Virtuozzo Hybrid Cloud

Self-Service Guide

May 16, 2024

Virtuozzo International GmbH

Vordergasse 59

8200 Schaffhausen

Switzerland

Tel: + 41 52 632 0411

Fax: + 41 52 672 2010

https://virtuozzo.com

Contents

CHAPTER 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.

CHAPTER 2

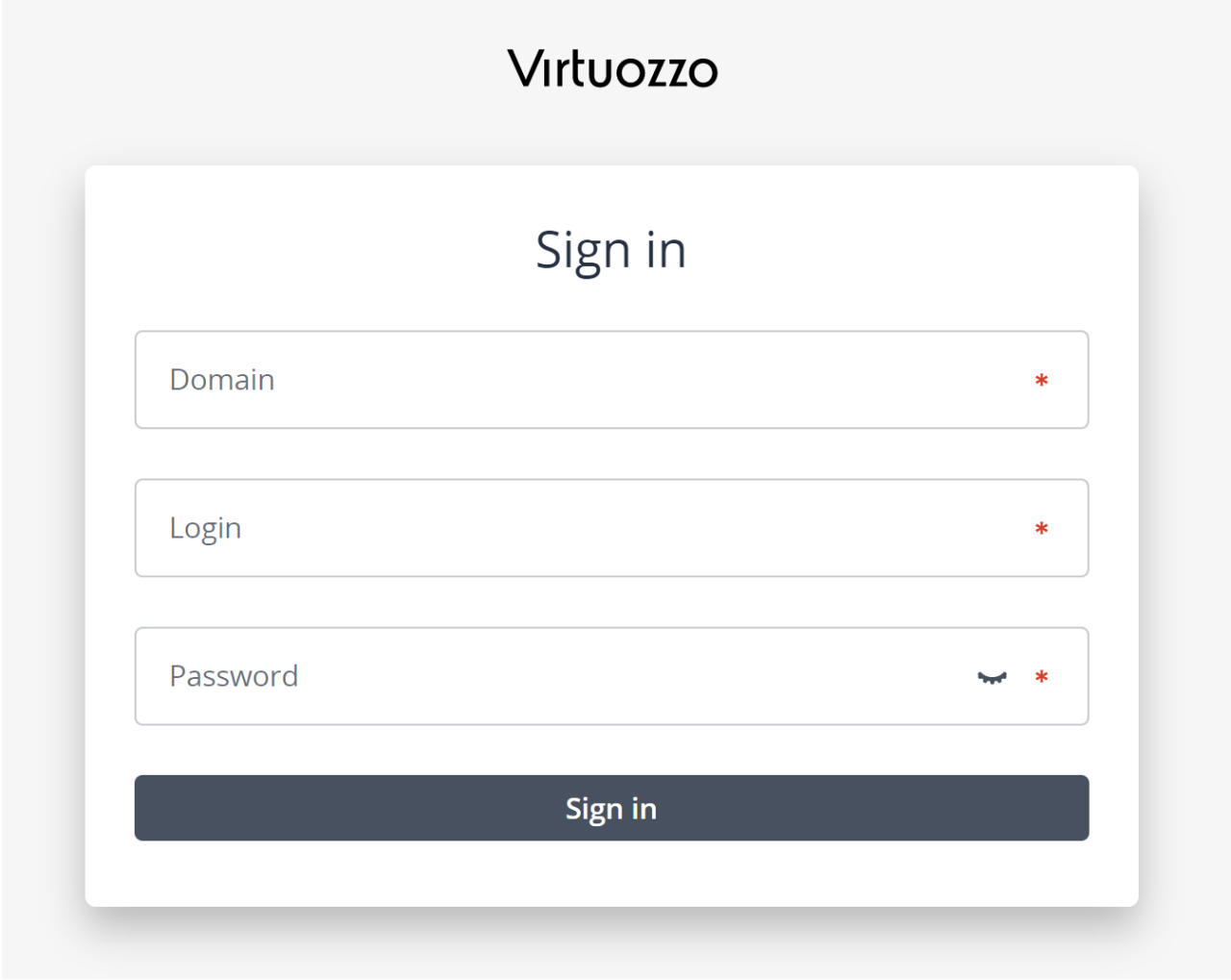
Logging in to Virtuozzo Hybrid Cloud Self-Service Panel

Prerequisites:

- You should be a Virtuozzo Hybrid Cloud customer to log in to the Virtuozzo Hybrid Cloud self-service panel.

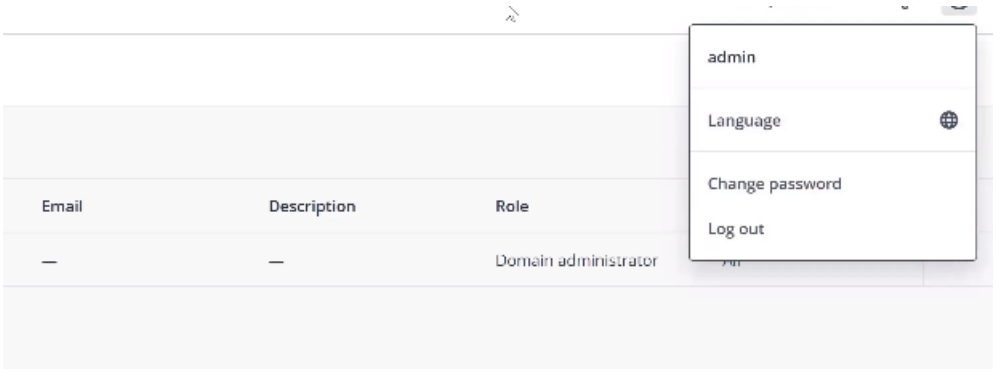
To log in:

1. Go to the required cloud location:
    - Europe (Frankfurt): https://eu1-cloud.virtuozzo.com
    - Europe (Amsterdam): https://eu3-cloud.virtuozzo.com
    - United States (Dallas): https://us1-cloud.virtuozzo.com
2. Enter your domain, login, and password.
3. Click **Sign in**.



