

Virtuozzo

Virtuozzo Hybrid Infrastructure 4.5

SELF-SERVICE GUIDE

Table of contents

1 About this guide	4
2 Logging in to the self-service panel	5
3 Managing users and projects	6
3.1 Creating users	6
3.2 Assigning users to projects	7
3.3 Viewing project quotas	9
4 Managing compute resources	11
4.1 Managing virtual machines	11
4.1.1 Supported guest operating systems	11
4.1.2 Creating virtual machines	12
4.1.3 Connecting to virtual machines	17
4.1.4 Managing virtual machine power state	18
4.1.5 Reconfiguring and monitoring virtual machines	18
4.1.6 Shelving virtual machines	19
4.1.7 Rescuing virtual machines	19
4.1.8 Managing guest tools	20
4.1.9 Troubleshooting virtual machines	22
4.1.10 Deleting virtual machines	23
4.2 Managing security groups	24
4.2.1 Creating and deleting security groups	24
4.2.2 Managing security group rules	25
4.2.3 Changing security group assignment	25
4.3 Managing Kubernetes clusters	26
4.3.1 Creating and deleting Kubernetes clusters	26
4.3.2 Managing Kubernetes worker groups	28
4.3.3 Using persistent volumes for Kubernetes pods	30
4.3.4 Creating external load balancers in Kubernetes	36
4.3.5 Assigning Kubernetes pods to specific nodes	38
4.4 Managing images	38
4.4.1 Uploading and removing images	39
4.4.2 Creating volumes from images	39
4.4.3 Mounting ISO images to virtual machines	40
4.4.4 Preparing templates	40
4.5 Managing volumes	45
4.5.1 Creating, editing, and removing volumes	45

4.5.2 Cloning volumes	46
4.5.3 Attaching and detaching volumes	47
4.5.4 Creating images from volumes	48
4.5.5 Managing volume snapshots	49
4.6 Managing virtual networks	51
4.7 Managing virtual routers	53
4.7.1 Managing router interfaces	54
4.7.2 Managing static routes	56
4.8 Managing floating IP addresses	58
4.9 Managing load balancers	59
4.9.1 Managing balancing pools	63
4.10 Managing SSH keys	64

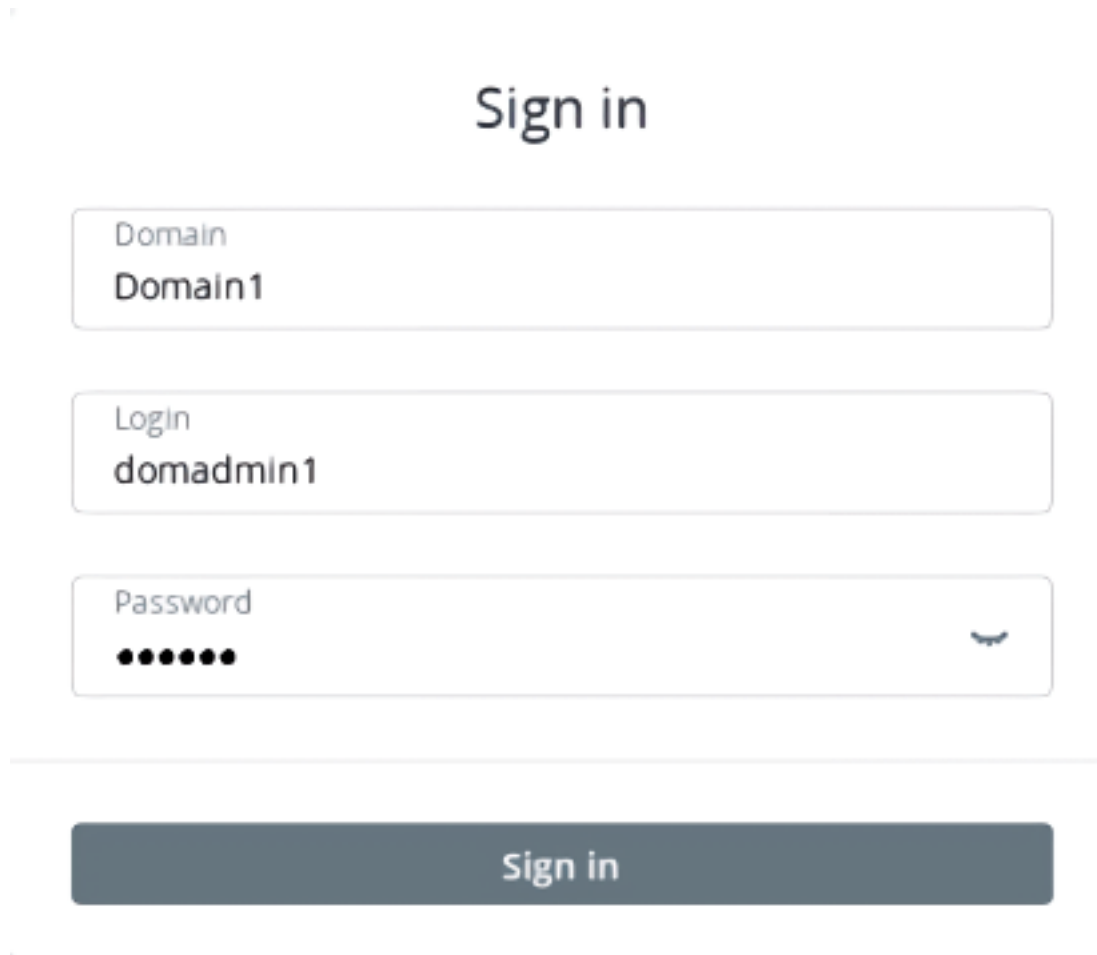
1 About this guide

This guide is intended for domain administrators and project members and explains how to manage project users and compute resources using the self-service panel.

2 Logging in to the self-service panel

To log in to the self-service panel, do the following:

1. Visit the panel's IP address on port 8800.
2. Enter your domain name (case sensitive) as well as user name and password. Alternatively, if you are given the link to the self-service panel for a specific domain, you will only need to provide the user name and password.



The screenshot shows a login interface with the following elements:

- Sign in** title centered at the top.
- Domain** input field containing the text **Domain1**.
- Login** input field containing the text **domadmin1**.
- Password** input field containing seven black dots, with a small eye icon on the right side to toggle visibility.
- A horizontal line separating the input fields from the button.
- A dark grey **Sign in** button at the bottom.

3 Managing users and projects

Important

Only domain administrators can manage users and projects.

In the self-service panel, you can create users and assign them to projects within a domain. When you create a user, you select its role. A user can be assigned one of the following roles:

- A domain administrator can manage virtual objects in all projects within the assigned domain as well as project and user assignment in the self-service panel.
- A project member acts as a project administrator in a specific domain in the self-service panel. A project member can be assigned to different projects and can manage virtual objects in them.

With users, you can also do the following:

- Edit the user credentials or permissions
- Allow or prohibit user login by enabling and disabling a user account
- Delete a user

With projects, you can only view their quotas and assign members to them.

3.1 Creating users

Domain administrators can create other domain administrators and project members.

To create a user, do as follows:

1. Select the domain in the drop-down list in the top right corner.
2. Open the **Users** screen and click **Create user**.
3. In the **Create user** window, specify the user name, password, and, if required, a user email address and description. The user name must be unique within a domain.
4. Select the desired role from the **Role** drop-down menu.
5. Click **Create**.

Create user ✕

user1

user1@example.com

●●●●●●●●

Domain administrator

Can create and manage projects and services in the assigned domain.

3.2 Assigning users to projects

Domain administrators can manage project members' assignment on the **Projects** and **Users** screens.

To assign a user to a project, do one of the following:

- On the **Projects** screen:
 1. Click the project to which you want to assign users (not the project name).
 2. On the project panel, click **Assign members**.
 3. In the **Assign members** window, choose one or multiple users to assign to the project. Only user accounts with the **Project member** role are displayed. Optionally, click **Create project member** to create a new project member in a new window.
 4. Click **Assign**.

Assign members ✕

Select users to assign as members to the project "dom1project1".

+ Create project member

<input checked="" type="checkbox"/>	Login ↑	Email
<input checked="" type="checkbox"/>	projectmember1	—

Cancel
Assign

- On the **Users** screen:
 1. Click the user account with the **Project member** role whom you want to assign to the project.
 2. On the user panel, click **Assign to project**.
 3. On the **Assign user to projects** window, select one or multiple projects, and then click **Assign**.

Assign user to projects ✕

Select projects to assign to the user "user1".






<input checked="" type="checkbox"/>	Name ↑	Description
<input checked="" type="checkbox"/>	project1	A custom project

Cancel
Assign


To unassign a user from a project, do one of the following:



- On the **All projects** screen:
 1. Click the project to unassign users from.
 2. On the project panel, open the **Members** tab.
 3. Click the cross icon next to a user you want to unassign.

project1 ✕

 Edit  + Assign members  Edit quotas  Disable  Delete





Properties **Members (1)** Quotas

Search 


Login ↑	Email	
 user1	user1@example.com	



- On the **All users** tab:
 1. Click the user to unassign from the project.
 2. On the user panel, open the **Projects** tab.
 3. Click the cross icon next to the project from which you want to unassign the user.

user1 ✕

 Edit  + Assign to project  Disable  Delete

Properties **Projects (1)**















Search 

Name ↑	Description	
 project1	A custom project	

3.3 Viewing project quotas

Each project is allocated a certain amount of compute resources by means of quotas. Domain administrators can view project quotas on the project details screen.

To view quotas of a project, open **Projects**, click the desired project in the list, and then switch to the **Quotas** tab.

Compute		
 vCPUs		1 / 24 cores
 RAM		512 MIB / 48 GIB
 Storage policy		
default		1 GIB / 2 TIB
 Floating IPs		1 / 20
 Load balancers		0 / 10
 Kubernetes clusters		0 / 10
 Placements		
placement1		1 / 20

4 Managing compute resources

4.1 Managing virtual machines

Each virtual machine (VM) is an independent system with an independent set of virtual hardware. Its main features are the following:

- A virtual machine resembles and works like a regular computer. It has its own virtual hardware. Software applications can run in virtual machines without any modifications or adjustment.
- Virtual machine configuration can be changed easily, for example, by adding new virtual disks or memory.
- Although virtual machines share physical hardware resources, they are fully isolated from each other (file system, processes, sysctl variables) and the compute node.
- A virtual machine can run any supported guest operating system.

The following table lists the current virtual machine configuration limits:

Resource	Limit
RAM	1 TiB
CPU	48 virtual CPUs
Storage	15 volumes, 512 TiB each
Network	15 NICs

4.1.1 Supported guest operating systems

The guest operating systems listed below have been tested and are supported in virtual machines.

Note

Only the x64 architecture is supported.

Windows versions

- Windows Server 2019 - Essentials, Standard, Datacenter editions
- Windows Server 2016 - Essentials, Standard, Datacenter editions
- Windows Server 2012 R2 - Essentials, Standard, Datacenter editions
- Windows Server 2012 - Standard, Datacenter editions
- Windows Server 2008/2008 R2 - Standard, Datacenter editions
- Windows 10 - Home, Professional, Enterprise, Enterprise 2016 LTSB editions
- Windows 8.1 - Home, Professional, Enterprise editions
- Windows 7 - Home, Professional, Enterprise editions

Linux distributions and versions

- CentOS 8.x, 7.x, 6.x
- Red Hat Enterprise Linux 8.x, 7.x
- Debian 10.x, 9.x
- Ubuntu 20.04.x, 18.04.x, 16.04.x

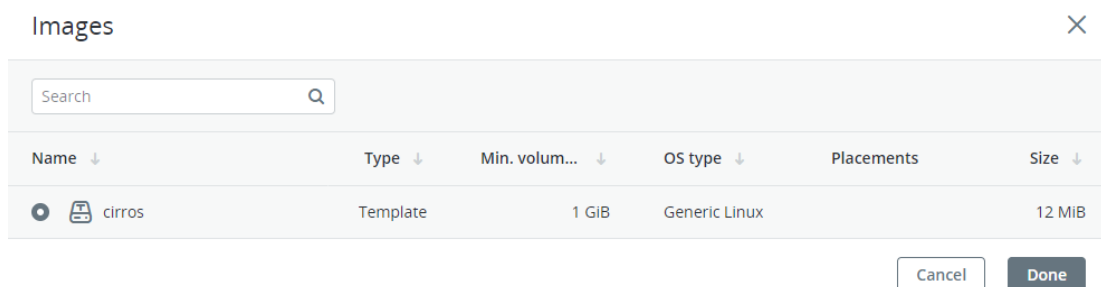
4.1.2 Creating virtual machines

Before you proceed to creating VMs, check that you have these:

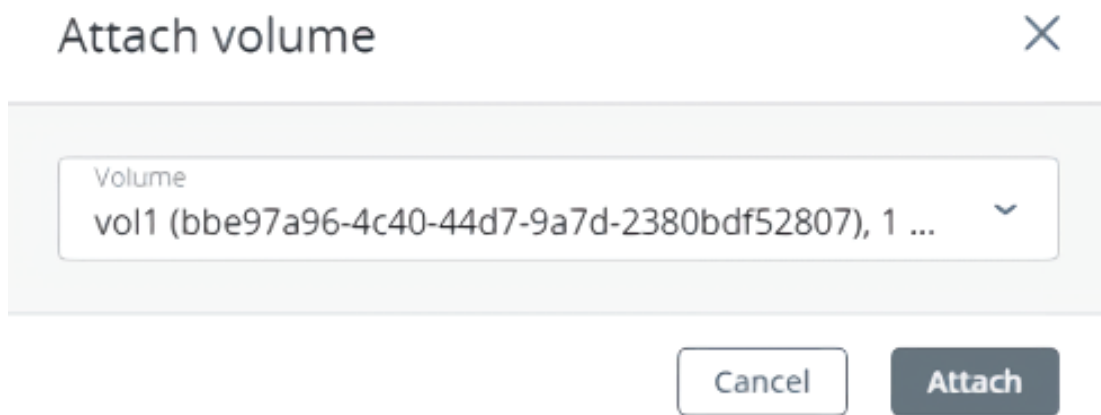
- Prepare a guest OS source (refer to "Managing images" (p. 38)):
 - A distribution ISO image of a guest OS to install in the VM
 - A template that is a boot volume in the QCOW2 format
 - A boot volume. You can obtain a boot volume after deleting a VM with the installed operating system. The boot volume will not be removed along with the VM only if during the creation it had the parameter **Delete on termination** set to **No**.
- Create one or more compute networks (refer to "Managing virtual networks" (p. 51)).
- Optionally, configure custom security groups (refer to "Managing security groups" (p. 24)).
- Add an SSH key (refer to "Managing SSH keys" (p. 64)). You can specify an SSH key only when creating VMs from a template or boot volume.

