 1. Taskcleaner - Утилита для очищения зависших процессов

Использование:

```
taskcleaner.sh [-h] [-s server] [-p PORT] [-d DATABASE] [-u USERNAME] [-l  
LOGFILE] [-t taskId] [-c commandId]  
[-z] [-R] [-C] [-J] [-q] [-v]
```

- **-s SERVERNAME** - Имя сервера базы данных (по умолчанию localhost).
- **-p PORT** - Порт для подключения к базе данных (по умолчанию 5432).
- **-d DATABASE** - Имя базы данных (по умолчанию engine).
- **-u USERNAME** - Имя пользователя для подключения к базе данных.
- **-l LOGFILE** - Файл журнала для логирования (по умолчанию taskcleaner.sh.log).
- **-t TASK\_ID** - Удалить задачу по ее идентификатору.
- **-c COMMAND\_ID** - Удалить все задачи, связанные с заданным идентификатором команды.
- **-z** - Удаляет/отображает задачу с типом "Зомби".
- **-R** - Удаляет все задания "Зомби".
- **-C** - Удаляет связанные данные.
- **-J** - Удаляет связанные шаги задачи.
- **-q** - Тихий режим, не запрашивать подтверждение.
- **-v** - Режим отладки (ВНИМАНИЕ: много вывода).
- **-h** - Справка

## 2. UnlockEntity - Утилита для разблокировки ВМ, Шаблона и связанных дисков

Использование:

```
./unlock_entity.sh [параметры] [ENTITIES]
```



- **-h** - Справка
- **-v** - Включить режим отладки (**ВНИМАНИЕ**: много выходных данных).
- **-l LOGFILE** - Расположение файла отладки.
- **-s HOST** - Имя сервера базы данных (по умолчанию localhost).
- **-p PORT** - Порт для подключения к базе данных (по умолчанию 5432).
- **-u USER** - Имя пользователя для подключения к базе данных (по умолчанию engine).
- **-d DATABASE** - Имя базы данных (по умолчанию engine).
- **-t TYPE** - Тип объекта {vm | template | disk | snapshot}.
- **-r** - Рекурсивный, разблокирует все диски под выбранным vm/template.
- **-q** - Запрос к БД для вывода списка заблокированных объектов.
- **ENTITIES** - Список имен объектов в случае vm/template, UUID в случае диска.

**Например:**

```
/usr/share/ovirt-engine/setup/dbutils/unlock_entity.sh -t all
```



разблокирует все сущности.

### 3. **Fkvalidator** - Утилита для проверки БД на целостность и исправления ошибок

**Использование:**

```
fkvalidator.sh [-h] [-s SERVERNAME [-p PORT]] [-d DATABASE] [-u USERNAME] [-l LOGFILE] [-f] [-v]
```



- **-s SERVERNAME** - Имя сервера базы данных (по умолчанию localhost).
- **-p PORT** - Порт для подключения к базе данных (по умолчанию 5432).
- **-d DATABASE** - Имя базы данных (по умолчанию engine).
- **-u USERNAME** - Имя пользователя для подключения к базе данных.
- **-l LOGFILE** - Файл журнала для логирования (по умолчанию fkvalidator.sh.log).
- **-f** - Удалить поврежденные данные.
- **-v** - Включить режим отладки (**ВНИМАНИЕ**: много выходных данных).

- **-h** - Справка.
- 

2025 orionsoft. Все права защищены.