Note: If you have lost your login credentials, please contact your sales representative or submit a support request.

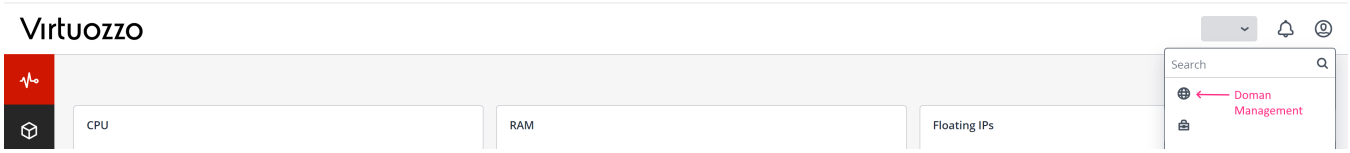
Once logged in to the Virtuozzo Hybrid Cloud self-service panel, you can select the preferred language and, if needed, change your password by clicking your account picture in the upper-right corner.


admin
Language
Change password
Log out
Email
Description
Role
—
—
Domain administrator

CHAPTER 3

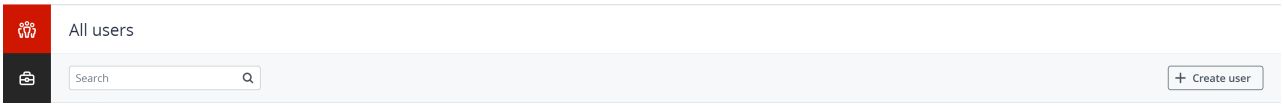
Managing Users and Projects

The domain administrator can manage other users within the domain. In the upper-right corner, in the drop-down list, select the domain name to access the user management panel.



3.1 Creating User

1. In the drop-down list upper right, select the domain name.
2. On the **All users** screen, click **+ Create user**.



3. In the **Create user** window, specify the user's login, password, and, if required, the user's email address and description.

Note: A user name must be unique within a domain.

---

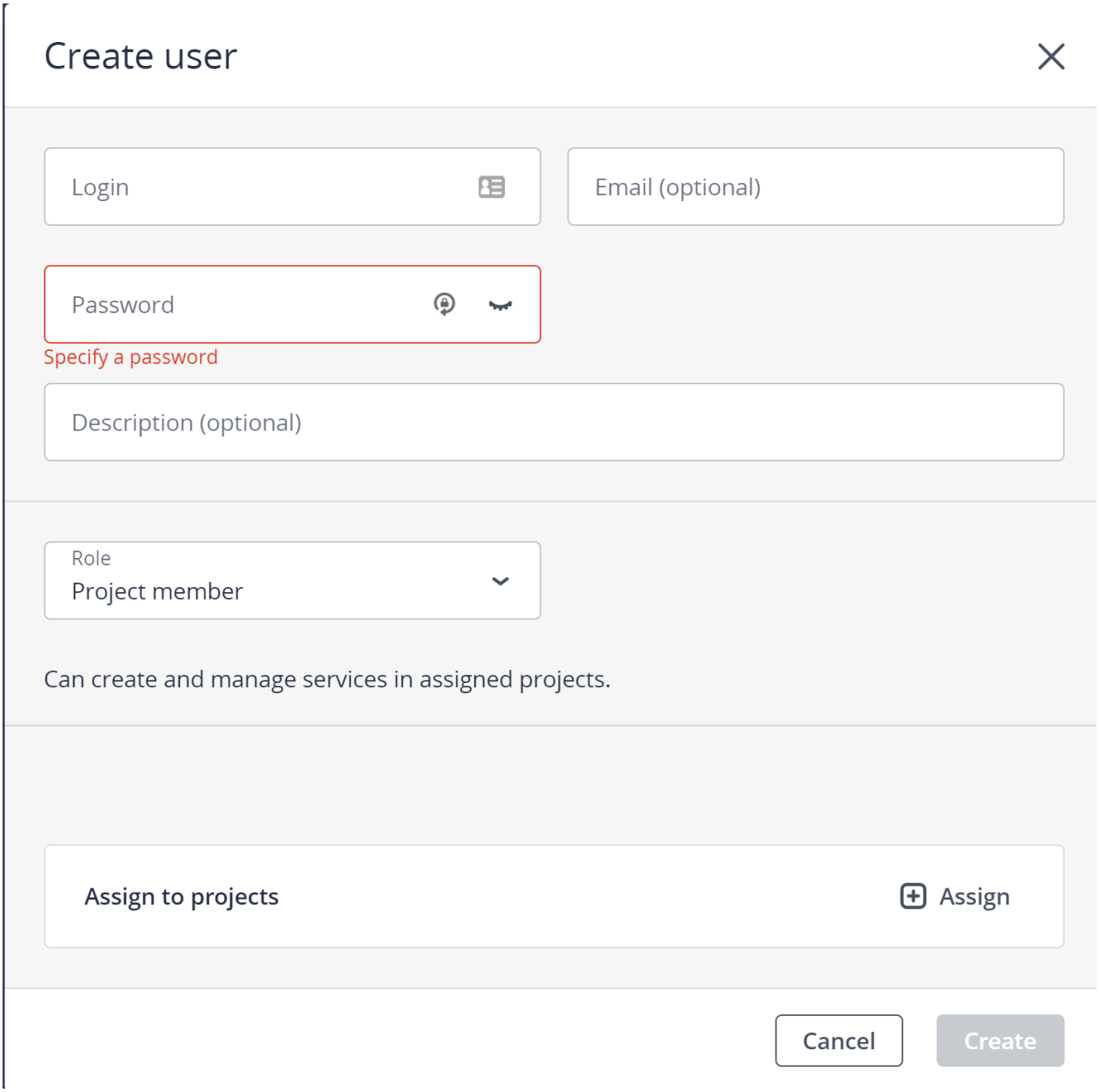
4. Select the desired role in the **Role** drop-down list:
   - **Domain administrator** can access all projects and manage users within their domain.
   - **Project member** can manage only projects they have access to. Click **+ Assign** to add the user