To create a virtual machine, do the following:

1. On the **Virtual machines** screen, click **Create virtual machine**. A window will open where you will need to specify the VM parameters.
2. Specify a name for the new VM.
3. Select the VM boot media:
 - If you have an ISO image or a template
 - a. Select **Image** in the **Deploy from** section, and then click **Specify** in the **Image** section.
 - b. In the **Images** window, select the ISO image or template, and then click **Done**.



- If you have a compute boot volume
 - a. Select **Volume** in the **Deploy from** section, and then click **Specify** in the **Volumes** section.
 - b. In the **Volumes** window, click **Attach**.
 - c. In the **Attach volume** window, find and select the volume, and then click **Attach**.



If you attach more than one volume, the first attached volume becomes the boot volume, by default. To select another volume as bootable, place it first in the list by clicking the up arrow button next to it.






After selecting the boot media, volumes required for this media to boot will be automatically added to the **Volumes** section.

4. Configure the VM disks:
 - a. In the **Volumes** window, make sure the default boot volume is large enough to accommodate the guest OS. Otherwise, click the ellipsis icon next to it, and then **Edit**. Change the volume size and click **Save**.
 - b. [Optional] Add more disks to the VM by creating or attaching volumes. To do this, click the pencil icon in the **Volumes** section, and then **Add** or **Attach** in the **Volumes** window.
 - c. Select volumes that will be removed during the VM deletion. To do this, click the pencil icon in the **Volumes** section, click the ellipsis icon next to the needed volume, and then **Edit**. Enable **Delete on termination** and click **Save**.
 - d. When you finish configuring the VM disks, click **Done**.
5. [Optional] If you have any placements, the **Placement** drop-down list is displayed. Placements are created by the administrator to group nodes or VMs sharing a distinctive feature, like a special license. Select the placement corresponding to the VM characteristics.
6. Choose the amount of RAM and CPU resources that will be allocated to the VM in the **Flavor** section. In the **Flavor** window, select a flavor, and then click **Done**.

Important

When choosing a flavor for a VM, ensure it satisfies the hardware requirements of the guest OS.

Flavor ✕

	Name ↓	vCPU ↓	Memory
<input checked="" type="radio"/>	 tiny	1	512 MiB
<input type="radio"/>	 small	1	2 GiB
<input type="radio"/>	 medium	2	4 GiB
<input type="radio"/>	 large	4	8 GiB
<input type="radio"/>	 xlarge	8	16 GiB


7. Add network interfaces to the VM in the **Networks** section:

- a. In the **Networks interfaces** window, click **Add**, select a compute network, and then click **Next**.

Add network interface ✕

Select a network

🔍

-  public: 10.136.16.0/22
-  private: 192.168.128.0/24

b. Select a network security group to assign the VM to:

- For a virtual network with enabled IP address management, spoofing protection is enabled and the **default** security group is selected by default. This security group allows all incoming and outgoing traffic on all the VM ports. You can select another security group or multiple security groups, and then click **Done**.
To disable spoofing protection, clear all of the check boxes and turn off the toggle switch. Security groups cannot be configured with disabled spoofing protection.
- For a virtual network with disabled IP address management, spoofing protection is disabled by default and cannot be enabled. Security groups cannot be configured for such a network.
- For a shared physical network, spoofing protection cannot be configured by a self-service user. If you want to enable or disable spoofing protection, contact your system administrator.

Then, click **Done**.

Add network interface ✕

Select a security group

Spoofing protection

Search

	Name ↓	Project	Rules	Description
<input checked="" type="checkbox"/>	default	admin	5	Default security group

A network interface connected to the selected network will appear in the **Network interfaces** list.

- c. [Optional] Edit additional parameters of newly added network interfaces, such as IP and MAC addresses and security groups. To do this, click the ellipsis icon, click **Edit**, and then set the parameters. You will not be able to edit the IP address later. To change the IP address assignment, you will be able to delete the old network interface and replace it with a new one.
 - d. When you finish configuring the VM network interfaces, click **Done**.
8. [Optional] If you have chosen to boot from a template or volume, which has cloud-init and OpenSSH installed:

Important

As cloud images have no default password, you can access VMs deployed from them only by using the key authentication method with SSH.

- Add an SSH key to the VM, to be able to access it via SSH without a password. In the **Select an SSH key** window, select an SSH key and then click **Done**.

Select an SSH key ✕

+ Add

	Name ↑	Description ↑	Created on	
<input checked="" type="radio"/>	root_node001vstoragedom	My public key	June 11, 2019 11:34 AM	⋮

ⓘ To be able to manage SSH keys, make sure the VM template has cloud-init installed.

Cancel Done

- Add user data to customize the VM after launch, for example, change a user password. Write a cloud-config or shell script in the **Customization script** field or browse a file on your local server to load the script from.

Provide a customization script



Provide user data to customize the VM after launch. User data can be in one of two formats: cloud-config or shell script. For the guest OS to be customizable, the template must have cloud-init installed.

```
Customization script
#cloud-config
user: myuser
password: password
chpasswd: {expire: False}
ssh_pwauth: True
```

Load from file

Cancel

Save

To inject a script in a Windows VM, refer to the [Cloudbase-Init documentation](#). For example, you can set a new password for the account using the following script:

```
#ps1
net user <username> <new_password>
```

9. After configuring all of the VM parameters, click **Deploy** to create and boot the VM.

If you are deploying the VM from an ISO image, you need to install the guest OS inside the VM by using the built-in VNC console. Virtual machines created from a template or a boot volume already have a preinstalled guest OS.

4.1.3 Connecting to virtual machines

To connect to a VM via the built-in VNC console, click **Console** on the VM right pane. The console will open in a separate browser window. In the console, you can send a key combination to a VM, take a screenshot of the console window, and download the console log (refer to "Troubleshooting virtual machines" (p. 22)).

To be able to connect to a VM via SSH, make sure it has cloud-init and OpenSSH installed (refer to "Preparing templates" (p. 40)). To connect to a VM via SSH, specify the username and VM IP address in the SSH terminal:

```
# ssh <username>@<VM_IP_address>
```

Linux cloud images have the default login, depending on the operating system, for example, `centos` or `ubuntu`. To connect to a Windows VM, enter the username that you specified during Cloudbase-Init installation.

If you have deployed a VM without specifying an SSH key, you also need to enter a password to log in to the VM.

4.1.4 Managing virtual machine power state

After you create a virtual machine, you can manage its power state by using available actions.

To see the full list of actions available for the VM current state, click the VM or the ellipsis button next to a it. You can do the following:

- To power up a VM, click **Run**.
- To gracefully shut down a running VM, click **Shut down**. The default shutdown timeout, after which a virtual machine will be powered off, is 10 minutes.
- To forcibly cut off power from a VM, click **Power off**.
- To softly reboot a running VM, click **Reboot**.
- To reboot a VM without the guest OS graceful shutdown, click **Hard reboot**.
- To save the current VM state to a file, click **Suspend**. This may prove useful, for example, if you need to restart the host but do not want to quit the applications currently running in the VM or restart its guest OS.
- To restore a VM from the suspended state, click **Resume**.

4.1.5 Reconfiguring and monitoring virtual machines

To monitor virtual machine's CPU, storage, and network usage, select the VM and open the **Monitoring** tab.

The default time interval for the charts is twelve hours. To zoom into a particular time interval, select the internal with the mouse; to reset zoom, double-click any chart.

The following performance charts are available:

CPU / RAM

CPU and RAM usage by the VM.

Network

Incoming and outgoing network traffic.

Storage read/write

Amount of data read and written by the VM.

Read/write latency

Read and write latency. Hovering the mouse cursor over a point on the chart, you can also see the average and maximum latency for that moment, as well as the 95 and 99 percentiles.

Note

Averaged values are calculated every five minutes.

To reconfigure a VM, select it and, on the **Overview** tab, click the pencil icon next to a parameter you need to change. You cannot do the following:

- Change, detach, or delete the boot volume
- Manage non-boot volumes, except attaching and detaching
- Modify previously added network interfaces
- Attach and detach network interfaces to and from shelved VMs
- Change the flavor for running and shelved VMs

4.1.6 Shelving virtual machines

You can unbind a stopped VM from the node it is hosted on and release its reserved resources such as CPU and RAM. A shelved VM remains bootable and retains its configuration, including the IP addresses.

To shelve a stopped VM, click **Shelve** on the VM right pane. Virtual machines in other states can be shelved by clicking **Shut down** or **Power off**, and then selecting the check box **Shelve virtual machine** in the confirmation window.

To spawn a shelved VM on a node with enough resources to host it, click **Unshelve** on the VM right pane.

4.1.7 Rescuing virtual machines

Note

This feature is available via GUI only for Linux virtual machines. It is also available via the command line for Windows VMs.

If a VM experiences boot problems, you can send it to the rescue mode to access its boot volume. When a VM in the "Active" state is sent to the rescue mode, it is shut down softly first. Once the VM is in the rescue mode, you can connect to it via SSH or via the console. Its previous boot disk is now attached as a secondary one. You can mount the disk and repair it.

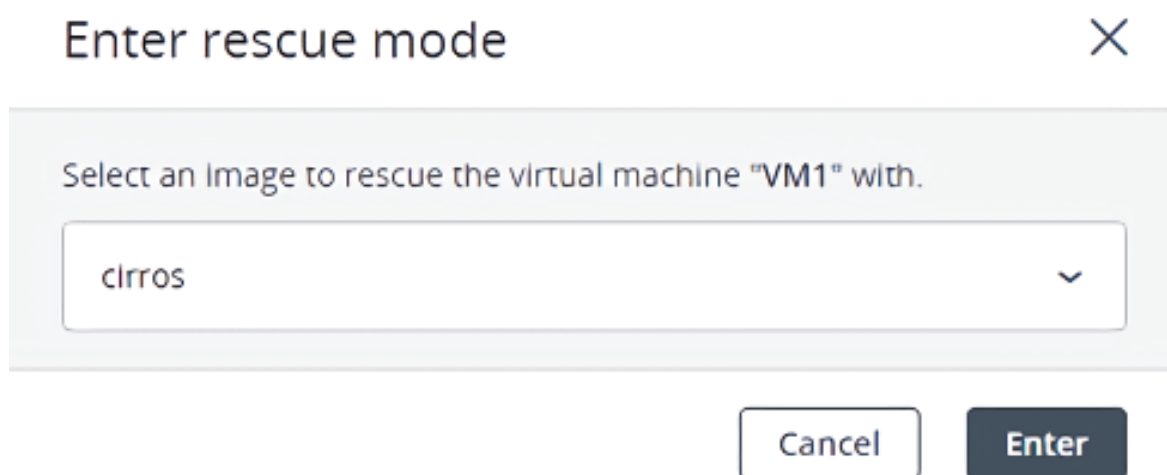
Before sending a VM to the rescue mode, consider the following:

- The rescue mode can only use qcow2 images (templates) for booting a VM. For instructions on making templates, refer to "Preparing templates" (p. 40).
- You can send a VM to the rescue mode only if its current status is "Active" or "Shut down".

- There are only three actions available for the VM in the rescue mode: **Console**, **Exit rescue mode**, and **Delete**.
- If a rescue image has cloud-init installed, then the VM booted from it can be accessed with the same SSH key that was used for its creation.

To put a VM to the rescue mode, follow the steps:

1. On the **Virtual machines** screen, click the required VM on the list.
2. On the VM right pane, click the ellipsis button on the toolbar. Then, click **Enter rescue mode**.
3. In the **Enter rescue mode** window, select an image to rescue the VM with. By default, the initial image used for creating the VM is selected. Click **Enter**.



The machine status changes to "Rescue".

To return the VM to normal operation from the rescue mode, follow these steps:

1. On the **Virtual machines** screen, click the required VM on the list.
2. On the VM right pane, click **Exit rescue mode**.
3. In the **Exit rescue mode** window, click **Exit**. The VM will be automatically rebooted.

The VM status changes to "Active" and it boots from the original root disk.

Note

If the VM status changes to "Error" when exiting the rescue mode, you can reset its status with the **Reset state** action. The VM should then return to the "Rescue" status again.

4.1.8 Managing guest tools

This section explains how to install and uninstall the guest tools. This functionality is required for creating consistent snapshots of a running VM's disks (refer to "Managing volume snapshots" (p. 49)).

Installing guest tools

To install the guest tools inside a virtual machine, do the following:

- Inside a Windows VM:
 1. Download the Windows guest tools ISO image provided by your system administrator.
 2. Mount the image inside the VM.
 - On Windows 8 or Windows Server 2012 or newer, you can natively mount an ISO image. To do this, right-click the guest tools ISO image and select **Mount**.
 - On Windows 7 and Windows Server 2008, you need a third-party application to mount ISO images.
 3. Go to the mounted optical drive in Explorer and install the guest tools by running **setup.exe**.
 4. After the installation is complete, restart the VM.
- Inside a Linux VM:
 1. Download the Linux guest tools ISO image provided by your system administrator.
 2. Create a mount point for the optical drive with the guest tools image and run the installer:

```
# mkdir /mnt/cdrom
# mount <path_to_guest_tools_iso> /mnt/cdrom
# bash /mnt/cdrom/install
```

Note

Guest tools rely on the QEMU guest agent that is installed alongside the tools. The agent service must be running for the tools to work.