with the Project manager role to the desired project. In the **Manage projects** window, select the checkbox next to the project you want to assign to the user and click **Save**.

Note: If a user with the Project manager role is not assigned a project, such a user cannot log in to the self-service panel.

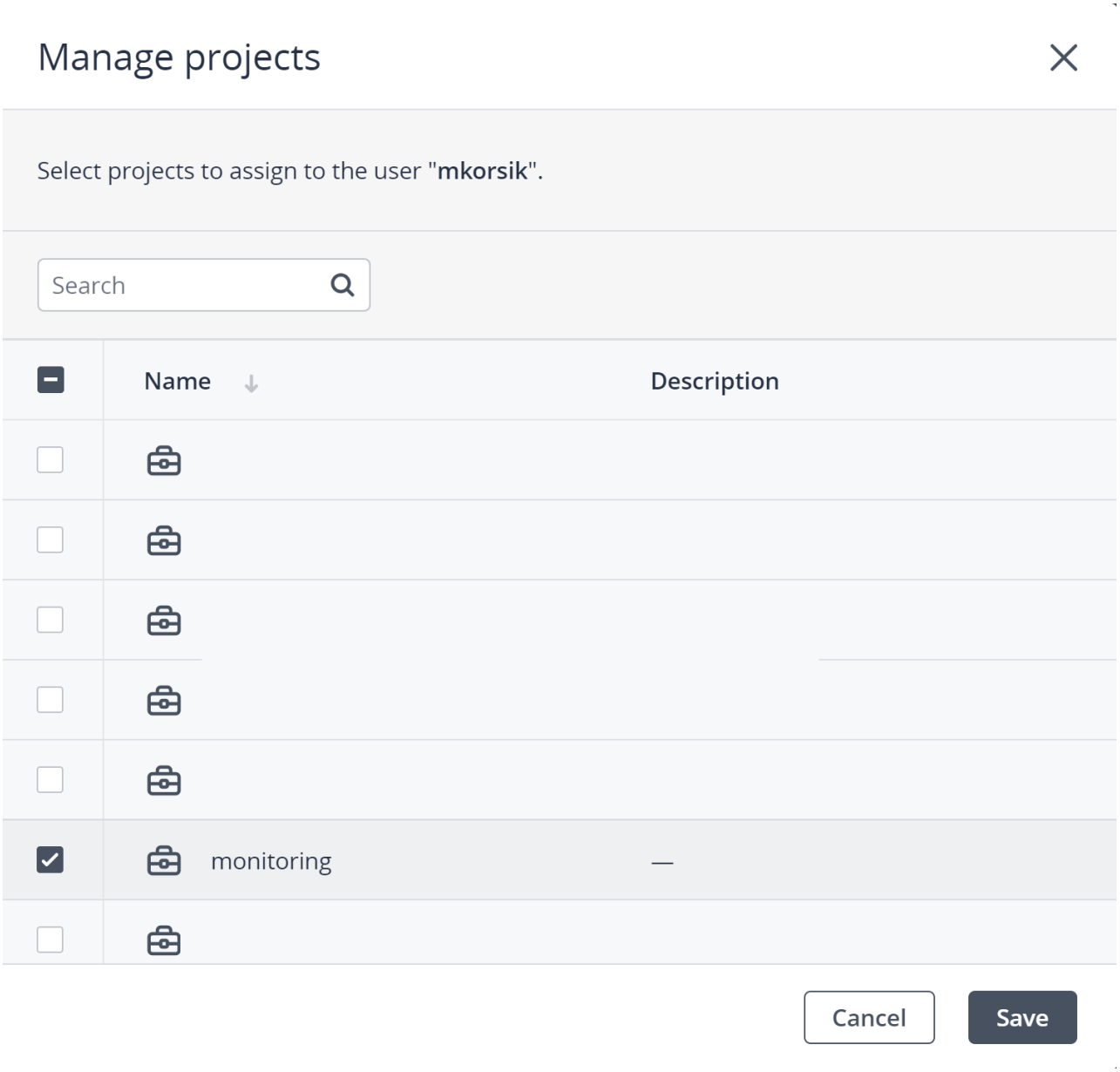
---

5. Click **Create**.



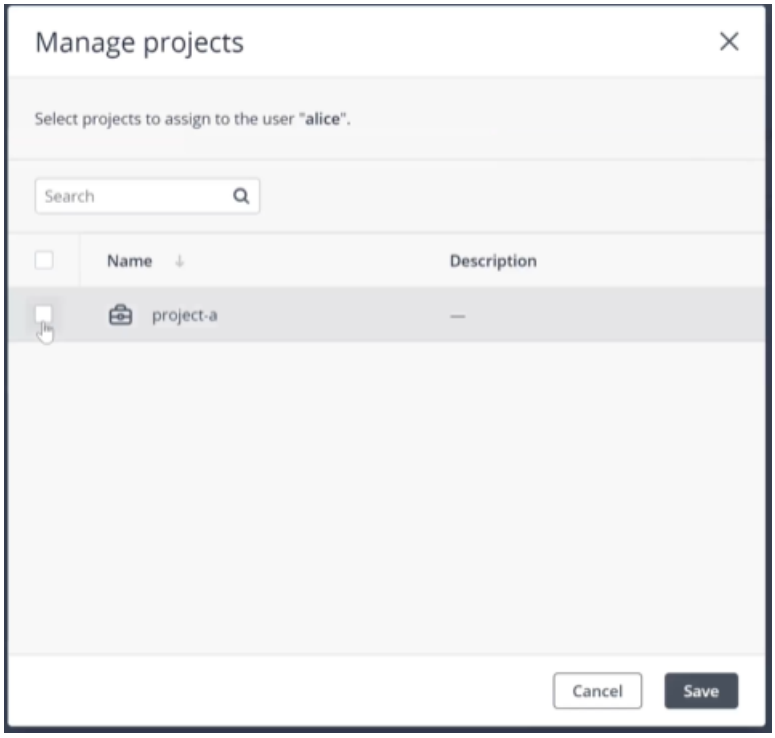
3.2 Assigning User to Project

1. In the drop-down list upper right, select the domain name.
2. On the **User details** screen, select the user you want to assign a project and click **Manage projects**.
3. In the **Manage projects** window, select the checkbox next to the project you want to assign to the user.
4. Click **Save**.



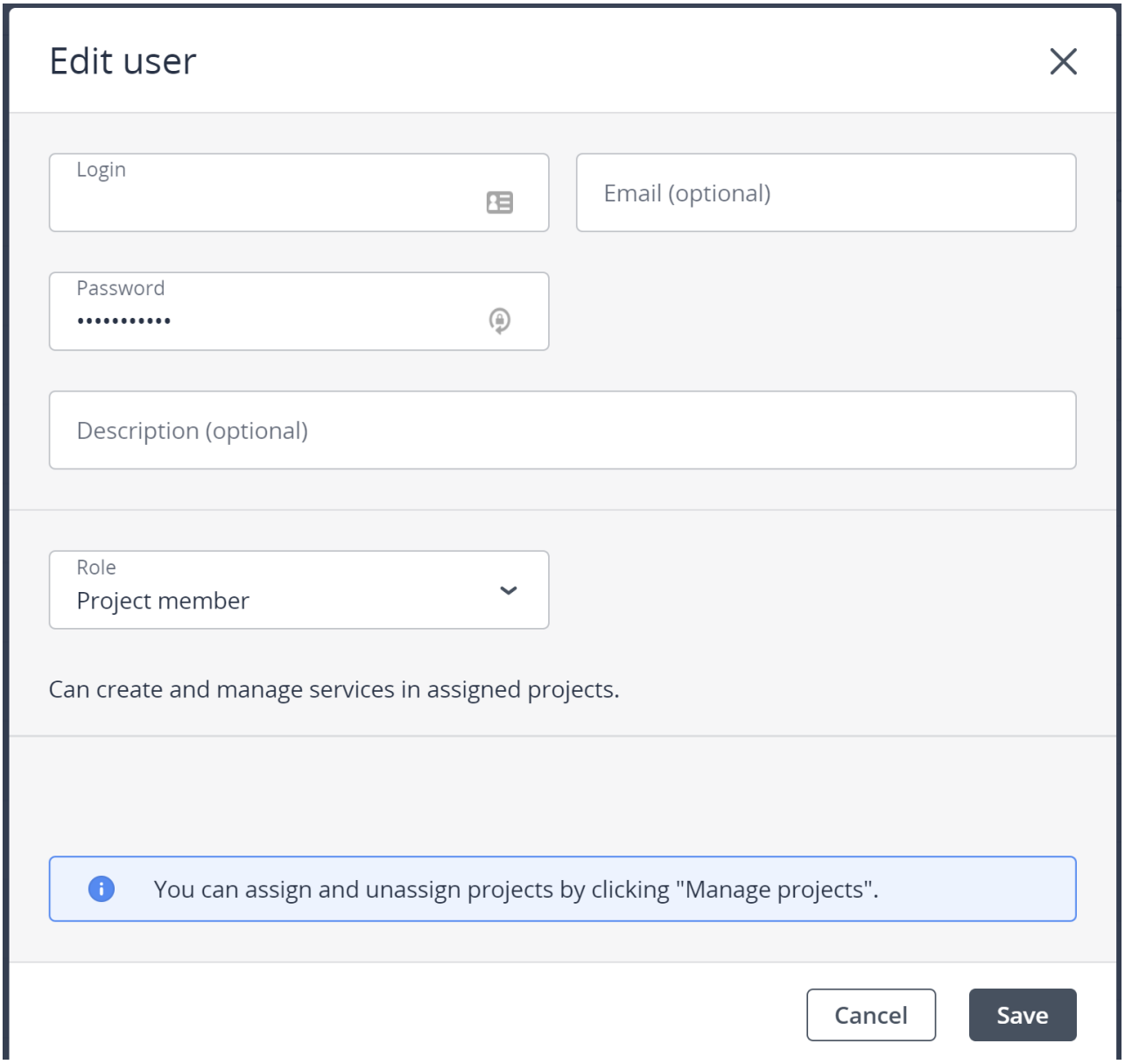
3.3 Unassigning User from Project

1. In the drop-down list upper right, select the domain name.
2. On the **User details** screen, select the user from whom you want to unassign a project and click **Manage projects**.
3. In the **Manage projects** window, clear the checkbox next to the project you want to unassign from the user.
4. Click **Save**.