Uninstalling guest tools

If you find out that the guest tools are incompatible with some software inside a virtual machine, you can uninstall them by doing the following:

- Inside a Windows VM:
 1. Remove the QEMU device drivers from the device manager.

Important

Do not remove the VirtIO/SCSI hard disk driver and NetKVM network driver. Without the former, the VM will not boot; without the latter, the VM will lose network connectivity.

2. Uninstall the QEMU guest agent and guest tools from the list of installed applications.
3. Stop and delete **Guest Tools Monitor**:

```
> sc stop VzGuestToolsMonitor
> sc delete VzGuestToolsMonitor
```

4. Unregister **Guest Tools Monitor** from **Event Log**:

```
> reg delete HKLM\SYSTEM\CurrentControlSet\services\eventlog\Application\VzGuestToolsMonitor
```

5. Delete the autorun registry key for **RebootNotifier**:

```
> reg delete HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v \
VzRebootNotifier
```

6. Delete the **C:\Program Files\Qemu-ga** directory.

If **VzGuestToolsMonitor.exe** is locked, close all the Event Viewer windows. If it remains locked, restart the eventlog service:

```
> sc stop eventlog
> sc start eventlog
```

After removing the guest tools, restart the virtual machine.

- Inside a Linux VM:

1. Remove the packages:

- a. On RPM-based systems (CentOS and other):

```
# yum remove dkms-vzvirtio_balloon prl_nettool qemu-guest-agent-vz \
vz-guest-udev
```

- b. On DEB-based systems (Debian and Ubuntu):

```
# apt-get remove vzvirtio-balloon-dkms prl-nettool qemu-guest-agent-vz \
vz-guest-udev
```

If any of the packages listed above are not installed on your system, the command will fail. In this case, exclude these packages from the command and run it again.

2. Remove the files:

```
# rm -f /usr/bin/prl_backup /usr/share/qemu-ga/VERSION \
/usr/bin/install-tools \
/etc/udev/rules.d/90-guest_iso.rules /usr/local/bin/fstrim-static \
/etc/cron.weekly/fstrim
```

3. Reload the udev rules:

```
# udevadm control --reload
```

After removing guest tools, restart the virtual machine.

4.1.9 Troubleshooting virtual machines

If a VM fails to deploy, you can view the error message on the VM right pane. One of the possible root causes is that compute nodes lack free RAM or CPU resources to host the VM.

If a VM is stuck in a failed or transitional state, you can reset the VM to its last stable state: active, shut down or shelved. To do this, click **Reset state** on the VM right pane.

To troubleshoot a VM, you can also examine the VM console log by clicking **Download console log** on the VM right pane. The log will contain log messages only if logging is enabled inside the VM, otherwise the log will be empty.

To enable logging, do the following:

- In Linux VMs, enable the TTY1 and TTYS0 logging levels:
 1. Add the line `GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0"` to the file **/etc/default/grub**.
 2. Depending on the boot loader, run either

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

or

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

3. Reboot the VM.
- In Windows VMs, enable Emergency Management Services (EMS) console redirection:
 1. Start **Windows PowerShell** by using administrator privileges.
 2. In the PowerShell console, set the COM port and baud rate for EMS console redirection. As Windows VMs have only the COM1 port with the transmission rate of 9600 bps, run:

```
bcdedit /emssettings EMSPORT:1
```

3. Enable EMS for the current boot entry:

```
bcdedit /ems on
```

You may also enable driver status logging to see the list of loaded drivers. This can be useful for troubleshooting a faulty driver or long boot process. Do the following:

1. Start **System Configuration** by using administrator privileges.
2. In the **System Configuration** windows, open the **Boot** tab, and select the check boxes **OS boot information** and **Make all boot settings permanent**.
3. Confirm the changes and restart the system.

4.1.10 Deleting virtual machines

To remove a virtual machine, click its ellipsis button, and then click **Delete**. To remove multiple VMs at once, select them, and then click **Delete**.

A VM is removed along with its disks that have the **Delete on termination** option enabled during the VM deployment.

4.2 Managing security groups

A security group is a set of network access rules that control incoming and outgoing traffic to virtual machines assigned to this group. With security group rules, you can specify the type and direction of traffic that is allowed access to a virtual interface port. Traffic that does not satisfy any rule is dropped.

For each project, the **default** security group is automatically created in the compute cluster. This group allows all traffic on all ports for all protocols and cannot be deleted. When you attach a network interface to a VM, the interface is associated with the **default** security group, unless you explicitly select a custom security group.

You can assign one or more security groups to both new and existing virtual machines. When you add rules to security groups or remove them, the changes are enforced at runtime.

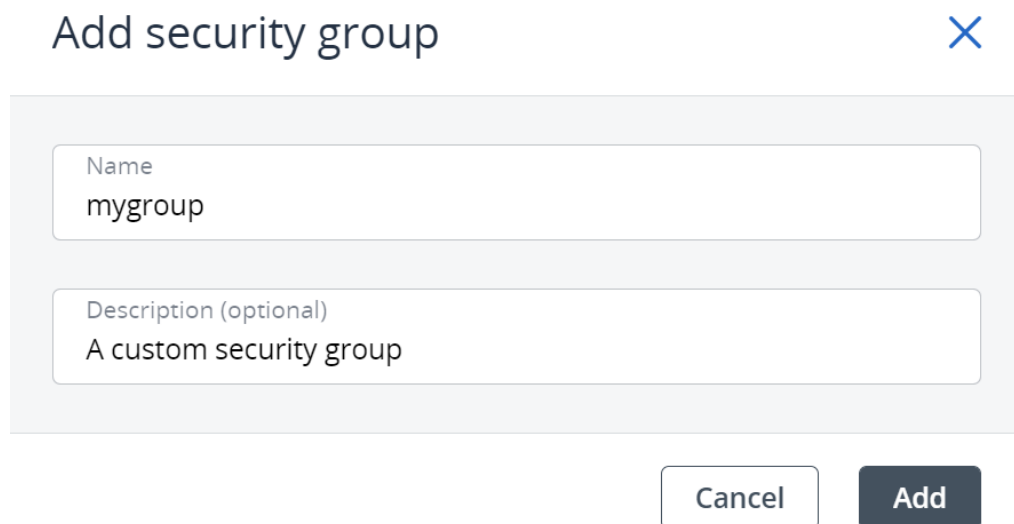
Important

You can manage only IPv4 security group rules.

4.2.1 Creating and deleting security groups

To create a security group, do the following:

1. On the **Security groups** screen, click **Add security group**.
2. In the **Add security group** window, specify a name and description for the group, and then click **Add**.



The screenshot shows a dialog box titled "Add security group" with a close button (X) in the top right corner. The dialog contains two text input fields. The first field is labeled "Name" and contains the text "mygroup". The second field is labeled "Description (optional)" and contains the text "A custom security group". At the bottom right of the dialog, there are two buttons: "Cancel" and "Add".

By default, the new security group will deny all incoming traffic and allow only outgoing traffic to assigned virtual machines.

To delete a security group, do the following:

1. On the **Security groups** screen, click the required security group.
2. On the group right pane, click **Delete**.
3. Click **Delete** in the confirmation window.

Note

You cannot delete a security group if it is assigned to a VM.

4.2.2 Managing security group rules

You can modify security groups by adding and removing rules. Editing rules is not available. If you need to change the existing rule, remove it and recreate with the required parameters.

To add a rule to a security group, do the following:

1. On the **Security groups** screen, click the security group to add a rule to.
2. On the group right pane, click **Add** in the **Inbound** or **Outbound** section to create a rule for incoming or outgoing traffic.
3. Specify the rule parameters:
 - a. Select a protocol from the list or enter a number from 0 to 255.
 - b. Enter a single port or a port range. Some protocols already have a predefined port range. For example, the port for SSH is 22.
 - c. Select a predefined subnet CIDR or an existing security group.

Protocol ⓘ	Port range	Source ⓘ		
SSH ▾	22	0.0.0.0/0 ▾	✓	✕

4. Click the check mark to save the changes.

As soon as the rule is created, it is applied to all of the virtual machines assigned to the security group.

To remove a rule from a security group, do the following:

1. On the **Security groups** screen, click the required security group.
2. On the group right pane, click the bin icon next to a rule you want to remove.

As soon as the rule is removed, this change is applied to all of the virtual machines assigned to the security group.

4.2.3 Changing security group assignment

When you create a VM, you select security groups for the VM network interfaces. You can also change assigned security groups later.

You can view all of the virtual machines assigned to a security group, as follows:

1. On the **Security groups** screen, click the required security group.
2. On the group right pane, navigate to the **Assigned VMs** tab. All the assigned virtual machines will be shown along with their status.

You can click the VM name to go to the VM **Overview** pane and change the security group assignment for its network interfaces.

To assign a security group to a virtual machine, do the following:

1. On the **Virtual machines** screen, click the required virtual machine.
2. On the **Overview** tab, click the pencil icon in the **Networks** section.
3. Click the ellipsis icon next to the network interface to assign a security group to, and then click **Edit**.
4. In the **Edit network interface** window, go to the **Security groups** tab.
5. Select one or more security groups from the drop-down list, and then click **Save**.

Important

You cannot configure security groups if spoofing protection is disabled or IP address management is disabled for the selected network.

The rules from chosen security groups will be applied at runtime.

4.3 Managing Kubernetes clusters

Self-service users can deploy ready-to-use Kubernetes clusters with persistent storage for managing containerized applications.

A Kubernetes cluster includes the following components:

Component	Name and version
Underlying OS	Fedora 32 CoreOS
Container runtime	Docker 19.03.11
Network plugin	Flannel with VXLAN

4.3.1 Creating and deleting Kubernetes clusters

The prerequisites for creating a Kubernetes cluster are:

- The Kubernetes-as-a-service component. It can be deployed along with the compute cluster or later (refer to the Administrator Guide).
- A network that will interconnect the Kubernetes master and worker nodes. It can be either a shared physical network or a virtual network linked to a physical one via a virtual router. The virtual network needs to have a gateway and a DNS server specified.
- An SSH key that will be installed on both the master and worker nodes.
- Enough resources for all of the Kubernetes nodes, taking their flavors into account.

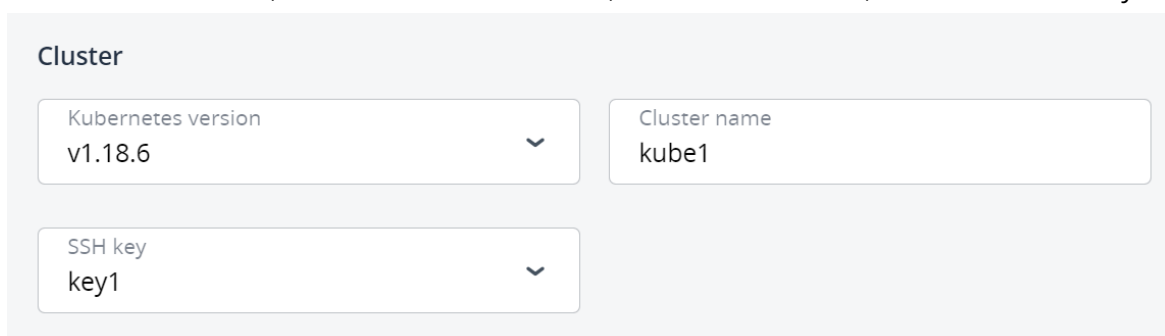
Important

It is also required that the network where you create a Kubernetes cluster does not overlap with these default networks:

- 10.100.0.0/24—Used for pod-level networking
 - 10.254.0.0/16—Used for allocating Kubernetes cluster IP addresses
-

To create a Kubernetes cluster, do the following:

1. Go to the **Kubernetes clusters** screen, and then click **Create** on the right. A window will open where you can set your cluster parameters
2. In the **Cluster** section, select a Kubernetes version, enter a cluster name, and select an SSH key.



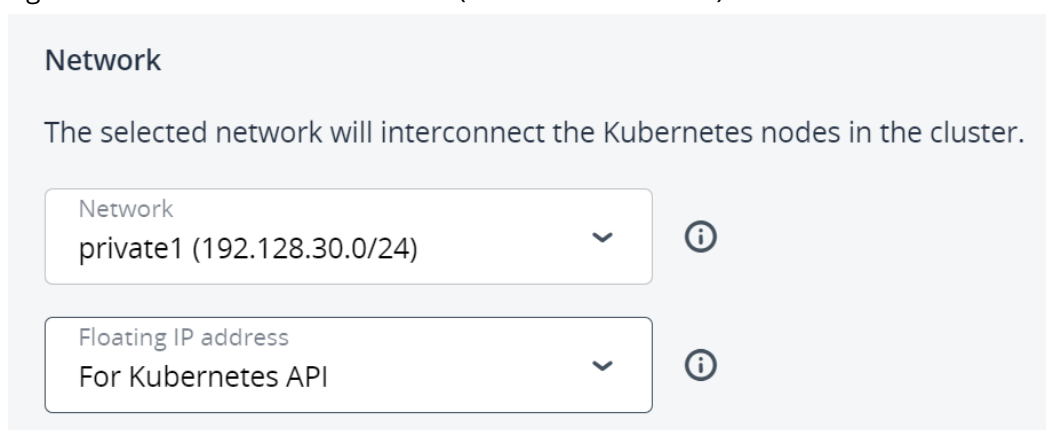
Cluster

Kubernetes version
v1.18.6

Cluster name
kube1

SSH key
key1

3. In the **Network** section, select a network that will interconnect the Kubernetes nodes in the cluster. If you select a virtual network, decide whether you need access to your Kubernetes cluster via a floating IP address:
 - If you select **None**, you will not have access to the Kubernetes API.
 - If you select **For Kubernetes API**, a floating IP address will be assigned to the master node or to the load balancer if the master node is highly available.
 - If you select **For Kubernetes API and nodes**, floating IP addresses will be additionally assigned to all of the Kubernetes nodes (masters and workers).