3.4 Editing User

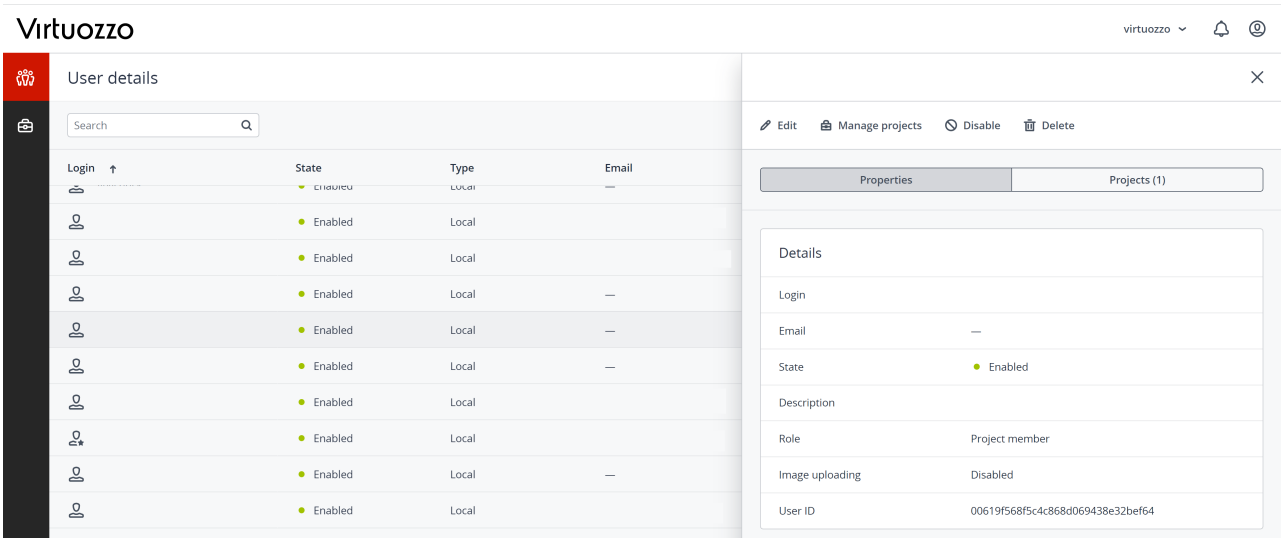
1. In the drop-down list upper right, select the domain name.
2. On the **User details** screen, select the user you want to edit and click **Edit**.
3. In the **Edit user** window, you can change the user's details.
4. Click **Save**.



3.5 Disabling User

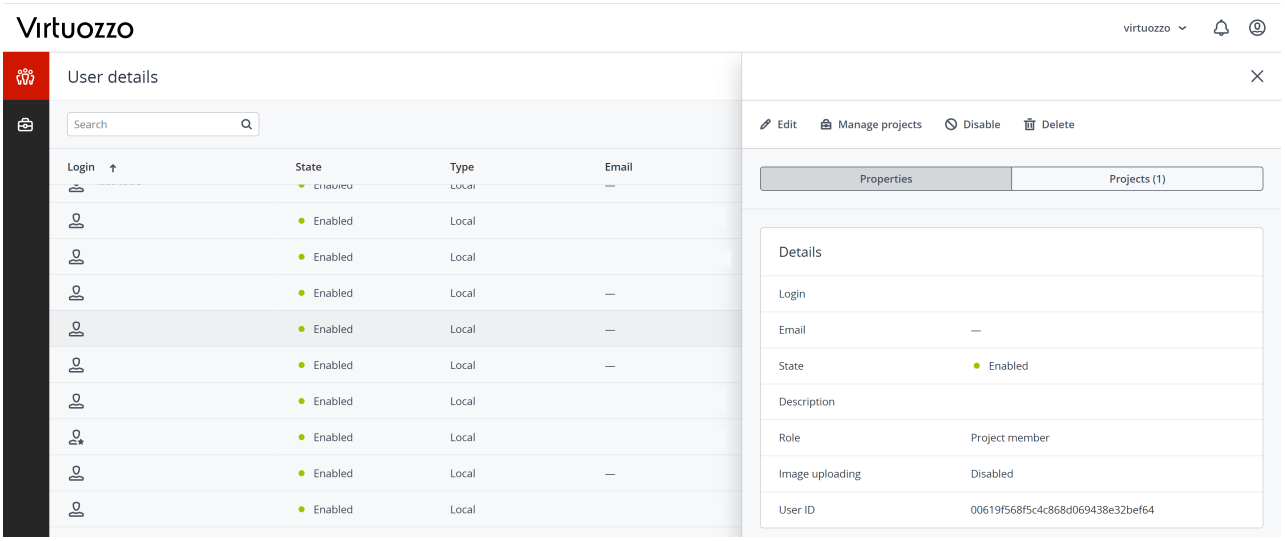
To disable a user from logging in to the self-service panel:

1. In the drop-down list upper right, select the domain name.
2. On the **User details** screen, select the user you want to disable and click **Disable**.



3.6 Deleting User

1. In the drop-down list upper right, select the domain name.
2. On the **User details** screen, select the user you want to delete and click **Delete**.