Network

The selected network will interconnect the Kubernetes nodes in the cluster.

Network
private1 (192.128.30.0/24)

Floating IP address
For Kubernetes API

4. In the **Master node** section, select a flavor, and then choose whether or not to enable high availability for the master node. If you enable high availability, three master node instances will be created. They will work in the Active/Active mode.

Master node

High availability ⓘ

Flavor
medium — 2 vCPUs, 4 GiB RAM

- In the **Container volume** section, select a storage policy, and then enter size for volumes on both master and worker nodes.

Container volume

These parameters apply to both master and worker nodes.

Storage policy
default

Disk size (GiB)
10

Min. 3 GiB,
Max. 512 TiB

- In the **Default worker group** section, set a number of workers to create, and then select a flavor for each worker.

Default worker group

Number of workers 3

Flavor
small — 1 vCPU, 2 GiB RAM

- Click **Create**.

Creation of the Kubernetes cluster will start. The master and worker nodes will appear on the **Virtual machines** screen, while their volumes will show up on the **Volumes** screen.

After the cluster is ready, click **Kubernetes access** for instructions on how you can access the dashboard.

To delete a Kubernetes cluster, click it on the **Kubernetes clusters** screen and click **Delete**. The master and worker VMs will be deleted along with their volumes.

4.3.2 Managing Kubernetes worker groups

Important

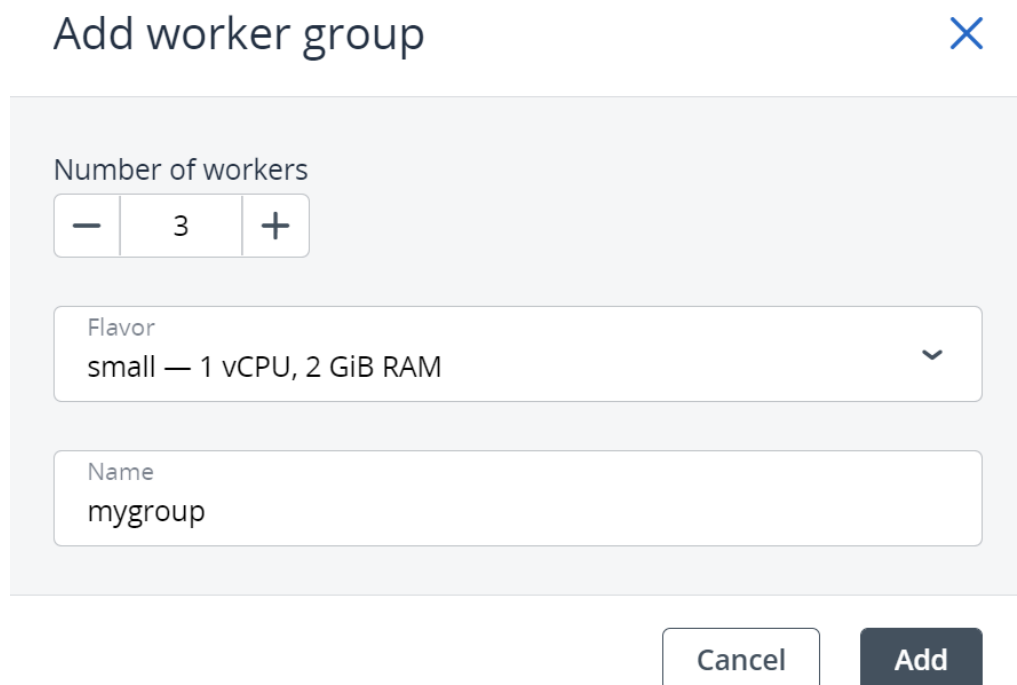
Worker groups are not available for Kubernetes version 1.15.x.

To meet system requirements of applications running in Kubernetes clusters, you can have worker nodes with different number of CPUs and amount of RAM. Creating workers with different flavors is possible by using worker groups.

When creating a Kubernetes cluster, you can specify the configuration of only one worker group, the default worker group. After the cluster is created, add as many worker groups as you need. If required, you can also edit the number of workers in a group later.

To add a worker group, do the following:

1. On the **Kubernetes clusters** screen, click a Kubernetes cluster.
2. On the cluster right pane, navigate to the **Groups** tab.
3. In the **Workers** section, click **Add**.
4. In the **Add worker group** window, set a number of workers to create, select a flavor for each worker, and then specify a name for the group. Then, click **Add**.



The screenshot shows a modal dialog titled "Add worker group" with a close button (X) in the top right corner. The dialog contains three input fields: "Number of workers" with a spinner control set to 3, "Flavor" with a dropdown menu showing "small — 1 vCPU, 2 GiB RAM", and "Name" with a text input field containing "mygroup". At the bottom of the dialog, there are two buttons: "Cancel" and "Add".

When the worker group is created, you can assign pods to these worker nodes, as explained in "Assigning Kubernetes pods to specific nodes" (p. 38).

To edit the number of workers in a group, do the following:

1. On the Kubernetes cluster right pane, navigate to the **Groups** tab.
2. In the **Workers** section, click the pencil icon for the default worker group or the ellipsis icon for all other groups, and then select **Edit**.

Important

Only the user who created a Kubernetes cluster can edit its worker groups.

3. In the **Edit worker group** window, change the number of workers, and then click **Save**.

To delete a worker group, click the ellipsis icon next to the required worker group, and then select **Delete**. The worker group will be deleted along with all of its workers. After the deletion, the worker group data will be lost.

Note

The default worker group cannot be deleted.

4.3.3 Using persistent volumes for Kubernetes pods

Kubernetes allows using compute volumes as persistent storage for pods. Persistent volumes (PV) exist independently of pods, meaning that such a volume persists after the pod it is mounted to is deleted. This PV can be mounted to other pods for accessing data stored on it. You can provision PVs dynamically, without having to create them manually, or statically, using volumes that exist in the compute cluster.

Creating storage classes

In Virtuozzo Hybrid Infrastructure, storage classes map to compute storage policies defined in the admin panel. Creating a storage class is required for all storage operations in a Kubernetes cluster.

To create a storage class, click **+ Create** on the Kubernetes dashboard and specify a YAML file that defines this object. For example:

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: mysc
provisioner: cinder.csi.openstack.org
parameters:
  type: default
```

This manifest describes the storage class `mysc` with the storage policy `default`. The storage policy must exist in the compute cluster and be specified in the storage quotas to the current project.

Dynamically provisioning persistent volumes

Persistent volumes can be dynamically provisioned via persistent volume claims (PVC). A PVC requests for a PV of a specific storage class, access mode, and size. If a suitable PV exists in the cluster, it is bound to the claim. If suitable PVs do not exist but can be provisioned, a new volume is created and bound to the claim. Kubernetes uses a PVC to obtain the PV backing it and mounts it to the pod.

Important

A pod and the persistent volume claim it uses must exist in the same namespace.

You can dynamically provision a PV to a pod as follows:

1. Access the Kubernetes cluster via the dashboard. Click **Kubernetes access** for instructions.
2. On the Kubernetes dashboard, create a storage class, as described in "Creating storage classes" (p. 30).
3. Create a persistent volume claim. To do it, click + **Create** and specify the following YAML file:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: mysc
```

This manifest specifies the persistent volume claim `mypvc` that requests from the storage class `mysc` a volume of at least 10 GiB that can be mounted in the read/write mode by a single node. Creation of the PVC triggers dynamic provisioning of a persistent volume that satisfies the claim's requirements. Kubernetes then binds it to the claim.

Details

Name: mypvc

Name space: default

Annotations: pv.kubernetes.io/bind-completed: yes

pv.kubernetes.io/bound-by-controller: yes

volume.beta.kubernetes.io/storage-provisioner: csi-cinderpl..

Creation Time: 2020-02-04T14:38 UTC

Status: Bound

Volume: [pvc-b1b257ba-5588-4989-8517-006dc41e6629](#)

Access modes: ReadWriteOnce

Storage class: [mysc](#)

4. Create a pod and specify the PVC as its volume. To do it, click + **Create** and enter the following YAML file:

```
apiVersion: v1
kind: Pod
```

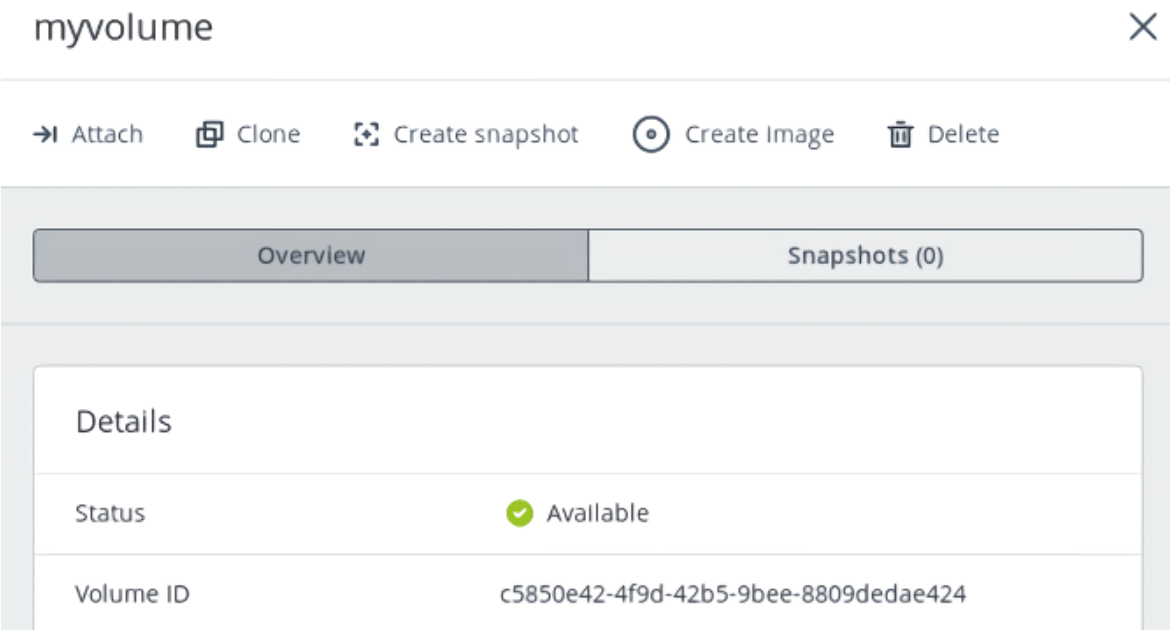
```
metadata:
  name: nginx
spec:
  containers:
  - image: nginx
    imagePullPolicy: IfNotPresent
    name: nginx
    ports:
    - containerPort: 80
      protocol: TCP
    volumeMounts:
    - mountPath: /var/lib/www/html
      name: mydisk
  volumes:
  - name: mydisk
    persistentVolumeClaim:
      claimName: mypvc
      readOnly: false
```

This configuration file describes the pod `nginx` that uses the persistent volume claim `mypvc`. The persistent volume bound to the claim will be accessible at `/var/lib/www/html` inside the `nginx` container.

Statically provisioning persistent volumes

You can mount existing compute volumes to pods using static provisioning of persistent volumes. To mount a compute volume, do the following:

- 1. In the self-service panel, obtain the ID of the desired volume.



- 2. Access the Kubernetes cluster via the dashboard. Click **Kubernetes access** for instructions.

3. On the Kubernetes dashboard, create a storage class, as described in "Creating storage classes" (p. 30).
4. Create a persistent volume. To do it, click **+ Create** and specify the following YAML file:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    pv.kubernetes.io/provisioned-by: cinder.csi.openstack.org
  name: mypv
spec:
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 10Gi
  csi:
    driver: cinder.csi.openstack.org
    fsType: ext4
    volumeHandle: c5850e42-4f9d-42b5-9bee-8809dedae424
  persistentVolumeReclaimPolicy: Delete
  storageClassName: mysc
```

This manifest specifies the persistent volume `mypv` from the storage class `mysc` that has 10 GiB of storage and access mode that allows it to be mounted in the read/write mode by a single node. The PV `mypv` uses the compute volume with the ID `c5850e42-4f9d-42b5-9bee-8809dedae424` as backing storage.

5. Create a persistent volume claim. Before you define the PVC, make sure the PV is created and has the status "Available". The existing PV must meet the claim's requirements to storage size, access mode and storage class. Click **+ Create** and specify the following YAML file:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: mysc
```

Once the persistent volume claim `mypvc` is created, the volume `mypv` is bound to it.

Details

Name: mypvc

Name space: default

Annotations: pv.kubernetes.io/bind-completed: yes

pv.kubernetes.io/bound-by-controller: yes

Creation Time: 2020-02-04T14:53 UTC

Status: Bound

Volume: [mypv](#)

Access modes: ReadWriteOnce

Storage class: [mysc](#)

6. Create a pod and specify the PVC as its volume. Use the example from Step 4 in "Dynamically provisioning persistent volumes" (p. 30).

In the self-service panel, the compute volume will be mounted to the virtual machine running the Kubernetes pod.

myvolume ✕

◀ Force detach 📷 Create snapshot

Overview Snapshots (0)

Details

Status	▶ In use
Volume ID	c5850e42-4f9d-42b5-9bee-8809dedae424
Usage	133 MIB of 10 GIB
Attached to	kube1-igjmbdx5lrgg-minion-1

Making Kubernetes deployments highly available

If a node that hosts a Kubernetes pod fails or becomes unreachable over the network, the pod is stuck in a transitional state. In this case, the pod's persistent volumes are not automatically detached, and it prevents the pod redeployment on another worker node. To make your Kubernetes applications highly available, you need to enforce the pod termination in the event of node failure by adding rules to the pod deployment.