CHAPTER 4

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	64 virtual CPUs
Storage	15 volumes, 512 TiB each
Network	15 NICs

4.1 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 virtual machines:

Version	Edition	CPU hot plug support	RAM hot plug support
Windows Server 2022	Standard	Yes	Yes
Windows Server 2019	Standard	Yes	Yes

Linux virtual machines:

Distribution	Version	CPU hot plug support	RAM hot plug support
AlmaLinux	8.x, 9.x	Yes	Yes
CentOS	8.x, 7.x	Yes	Yes
	6.x	No	No
Debian	10.x, 9.x	Yes	Yes
RockyLinux	8.x, 9.x	Yes	Yes
Ubuntu	22.04.x, 20.04.x, 18.04.x	Yes	Yes
	16.04.x	No	No

4.2 Creating Virtual Machines

Limitations:

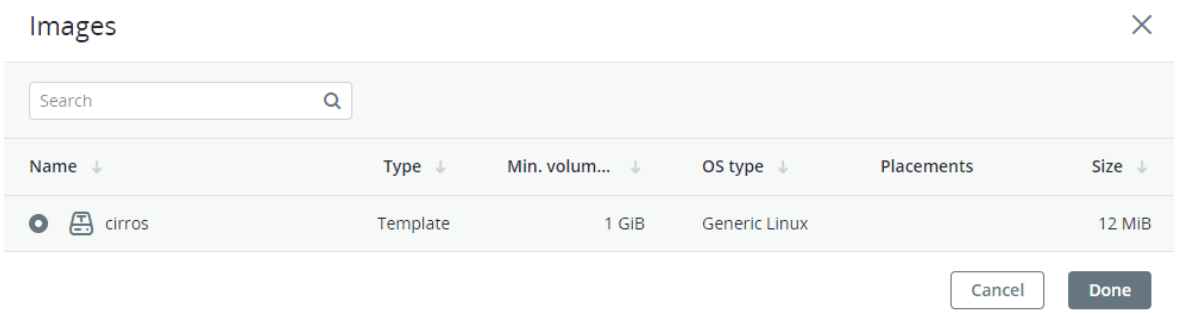
UEFI boot is not supported for CentOS 7.x virtual machines with less than 1 GiB of RAM.

Prerequisites:

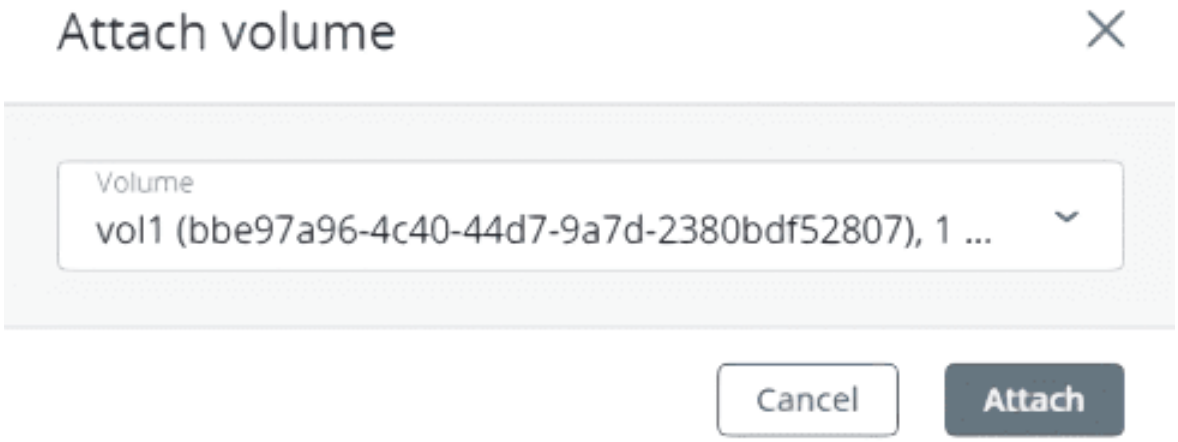
- You have a guest OS source prepared, as described in *Managing Images* on page 62.
- One or more compute networks are created by using the instructions in *Managing Virtual Private Networks* on page 75.
- [Optional] Custom security groups are configured, as instructed in *Managing Security Groups* on page 35.
- [Optional] An SSH key is added, as outlined in *Managing SSH Keys* on page 129. You can specify an SSH key only when creating VMs from a template or boot volume.

4.2.1 Ceating Virtual Machine

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:
     3.1. Select **Image** in the **Deploy from** section, and then click **Specify** in the **Image** section.
     3.2. In the **Images** window, select the ISO image or template, and then click **Done**.



   - If you have a compute boot volume:
     3.1. Select **Volume** in the **Deploy from** section, and then click **Specify** in the **Volumes** section.
     3.2. In the **Volumes** window, click **Attach**.
     3.3. 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.

Note: If you select an image or volume with an assigned placement, the created VM will also inherit this placement.

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:

   4.1. 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**.

   4.2. [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.

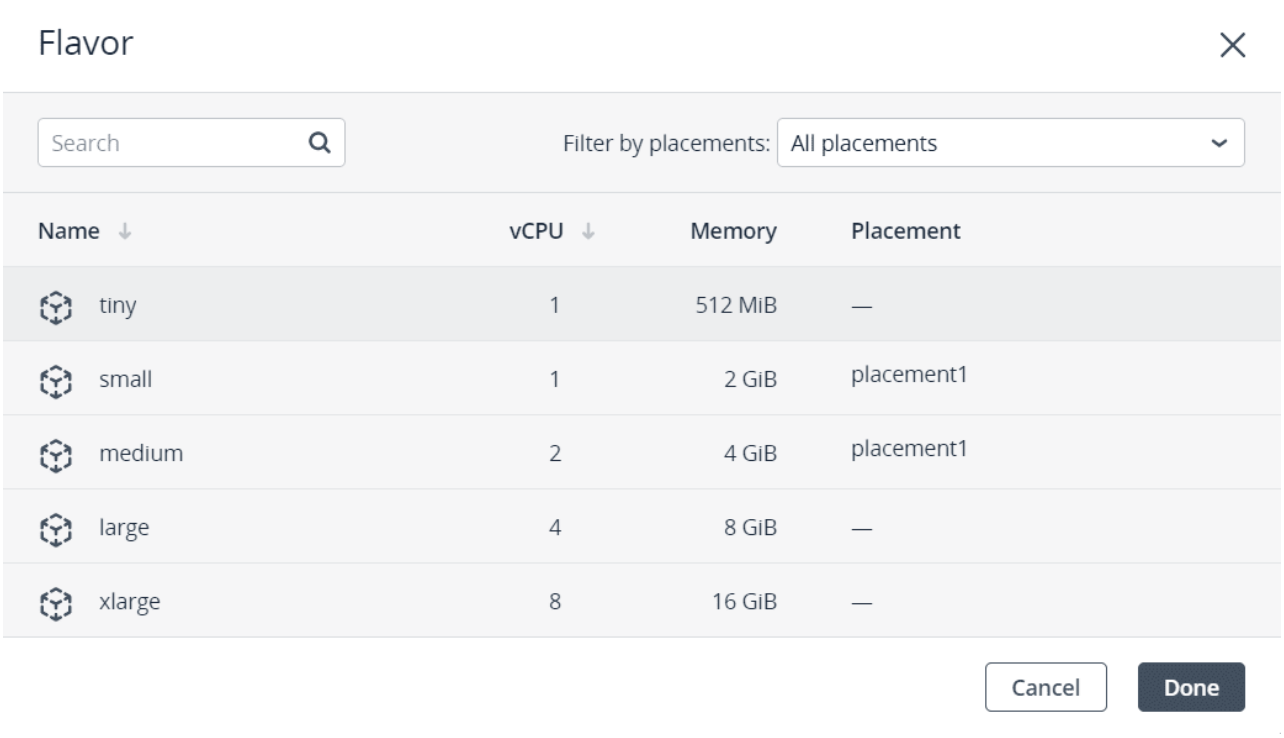
   4.3. Select volumes that will be removed during the VM deletion. To do this, click the pencil icon in the **Volumes** section, click the ellipsis icon next to the needed volume, and then **Edit**. Enable **Delete on termination** and click **Save**.

   4.4. When you finish configuring the VM disks, click **Done**.

5. 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.

Note: To select a flavor with an assigned placement, you can filter flavors by placement. The VM created from such a flavor will also inherit this placement.



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

   6.1. In the **Network interfaces** window, click **Add** to attach a network interface.

   6.2. In the **Add network interface** window, select a compute network to connect to, and then specify MAC address, IPv4 and/or IPv6 addresses, and security groups. By default, MAC and primary IP addresses are assigned automatically. To specify them manually, clear the **Assign automatically** check boxes, and enter the desired addresses. Optionally, assign additional IP addresses to the network interface in the **Secondary IP addresses** section. Note that a secondary IPv6 address is not available for an IPv6 subnet that works in the SLAAC or DHCPv6 stateless mode.

   Note: Secondary IP addresses, unlike the primary one, will not be automatically assigned to the

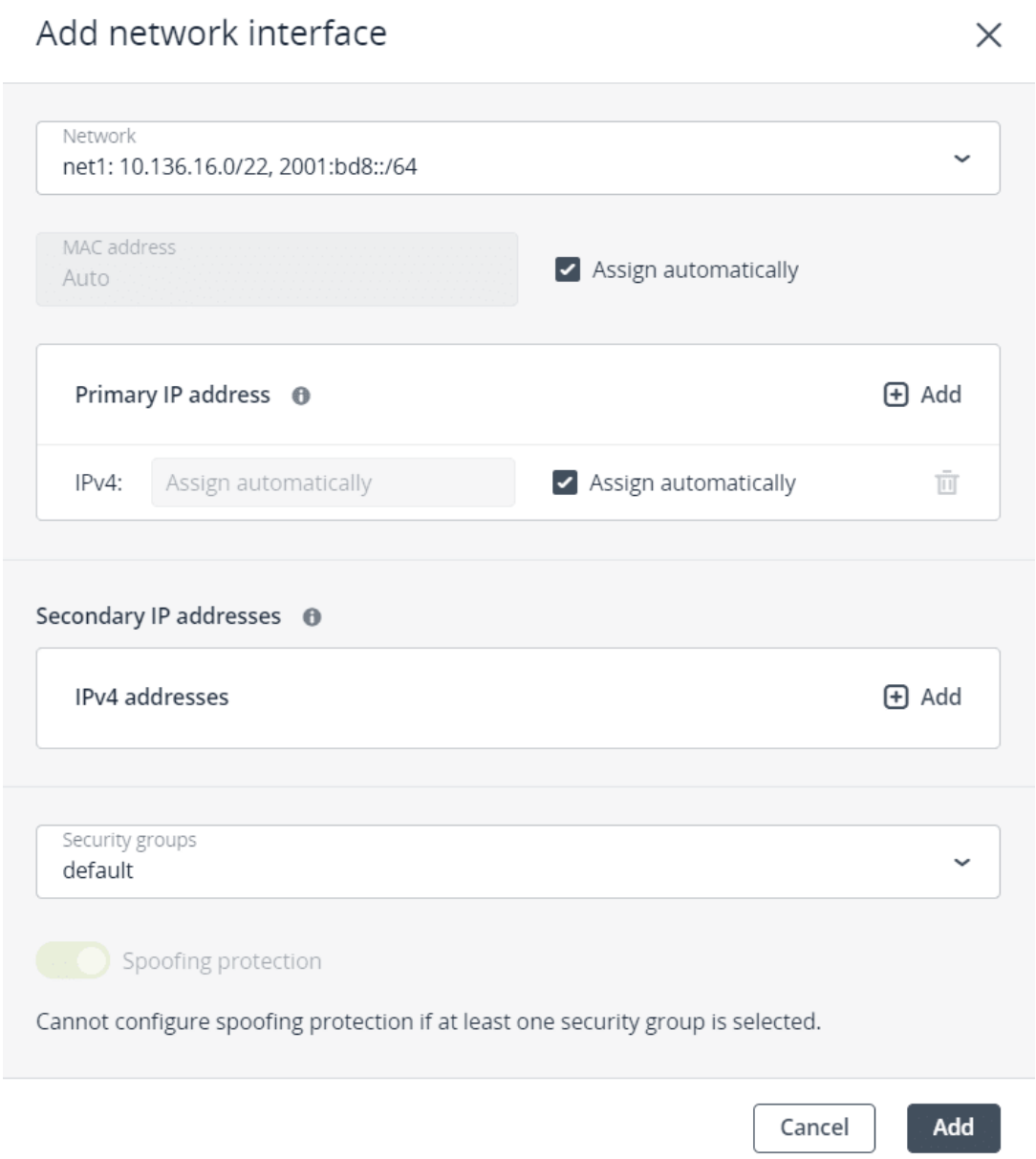
network interface inside the virtual machine guest OS. You should assign them manually.