To terminate a stuck pod, add the following lines to the `spec` section of the deployment configuration file:

```
terminationGracePeriodSeconds: 0
tolerations:
- effect: NoExecute
  key: node.kubernetes.io/unreachable
  operator: Exists
  tolerationSeconds: 2
- effect: NoExecute
  key: node.kubernetes.io/not-ready
  operator: Exists
  tolerationSeconds: 2
```

If the node's state changes to "NotReady" or "Unreachable", the pod will be automatically terminated in 2 seconds.

The entire YAML file of a deployment may look as follows:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      terminationGracePeriodSeconds: 0
      tolerations:
      - effect: NoExecute
        key: node.kubernetes.io/unreachable
        operator: Exists
        tolerationSeconds: 2
      - effect: NoExecute
        key: node.kubernetes.io/not-ready
        operator: Exists
```

```

    tolerationSeconds: 2
  containers:
  - image: nginx
    imagePullPolicy: IfNotPresent
    name: nginx
    ports:
    - containerPort: 80
      protocol: TCP
    volumeMounts:
    - mountPath: /var/lib/www/html
      name: mydisk
  volumes:
  - name: mydisk
    persistentVolumeClaim:
      claimName: mypvc

```

The manifest above describes the deployment `nginx` with one pod that uses the persistent volume claim `mypvc` and will be automatically terminated in 2 seconds in the event of node failure.

4.3.4 Creating external load balancers in Kubernetes

In Kubernetes, you can create a service with an external load balancer that provides access to it from public networks. The load balancer will receive a publicly accessible IP address and route incoming requests to the correct port on the Kubernetes cluster nodes.

To create a service with an external load balancer, do the following:

1. Access the Kubernetes cluster via the dashboard. Click **Kubernetes access** for instructions.
2. On the Kubernetes dashboard, create a deployment and service of the **LoadBalancer** type. To do it, click **+ Create** and specify a YAML file that defines these objects. For example:
 - If you have deployed the Kubernetes cluster in a shared physical network, specify the following manifest:

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx

```

```

    ports:
      - containerPort: 80
  ---
kind: Service
apiVersion: v1
metadata:
  name: load-balancer
  annotations:
    service.beta.kubernetes.io/openstack-internal-load-balancer: "true"
spec:
  selector:
    app: nginx
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP

```

The manifest above describes the deployment `nginx` with a replica set of two pods and the service `load-balancer` with the `LoadBalancer` type. The annotation used for the service indicates that the load balancer will be internal.

Once the load balancer is created, it will be allocated an IP address from the shared physical network and can be accessed at this external endpoint.

Details

Name: load-balancer	Connection
Namespace: default	Cluster IP: 10.254.147.243
Annotations: <code>service.beta.kubernetes.io/openstack-internal-load-balancer: true</code>	Internal endpoints: load-balancer:80 TCP load-balancer:32069 TCP
Creation Time: 2020-05-26T14:37 UTC	External endpoints: 10.94.156.196:80
Label selector: <code>app: nginx</code>	
Type: LoadBalancer	
Session Affinity: None	

- If you have deployed the Kubernetes cluster in a virtual network linked to a physical one via a virtual router, you can use the YAML file above without the `annotations` section for the `load-balancer` service. The created load balancer will receive a floating IP address from the physical network and can be accessed at this external endpoint.

The load balancer will also appear in the self-service panel, where you can monitor its performance and health. For example:

Load balancers

<input type="checkbox"/>	Name ↑	Status ↓	IP address ↓	Floating IP ↓	Members state	Members ... ↓	
<input type="checkbox"/>	kube_service_d66...	Active	192.168.10.201	10.94.129.73		2	...

4.3.5 Assigning Kubernetes pods to specific nodes


By using worker groups, you can assign a pod in Kubernetes to specific nodes. When you create a custom worker group, its nodes are added a label with the group name. If you want your pod to be scheduled on a node from a specific worker group, add the node selector section with the node label to the pod's configuration file.

To create a pod that will be scheduled on a specific node, click **+ Create** on the Kubernetes dashboard and specify a YAML file that defines this object. For example:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    env: test
spec:
  containers:
  - name: nginx
    image: nginx
    imagePullPolicy: IfNotPresent
  nodeSelector:
    magnum.openstack.org/nodegroup: mygroup
```

This manifest describes the pod `nginx` that will be assigned to a node from the node group `mygroup`.

When the pod is created, check that the hosting node belongs to the specified worker group.

Pods								
Name	Namespace	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
 <code>nginx</code>	default	<code>env: test</code>	kube1-mygroup-vogevh53o-node-1	Running	0	-	-	a minute ago

4.4 Managing images

Virtuozzo Hybrid Infrastructure allows you to upload ISO images and templates that can be used to create VM volumes:

- An ISO image is a typical OS distribution that needs to be installed on disk.
- A template is a ready volume in the QCOW2 or RAW format with an installed operating system and applications and a set minimum size. Many OS vendors offer templates of their operating systems under the name "cloud images". You can use a cloud image from the [OS official repository](#) or prepare your own template.

For a list of guest OSes that are supported in virtual machines, refer to "Supported guest operating systems" (p. 11).

4.4.1 Uploading and removing images

To add an image, do the following:

1. On the **Images** screen, click **Add image**.
2. In the **Add image** window, do the following:
 - a. Click **Browse** and select a file in one of the supported formats: .iso, .img, .qcow2, .raw.
 - b. Specify an image name to be shown in the admin panel.
 - c. Select the correct OS type from the drop-down list.

Important

The OS type affects VM parameters such as hypervisor settings. VMs created from an image with an incorrect OS type may not work correctly, for example, they may crash.

3. Click **Add** to start uploading the image. Upload progress will be shown in the bottom right corner.

Important

Do not reload the page while the image is being uploaded or the process will fail.

Note

If you select an image assigned to a placement, the created virtual machine will also be included in this placement. For more information, see the Administrator Guide.

To download or remove an image, click the ellipsis button next to it and click the desired action.

4.4.2 Creating volumes from images

You can create volumes from both ISO images and templates. Do the following:

1. Go to the **Images** screen, and then click the required image.
2. On the image panel, click **Create volume**.
3. In the **Create volume** window, specify the volume name, size, and select a storage policy.

Create volume ✕

Name
vol1

Size (GiB) **10** Min. 1 GiB,
Max. 512 TiB

Storage policy
default ▾

Image: **cirros**

4. Click **Create**.

The new volume will appear on the **Volumes** screen.

4.4.3 Mounting ISO images to virtual machines

Note

This feature is supported only for Linux virtual machines.

To mount an ISO image to a Linux VM, do the following:

1. Create a volume from the ISO image, as described in "Creating volumes from images" (p. 39).
2. Attach the resulting volume to the desired VM, as described in "Attaching and detaching volumes" (p. 47).

The mounted disk will appear inside the Linux VM.

4.4.4 Preparing templates

You may need to create a template in these cases:

- To rescue a virtual machine
- To create a VM accessible via SSH

- To create a VM customizable with user data

To prepare a template, you need to install cloud-init and OpenSSH Server in an existing boot volume, and then create an image from this volume.

Preparing Linux templates

As all Linux guests have OpenSSH Server preinstalled by default, you only need to make sure a Linux template has cloud-init installed.

The easiest way to get a Linux template with cloud-init installed is to obtain it from [its official repository](#). You can also create a Linux template from an existing boot volume.

Preparing Windows templates

Windows guests have neither Cloudbase-Init nor OpenSSH Server preinstalled by default. You need to install and configure them manually.

Do the following:

1. Log in to a Windows VM.
2. Create a new administrator account that will be used for SSH connections and log in with it.
3. To install and configure OpenSSH Server:
 - a. Run Windows PowerShell with administrator privileges and set the execution policy to unrestricted to be able to run scripts:

```
> Set-ExecutionPolicy Unrestricted
```

- b. Download OpenSSH Server (for example, from the [GitHub repository](#)), extract the archive into the **C:\Program Files** directory, and then install it by running:

```
> & 'C:\Program Files\OpenSSH-Win64\install-sshd.ps1'
```

- c. Start the sshd service and set its startup type to "Automatic":

```
> net start sshd  
> Set-Service sshd -StartupType Automatic
```

- d. Open TCP port 22 for the OpenSSH service in the Windows Firewall:

- On Windows 8.1, Windows Server 2012, and newer versions, run

```
> New-NetFirewallRule -Protocol TCP -LocalPort 22 -Direction Inbound -  
Action Allow -DisplayName OpenSSH
```

- On Windows 7, Windows Server 2008, and Windows Server 2008 R2, run

```
> netsh advfirewall firewall add rule name=sshd dir=in action=allow  
protocol=TCP localport=22
```

- e. Open the **C:\ProgramData\ssh\sshd_config** file:

```
> notepad 'C:\ProgramData\ssh\sshd_config'
```

Comment out the following lines at the end of the file:

```
#Match Group administrators  
#AuthorizedKeysFile __PROGRAMDATA__/ssh/administrators_authorized_keys
```

Save the changes.

- f. Create the **.ssh** directory in **C:\Users\<current_user>** and an empty **authorized_keys** file inside it:

```
> cd C:\Users\<current_user>  
> mkdir .ssh  
> notepad .\.ssh\authorized_keys
```

Remove the **.txt** extension from the created file:

```
> move .\.ssh\authorized_keys.txt .\.ssh\authorized_keys
```

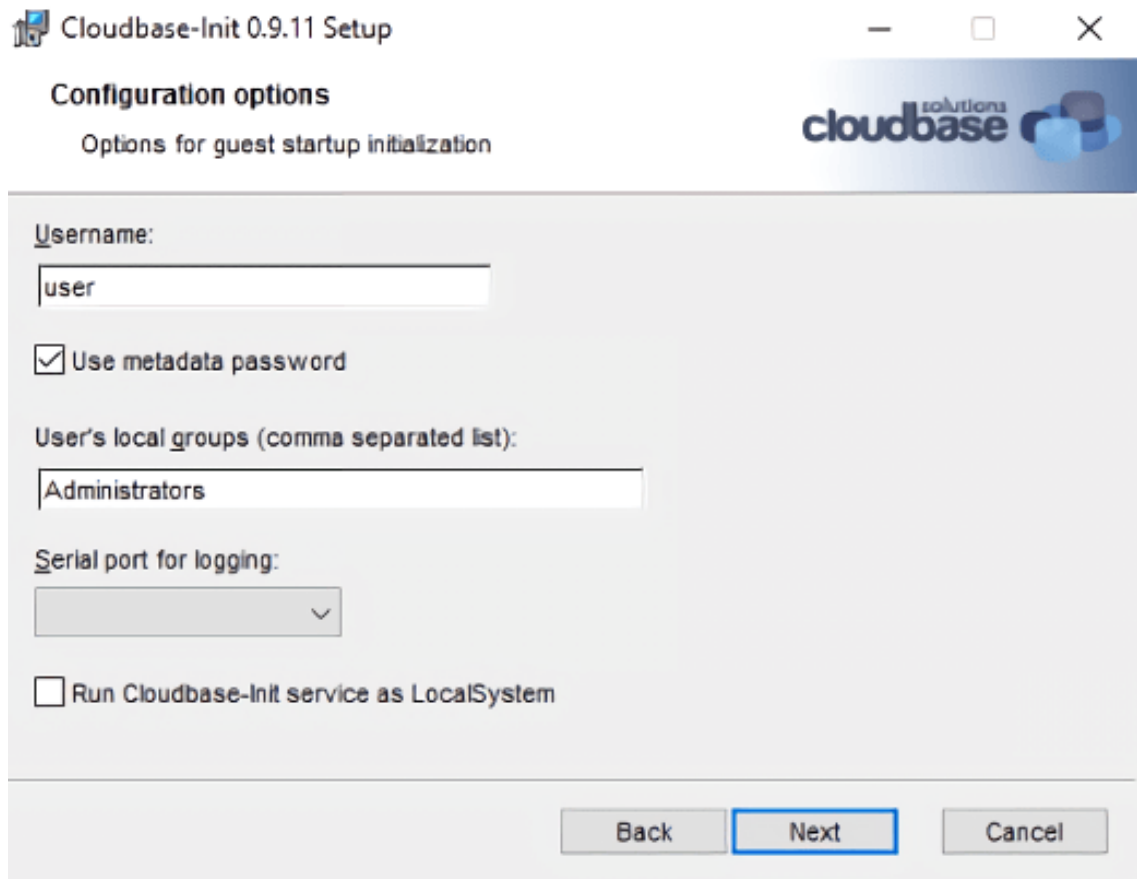
- g. Modify the permissions for the created file to disable inheritance:

```
> icacls .\.ssh\authorized_keys /inheritance:r
```

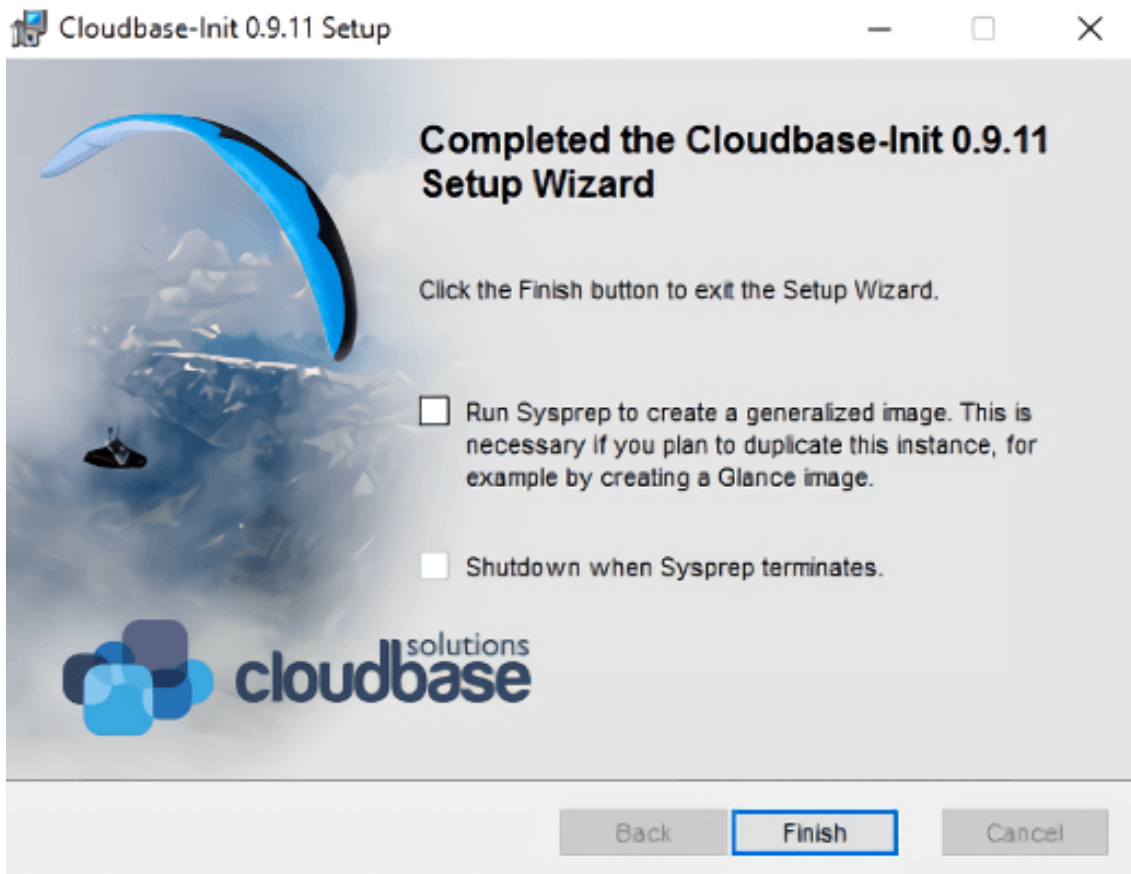
4. Download Cloudbase-Init (for example, from the [official site](#)), launch the installation, and then follow the on-screen instructions:
 - a. In the **Configuration options** window, enter the current username in the **Username** field:

Important

The user account password will be reset on the next VM startup. You will be able to log in with this account by using the key authentication method or you can set a new password with a customization script.



- b. When the installation is complete, do not run Sysprep and click **Finish**:



- c. Run Windows PowerShell with administrator privileges and open the file **C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf\cloudbase-init.conf**:

```
> notepad 'C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf\cloudbase-init.conf'
```

Add `metadata_services` and `plugins` on two lines:

```
metadata_services=\
cloudbaseinit.metadata.services.configdrive.ConfigDriveService,\
cloudbaseinit.metadata.services.httpservice.HttpService\
plugins=cloudbaseinit.plugins.common.mtu.MTUPlugin,\
cloudbaseinit.plugins.windows.ntpclient.NTPClientPlugin,\
cloudbaseinit.plugins.common.sethostname.SetHostNamePlugin,\
cloudbaseinit.plugins.windows.createuser.CreateUserPlugin,\
cloudbaseinit.plugins.common.networkconfig.NetworkConfigPlugin,\
cloudbaseinit.plugins.windows.licensing.WindowsLicensingPlugin,\
cloudbaseinit.plugins.common.sshpublickeys.SetUserSSHPublicKeysPlugin,\
cloudbaseinit.plugins.windows.extendvolumes.ExtendVolumesPlugin,\
cloudbaseinit.plugins.common.setuserpassword.SetUserPasswordPlugin,\
cloudbaseinit.plugins.common.userdata.UserDataPlugin,\
cloudbaseinit.plugins.windows.winrmlistener.ConfigWinRMListenerPlugin,\
cloudbaseinit.plugins.windows.winrmcertificateauth.\
ConfigWinRMCertificateAuthPlugin,\
cloudbaseinit.plugins.common.localscripts.LocalScriptsPlugin
```

Note

Make sure to remove all backslashes in the lines above.

Save the changes.

5. Stop the VM.
6. After the VM is shut down, convert its boot volume to a template (refer to "Creating images from volumes" (p. 48)).

4.5 Managing volumes

A volume in Virtuozzo Hybrid Infrastructure is a virtual disk drive that can be attached to a virtual machine. The integrity of data in volumes is protected by the redundancy mode specified in the storage policy.

Note

Additional virtual disks attached to VMs need to be initialized inside the guest OS by standard means before they can be used.

4.5.1 Creating, editing, and removing volumes

To create a volume, do the following:

1. On the **Volumes** screen, click **Create volume**.

Create volume ✕

Name
vol1

Size (GiB)
1

Min. 1 GiB,
Max. 512 TiB

Storage policy
default ▾

Cancel Create

2. In the **Create volume** window, specify a volume name and size in gigabytes, select a storage policy, and then click **Create**.

To edit a volume, select it and click the pencil icon next to a parameter you need to change. Note the following restrictions:

- You cannot shrink volumes.
- To extend volumes that are in use, stop the VM first.
- You cannot change the volume redundancy type.

To remove a volume, click its ellipsis button then click **Delete**. To remove multiple volumes at once, select them and click **Delete**. To remove a volume that is in use, detach it first.

Note

A volume is removed along with all of its snapshots.

4.5.2 Cloning volumes

You can clone volumes that are not attached to VMs or attached to stopped VMs. To clone a volume, do the following:

1. On the **Volumes** screen, click a volume.
2. In the volume right pane, click **Clone**.

3. In the **Clone volume** window, specify a volume name, size, and storage policy. Click **Clone**.

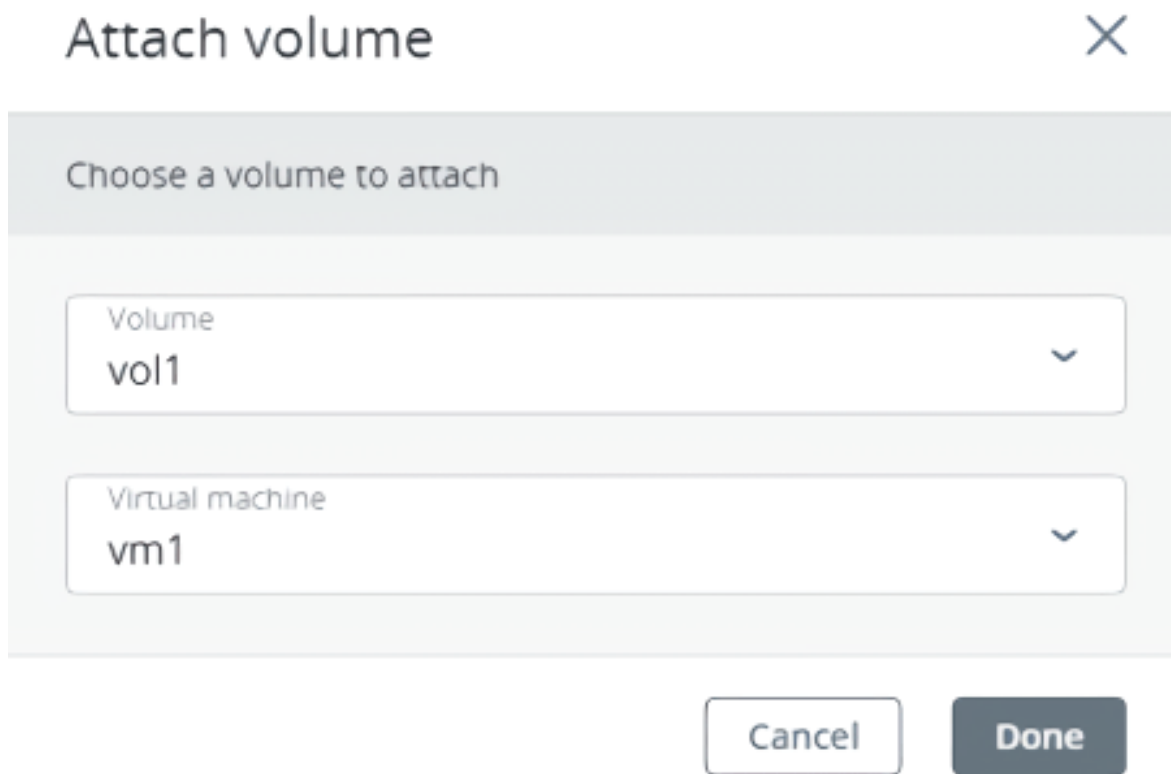
The screenshot shows a 'Clone volume' dialog box. The title bar contains the text 'Clone volume' and a close button (X). The dialog body contains three input fields: 'Name' with the value 'Clone_vol1', 'Size (GiB)' with the value '1' and a range 'Min. 1 GiB, Max. 512 TiB', and 'Storage policy' with the value 'default'. At the bottom are 'Cancel' and 'Clone' buttons.

4.5.3 Attaching and detaching volumes

You can add a writable virtual disk drive to a VM by attaching a volume to the VM. To do this:

1. On the **Volumes** screen, click the ellipsis button next to an unused volume, and then click **Attach** in the context menu.

2. In the **Attach volume** window, select the VM from the drop-down list, and then click **Done**.



To detach a volume, do the following:

1. Click the ellipsis button next to the volume that is in use.
2. If the VM is not running, click **Detach**.
3. If the VM is running, you can only click **Force detach** to immediately detach the volume.

Warning!

There is a risk of data loss.

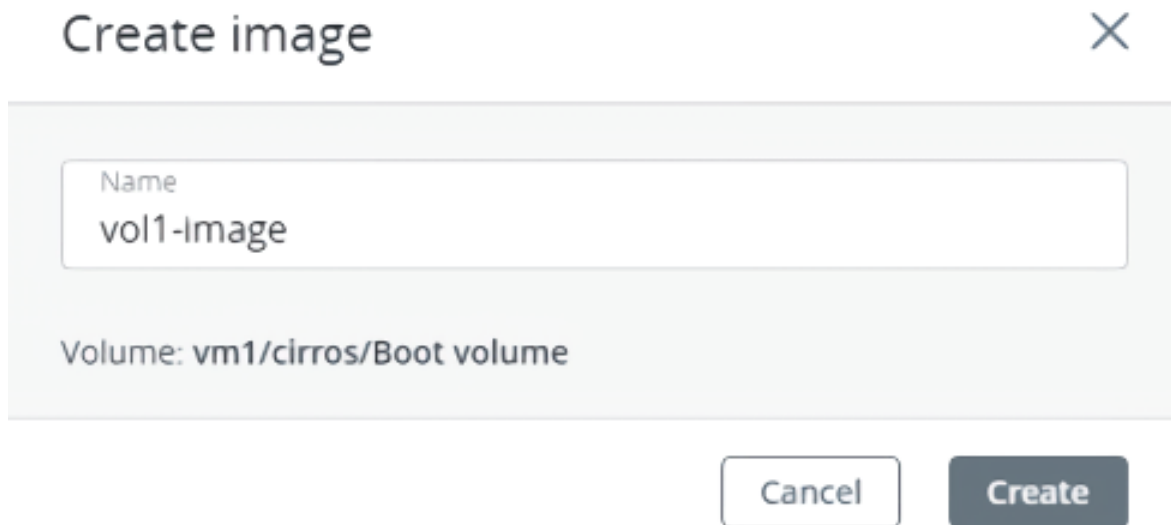
4.5.4 Creating images from volumes

To create multiple VMs with the same boot volume, you can create a template from an existing boot volume and deploy VMs from it. Make sure to install cloud-init in the volume before creating the image, as described in "Preparing templates" (p. 40).

Do the following:

1. Power off the VM that the original volume is attached to.
2. Switch to the **Volumes** screen, click volume's ellipsis button and select **Create image**.

3. In the **Create image** window, enter an image name, and then click **Create**.



Create image ×

Name
vol1-image

Volume: vm1/cirros/Boot volume

Cancel Create


The new image will appear on the **Images** screen.

4.5.5 Managing volume snapshots

You can save the current state of a VM file system or user data by creating a snapshot of a volume. A snapshot of a boot volume may be useful, for example, before updating VM software. If anything goes wrong, you will be able to revert the VM to a working state at any time. A snapshot of a data volume can be used for backing up user data and testing purposes.

To create a snapshot of a volume, do the following:

1. On the **Volumes** screen, click a volume.
2. In the volume right pane, switch to **Snapshots**, and then click **Create snapshot**.



vm1/cirros/Boot volume ×

Create snapshot

Overview Snapshots

Details

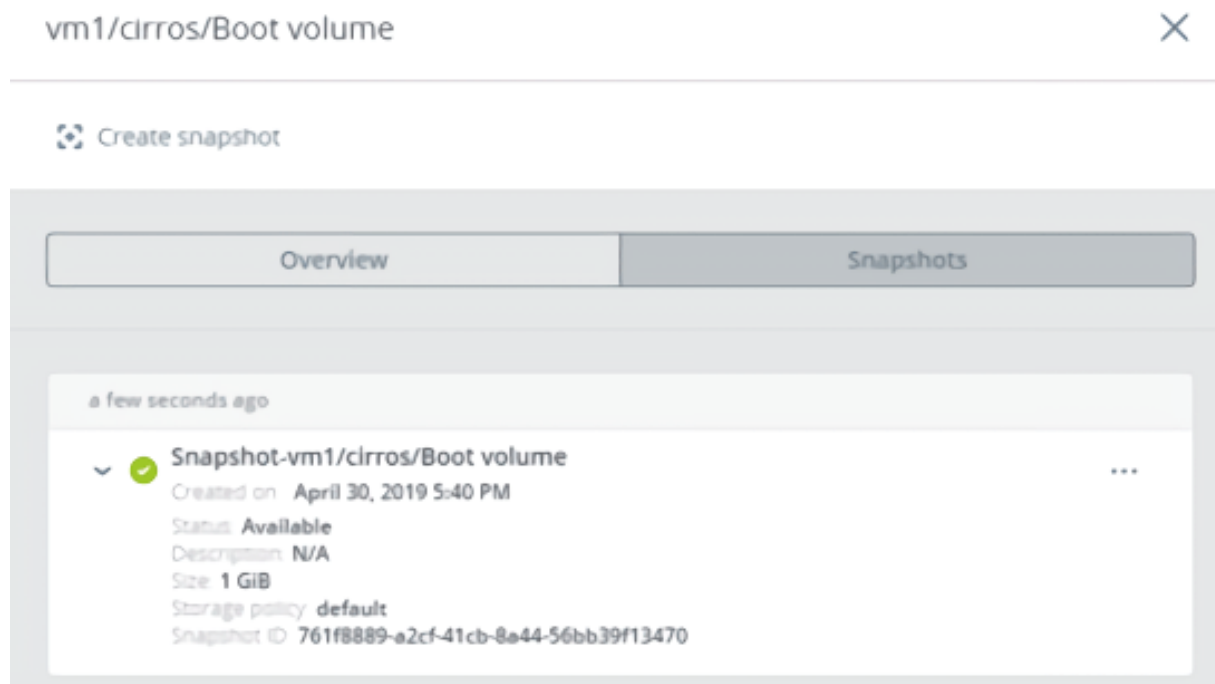
Status ▶ In use

Volume ID 821ad5f5-deeb-4412-b322-18d662ba1b57

Note

To create a consistent snapshot of a running VM's volume, make sure the guest tools are installed in the VM. The QEMU guest agent included in the guest tools image automatically quiesces the filesystem during snapshotting. For instructions on installing the guest tools, refer to "Installing guest tools" (p. 20).

Once the snapshot is created, you can see and manage it on the **Snapshots** tab on the volume right pane.



You can do the following:

- Create a new volume from the snapshot.
- Create a template from the snapshot.
- Discard all changes that have been made to the volume since the snapshot was taken. This action is available only for VMs with the "Shut down" and "Shelved offloaded" statuses.

Warning!

As each volume has only one snapshot branch, all snapshots created after the snapshot you are reverting to will be deleted. If you want to save a subsequent snapshot before reverting, create a volume or an image from it first.

- Change the snapshot name and description.
- Reset the snapshot stuck in an "Error" state or transitional state to the "Available" state.
- Remove the snapshot.

To perform these actions, click the ellipsis button next to a snapshot, and then click the corresponding action.

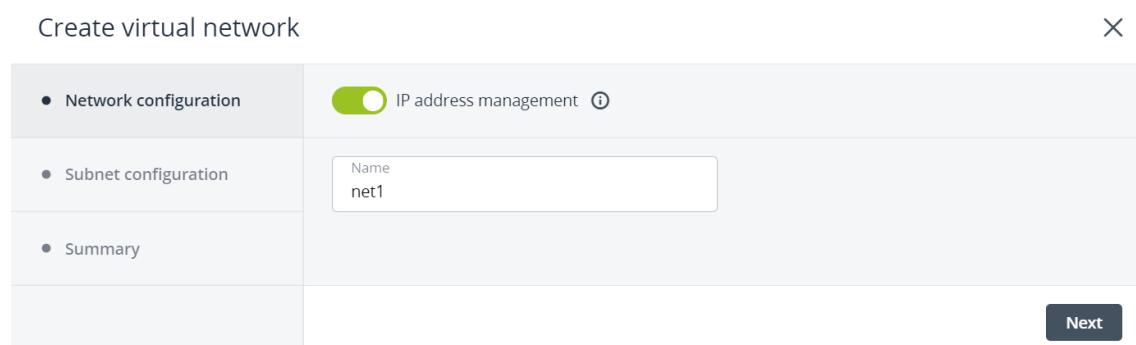
4.6 Managing virtual networks

To add a new virtual network, do the following:

1. On the **Networks** screen, click **Create virtual network**.
2. On the **Network configuration** step, do the following:
 - a. Enable or disable IP address management:
 - With IP address management enabled, VMs connected to the network will automatically be assigned IP addresses from allocation pools by the built-in DHCP server and use custom DNS servers. Additionally, spoofing protection will be enabled for all VM network ports by default. Each VM network interface will be able to accept and send IP packets only if it has IP and MAC addresses assigned. You can disable spoofing protection manually for a VM interface, if required.
 - With IP address management disabled, VMs connected to the network will obtain IP addresses from the DHCP servers in that network, if any. Also, spoofing protection will be disabled for all VM network ports, and you cannot enable it manually. This means that each VM network interface, with or without assigned IP and MAC addresses, will be able to accept and send IP packets.

In any case, you will be able to manually assign static IP addresses from inside the VMs.

- b. Specify a name, and then click **Next**.



Create virtual network ×

- Network configuration IP address management ⓘ
- Subnet configuration
- Summary

Next

3. If you enabled IP address management, you will move on to the **Subnet configuration** step, where you can configure the network settings for IP address management:
 - a. Specify the network's IPv4 address range and, optionally, specify a gateway. If you leave the **Gateway** field blank, the gateway will be omitted from network settings.
 - b. Enable or disable the built-in DHCP server:
 - With the DHCP server enabled, VM network interfaces will automatically be assigned IP addresses: either from allocation pools or, if there are no pools, from the network's entire IP range.
 - With the DHCP server disabled, VM network interfaces will still get IP addresses, but you will have to manually assign them inside VMs.

The virtual DHCP service will work only within the current network and will not be exposed to other networks.

- c. Specify one or more allocation pools (ranges of IP addresses that will be automatically assigned to VMs).
- d. Specify DNS servers that will be used by virtual machines. These servers can be delivered to VMs via the built-in DHCP server or by using the cloud-init network configuration (if cloud-init is installed in the VM).
- e. Click **Next**.

Create virtual network ✕

- Network configuration
- Subnet configuration
- Summary

Configure network settings for IP address management.

Subnet CIDR
192.168.30.0/24

Gateway (optional)
192.168.30.1

Built-in DHCP server ⓘ

Allocation pools + Add

192.168.30.2

—

192.168.30.128

127 addresses available

🗑

DNS servers + Add

8.8.8.8

🗑

Back
Next

4. On the **Summary** step, review the configuration, and then click **Create virtual network**.

Create virtual network ✕

- Network configuration
- Subnet configuration
- Summary

Review the virtual network details and go back to change them if necessary.

Type	Virtual (VXLAN-based)
Name	net1
CIDR	192.168.30.0/24
Gateway	192.168.30.1
Built-in DHCP server	Yes
Allocation pools	192.168.30.2 — 192.168.30.128 127 addresses available
DNS servers	8.8.8.8

Back
Create virtual network

To view and edit parameters of a virtual network, click it on the **Networks** screen. On the network right pane, you can edit the network name and IP management settings. To do this, click the pencil icon next to the required section, and then make changes.

To delete a compute network, click the ellipsis icon next to it, and then click **Delete**. To remove multiple compute networks at once, select them, and then click **Delete**. Before deleting a compute network, make sure no VMs are connected to it.

4.7 Managing virtual routers

Virtual routers provide L3 services such as routing and Source Network Address Translation (SNAT) between virtual and physical networks, or different virtual networks:

- A virtual router between virtual and physical networks provides access to public networks, such as the Internet, for VMs connected to this virtual network.
- A virtual router between different virtual networks provides network communication for VMs connected to these virtual networks.

A virtual router has two types of ports:

- An external gateway that is connected to a physical network.
- An internal port that is connected to a virtual network.

With virtual routers, you can do the following:

- Create virtual routers
- Change external or internal router interfaces
- Create, edit, and delete static routes
- Change a router name
- Delete a router

Note

A router can only connect networks that have IP management enabled.

To create a virtual router, do the following:

1. On the **Networks** screen, make sure the compute networks that are to be connected to a router have a gateway specified.
2. Navigate to the **Routers** screen, and then click **Add router**.
3. In the **Add router** window:
 - a. Specify a router name.
 - b. From the **Network** drop-down menu, select a physical network through which external access will be provided via an external gateway. The new external gateway will pick an unused IP address from the selected physical network.
 - c. In the **Add internal interfaces** section, select one or more virtual networks to connect to a router via internal interfaces. The new internal interfaces will attempt to use the gateway IP address of the selected virtual networks by default.
 - d. [Optional] Select or deselect the **SNAT** check box to enable or disable SNAT on the external gateway of the router. With SNAT enabled, the router replaces VM private IP addresses with the public IP address of its external gateway.

Add virtual router

×

Name

router1

Specify a network through which public networks will be accessed.

Network

public: 10.94.0.0/16

SNAT ⓘ

Add internal interfaces

+ Add

private: 192.168.128.0/24

Cancel

Create

4. Click **Create**.

4.7.1 Managing router interfaces

To add an external router interface, do the following:

1. If you already have an external gateway, remove the existing one first.
2. On the **Routers** screen, click the router name to open the list of its interfaces.
3. Click **Add** on the toolbar, or click **Add interface** if there are no interfaces to show.
4. In the **Add interface** window, do the following:

- a. Select **External gateway**.
- b. From the **Network** drop-down menu, select a physical network to connect to the router. The new interface will pick an unused IP address from the selected physical network. You can also provide a specific IP address from the selected physical network to assign to the interface in the **IP address** field.
- c. [Optional] Select or deselect the **SNAT** check box to enable or disable SNAT on the external gateway of the router. With SNAT enabled, the router replaces VM private IP addresses with the public IP address of its external gateway.

Add interface ✕

External gateway Internal Interface

Specify new interface parameters

Network

public: 10.94.0.0/16 ▼

IP address (optional)

By adding a router interface you connect the selected network to the router. The new interface will pick an unused IP address from the selected public network. You can also provide a specific IP address from the selected public network to assign to the interface.

SNAT ⓘ

CancelAdd

5. Click **Add**.

To edit the external gateway parameters, click the ellipsis icon next to it, and then **Edit**. In the **Edit interface** window, you can change the external gateway IP address and enable or disable SNAT on it. To save your changes, click **Save**.

To add an internal router interface, do the following:

1. On the **Routers** screen, click the router name to open the list of its interfaces.
2. Click **Add**.
3. In the **Add interface** window, select a network to connect to the router from the **Network** drop-down menu. The new interface will attempt to use the gateway IP address of the selected virtual network by default. If it is in use, specify an unused IP address from the selected virtual network to assign to the interface in the **IP address** field.

Add interface ✕

Specify new interface parameters

Network
private2: 192.168.30.0/24 ▾

IP address (optional)

By adding a router interface you connect the selected network to the router. The new interface will attempt to use the gateway IP address of the selected private network by default. If it is in use, specify an unused IP address from the selected private network to assign to the interface.

Cancel Add

4. Click **Add**.

To remove a router interface, click the ellipsis icon next to it, and then click **Delete**.

4.7.2 Managing static routes

You can also configure static routes of a router by manually adding entries into its routing table. This can be useful, for example, if you do not need a mutual connection between two virtual networks and want only one virtual network to be accessible from the other.

Consider the following example:

- The virtual machine **VM1** is connected to the virtual network **private1** (192.168.128.0/24) via the network interface with IP address 192.168.128.10.
- The virtual machine **VM2** is connected to the virtual network **private2** (192.168.30.0/24) via the network interface with IP address 192.168.30.10.
- The router **router1** connects the network **private1** to the physical network via the external gateway with the IP address 10.94.129.73.
- The router **router2** connects the network **private2** to the physical network via the external gateway with the IP address 10.94.129.74.

To be able to access **VM2** from **VM1**, you need to add a static route for **router1**, specifying the CIDR of **private2**, that is 192.168.30.0/24, as the destination subnet and the external gateway IP address of **router2**, that is 10.94.129.74, as the next hop IP address. In this case, when an IP packet for 192.168.30.10 reaches **router1**, it will be forwarded to **router2** and then to **VM2**.

To create a static route for a router, do the following:

1. On the **Routers** screen, click the router name. Open the **Static routes** tab, and then click **Add** on the right pane. If there are no routes to show, click **Add static route**.
2. In the **Add static route** window, specify the destination subnet range and mask in CIDR notation and the next hop's IP address. The next hop's IP address must belong to one of the networks that the router is connected to.

Add static route ✕

Specify static route parameters

Destination subnet and mask

192.168.30.0/24

Next hop

10.94.129.74

The next hop's IP address must belong to one of the networks that the router is connected to.

Cancel
Add

3. Click **Add**.

To edit a static route, click the ellipsis icon next to it, and then click **Edit**. In the **Edit static route** window, change the desired parameters, and then click **Save**.

To remove a static route, click the ellipsis icon next to it, and then click **Delete**. To remove multiple routes at once, select them, and then click **Delete**.

4.8 Managing floating IP addresses

A virtual machine connected to a virtual network can be accessed from public networks, such as the Internet, by means of a floating IP address. Such an address is picked from a physical network and mapped to the VM's private IP address. The floating and private IP addresses are used at the same time on the VM's network interface. The private IP address is used to communicate with other VMs on the virtual network. The floating IP address is used to access the VM from public networks. The VM guest operating system is unaware of the assigned floating IP address.

Note the following prerequisites:

- A VM must have a fixed private IP address.
- A virtual router must connect the physical network, from which a floating IP will be picked, with the VM's virtual network.

To create a floating IP address and assign it to a VM, do the following:

1. On the **Floating IPs** screen, click **Add floating IP**.
2. In the **Add floating IP address**, select a physical network, from which a floating IP will be picked, and a VM network interface with a fixed private IP address.

The screenshot shows a dialog box titled "Add floating IP address" with a close button (X) in the top right corner. The dialog contains two dropdown menus. The first dropdown is labeled "Network" and displays "public: 10.94.0.0/16". The second dropdown is labeled "Virtual machine" and displays "myvm — private: 192.168.128.6". At the bottom of the dialog, there are two buttons: "Cancel" and "Add".

3. Click **Add**.

A floating IP address can be re-assigned to another virtual machine. Do the following:

1. Click the ellipsis icon next to the floating IP address, and then click **Unassign**.
2. Once the VM name disappears in the **Assigned to** column, click the ellipsis icon again, and then select **Assign**.
3. In the **Assign floating IP address** window, select a VM network interface with a fixed private IP address.
4. Click **Assign**.

To remove a floating IP address, unassign it from a VM, as described above, then click the ellipsis icon again and select **Delete**.

4.9 Managing load balancers

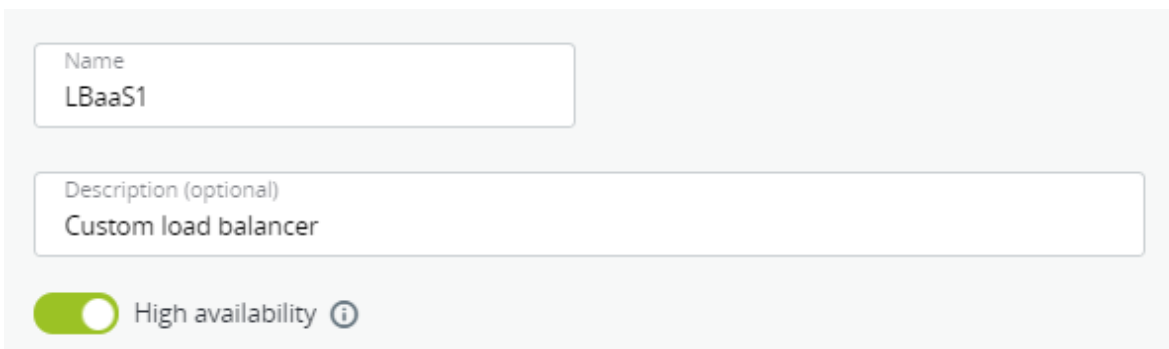
Virtuozzo Hybrid Infrastructure offers load balancing as a service for the compute infrastructure. Load balancing ensures fault tolerance and improves performance of web applications by distributing incoming network traffic across virtual machines from a balancing pool. A load balancer receives and then routes incoming requests to a suitable VM based on a configured balancing algorithm and VM health.

Note the following requirements:

- A load balancer can only operate in networks with enabled IP management.
- All VMs in balancing pools must have fixed IP addresses.

You can create a load balancer with balancing pools as follows:

1. On the **Load balancers** screen, click **Create load balancer**.
2. In the **Create load balancer** window, do the following:
 - a. Specify a name and optionally description.
 - b. Enable or disable high availability:
 - With high availability enabled, two load balancer instances will be created. They will work in the Active/Standby mode according to the Virtual Router Redundancy Protocol (VRRP).
 - With high availability disabled, a single load balancer instance will be created.



The screenshot shows a form for creating a load balancer. It has two text input fields: the first is labeled 'Name' and contains the text 'LBaaS1'; the second is labeled 'Description (optional)' and contains the text 'Custom load balancer'. Below these fields is a toggle switch for 'High availability', which is currently turned on (indicated by a green circle). To the right of the toggle is an information icon (a circle with an 'i').

3. In the **Network settings** section, select the network that the load balancer will operate in and, optionally, specify an IP address that will be allocated to the load balancer.
4. If a chosen virtual network is connected to a physical network via a router, you can assign a floating IP address to the load balancer. To do it, select **Use a floating IP address**, and from the

drop-down menu that appears, choose either to use an available floating IP address or to create a new one.

The screenshot shows a 'Network settings' panel. At the top, there is an information icon and the text 'Cannot be changed after the load balancer is added.' Below this, there are two input fields: 'Network' with a dropdown menu showing 'private1: 192.168.30.0/24' and 'IP address (optional)'. A checkbox labeled 'Use a floating IP address' is checked. Below the checkbox is another dropdown menu for 'Floating IP address' showing '192.168.30.0/24 → 10.94.129.66 (pu...'

5. In the **Balancing pools** section, create a balancing pool to forward traffic from the load balancer to virtual machines by clicking **Add**.

In the **Create balancing pool** window that opens, do the following:

- a. In the **Forwarding rule** section, select a forwarding rule from the load balancer to the backend protocol, and then specify the ports for incoming and destination connections.

Note the following:

- With the **HTTPS -> HTTPS** rule, all virtual machines need to have the same SSL certificate (or a certificate chain).
- With the **HTTPS -> HTTP** rule, you need to upload an SSL certificate (or a certificate chain) in the PEM format and a private key in the PEM format.

Important

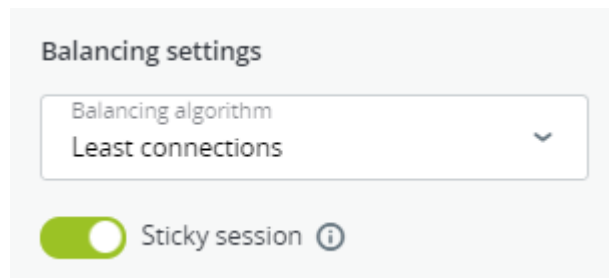
The forwarding rule cannot be changed after the load balancer is created.

The screenshot shows a 'Forwarding rule' panel. At the top, there is an information icon and the text 'Cannot be changed after the load balancer is added.' Below this, there are three input fields: 'From load balancer to backend protocol' with a dropdown menu showing 'HTTP → HTTP', 'LB port' with the value '80', and 'Backend port' with the value '80'.

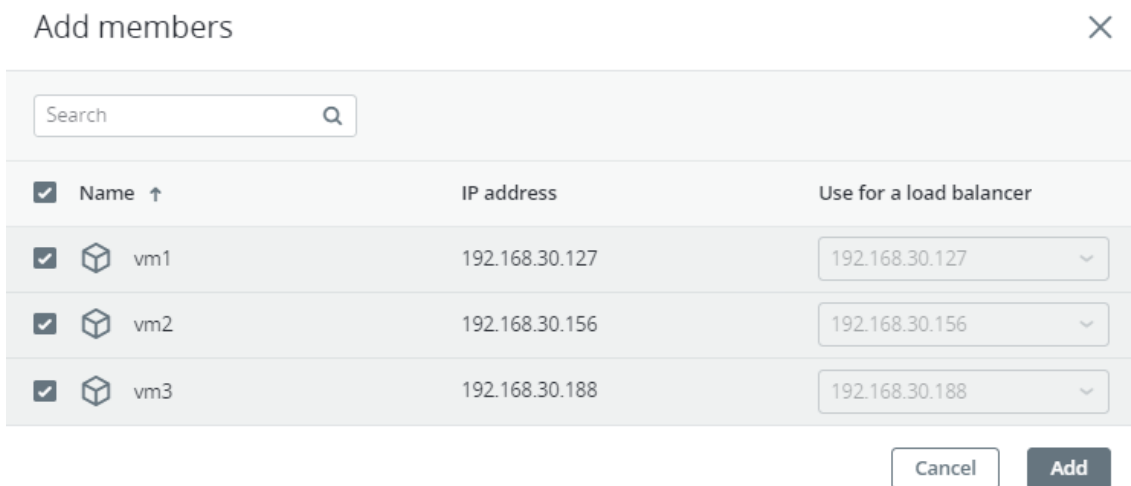
- b. In the **Balancing settings** section, select the balancing algorithm:
 - **Least connections.** Requests will be forwarded to the VM with the least number of active connections.
 - **Round robin.** All VMs will receive requests in the round-robin manner.
 - **Source IP.** Requests from a unique source IP address will be directed to the same VM.Enable/disable the **Sticky session** option to enable/disable session persistence. The load balancer will generate a cookie that will be inserted into each response. The cookie will be used to send future requests to the same VM.

Note

This option is not available in the SSL passthrough mode.



- c. In the **Members** section, add members, that is, virtual machines, to the balancing pool by clicking **Add**. Each VM can be included to multiple balancing pools. In the **Add members** window that opens, select the desired VMs, and then click **Add**.



- d. In the **Health monitor** section, select the protocol that will be used for monitoring members availability:
- **HTTP/HTTPS**. The HTTP/HTTPS method GET will be used to check for the response status code 200. Additionally, specify the URL path to the health monitor.
 - **TCP/UDP**. The health monitor will check the TCP/UDP connection on the backend port.
 - **PING**. The health monitor will check members' IP addresses.

Important

The protocol cannot be changed after the load balancer is created.

Health monitor

The health monitor defines how the load balancer monitors the availability of members in the pool.

i The protocol cannot be changed after the load balancer is created.

Protocol HTTP	URL path /
------------------	---------------

The HTTP method GET will be used to check for the response status code 200.

[Edit parameters](#)

By default, the health monitor removes a member from a balancing pool if it fails three consecutive health checks of five-second intervals. When a member returns to operation and responds successfully to three consecutive health checks, it is added to the pool again. You can manually set the health monitor parameters, such as the interval after which VM health is checked, the time after which the monitor times out, healthy and unhealthy thresholds. To change the default parameters, click **Edit parameters**, enter the desired values, and then click **Save**.

Edit health monitor parameters



Interval	<input type="text" value="5"/>
Interval after which member health is checked.	from 5 to 300 seconds
Timeout	<input type="text" value="5"/>
The time a monitor has to poll a member. Must be less than the interval.	from 5 to 60 seconds
Healthy threshold	<input type="text" value="3"/>
The number of consecutive successful checks after which a member is marked as healthy.	from 1 to 10 attempts
Unhealthy threshold	<input type="text" value="3"/>
The number of consecutive unsuccessful checks after which a member is marked as unhealthy.	from 1 to 10 attempts

Cancel




Save

- e. Click **Create**.
6. Add more balancing pools, as described above, if required.

7. Click **Create**.

Create one or more balancing pools to forward traffic from the load balancer to members.

Balancing pools (1) + Add

 HTTP on port 80 → HTTP on port 80 3 members  | 

Cancel Create

Once the load balancer is created, you can monitor its performance and health on the **Overview** tab of its right pane.

The following charts are available:

Members state

The total number of members in the balancing pools grouped by status: "Healthy," "Unhealthy," "Error," and "Disabled".

CPU/RAM

CPU and RAM usage by the load balancer.

Network

Incoming and outgoing network traffic.

Active connections

The number of active connections.

Error requests

The number of error requests.










You can see the load balancer parameters on its **Properties** tab.

To edit the name or description of a load balancer, click the ellipsis icon next to it, and then click **Edit**.

To disable/enable or remove a load balancer, click the ellipsis icon next to it, and then click the desired action. To remove multiple load balancers at once, select them, and then click **Delete**.

4.9.1 Managing balancing pools

To see a list of balancing pools in a load balancer, click its name.

<input type="checkbox"/>	Balancing pool	Status	Members state	Members total	
<input type="checkbox"/>	 HTTP on port 80 → HTTP on port 80	 Active		3	
<input type="checkbox"/>	 HTTPS on port 443 → HTTPS on port 443	 Active		3	

To add another balancing pool to a load balancer, click **Create balancing pool**, and then fill in the fields, as described in "Managing load balancers" (p. 59). The newly added pool will appear in the list of balancing pools.

You can open the pool right pane to monitor its performance and health on the **Overview** tab, see its parameters on the **Properties** tab, and manage its members on the **Members** tab.

To edit the balancing settings such as the balancing algorithm and session persistence, click the ellipsis icon next to a pool, and then click **Edit**. To edit the health monitor parameters, click the ellipsis icon next to a pool, and then click **Edit health monitor**.

To add more members to a balancing pool, click the ellipsis icon next to it, and then click **+ Add members**.

To remove a balancing pool, click the ellipsis icon next to it, and then click **Delete**. To remove multiple balancing pools at once, select them, and then click **Delete**.

4.10 Managing SSH keys

Use of SSH keys allows you to secure SSH access to virtual machines. You can generate a key pair on a client from which you will connect to VMs via SSH. The private key will be stored on the client and you will be able to copy it to other nodes. The public key will need to be uploaded to Virtuozzo Hybrid Infrastructure and specified during VM creation. It will be injected into the VM by cloud-init and used for OpenSSH authentication. Keys injection is supported for both Linux and Windows virtual machines.

Note

You can specify an SSH key only if you deploy a VM from a template or boot volume (not an ISO image).

Before using the SSH keys feature, make sure the following requirements are met:

- The `cloud-init` utility is installed in a VM template or boot volume.
- OpenSSH Server is installed in a Windows template or boot volume.

For the instructions on preparing templates or boot volumes, refer to "Preparing templates" (p. 40).

To add a public key, do the following:

1. Generate an SSH key pair on a client by using the `ssh-keygen` utility:

```
# ssh-keygen -t rsa
```

2. On the **SSH keys** screen, click **Add key**.
3. In the **Add SSH key** window, specify a key name and copy the key value from the generated public key located in `/root/.ssh/id_rsa.pub`. Optionally, you can add a key description.

Add SSH key ✕

For the key to be successfully injected into the VM, the template must contain the cloud-init package.

Name
root_node001vstoragedomain

Description (optional)
My public key

Key value
n1h0culzIqbj2AHYqgiUWX7W3bE3nCCUxEX9DuHH2GJPy8Kz7HKa
RY0GULMIOJz7QRyzwBThgQ3TI1YX+OJ5i7kbUek9hygy+RR/kjnMMI
rg6gyP2b4BrDflpZUNx4Nx1L9IGCGUoTWPieIc0n2LQMh2fAfxBBh
mSDVUPBLpowxuAibOOkemW5IDjsKxuDuIqt35X27anWPcjFKTZN
47RnyCDT/X6tBYdxQJ6ARIQsp1JDWkjN7B65h9rwNZJ/PpyXI5wEVh
SLXrIMam93bh3YwMzQYhVILXGuvgbP+dF5Cq6Bg8FthXEfktpt121
5P/FD root@node001.vstoragedomain

Cancel Add

Note

If a key has been injected into one or more VMs, it will remain inside those VMs even if you delete it from the panel.