- If you selected a virtual network with enabled IP address management: In this case, 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. If required, you can select another security group or multiple security groups.

  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.

- If you selected a virtual network with disabled IP address management: In this case, spoofing protection is disabled by default and cannot be enabled. Security groups cannot be configured for such a network.

- If you selected a shared physical network: In this case, spoofing protection cannot be configured by a self-service user. If you want to enable or disable spoofing protection, contact your system administrator.



After specifying the network interface parameters, click **Add**. The network interface will appear in the **Network interfaces** list.

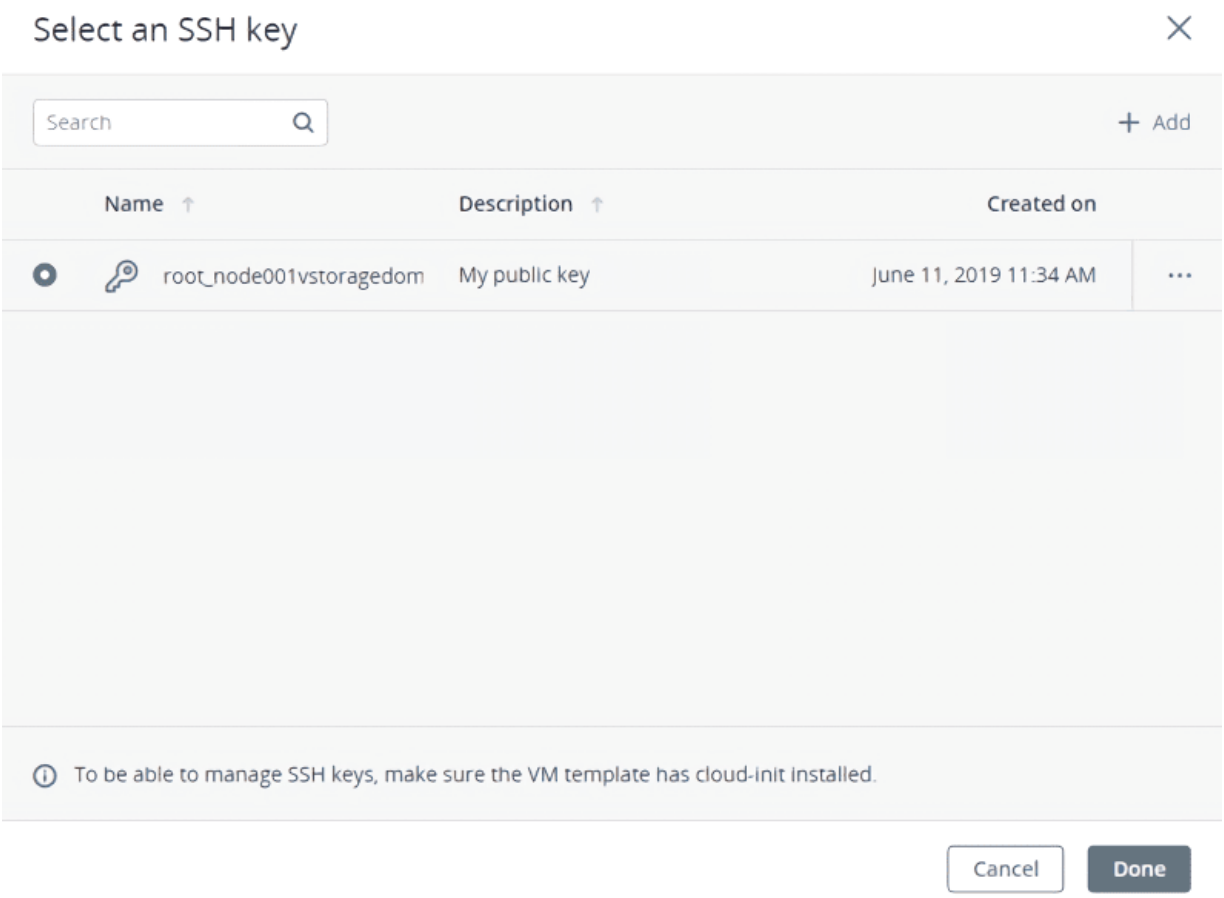
6.3. [Optional] If required, edit IP addresses and security groups of newly added network interfaces. To do this, click the ellipsis icon, click **Edit**, and then set the parameters.

6.4. When you finish configuring the VM network interfaces, click **Done**.

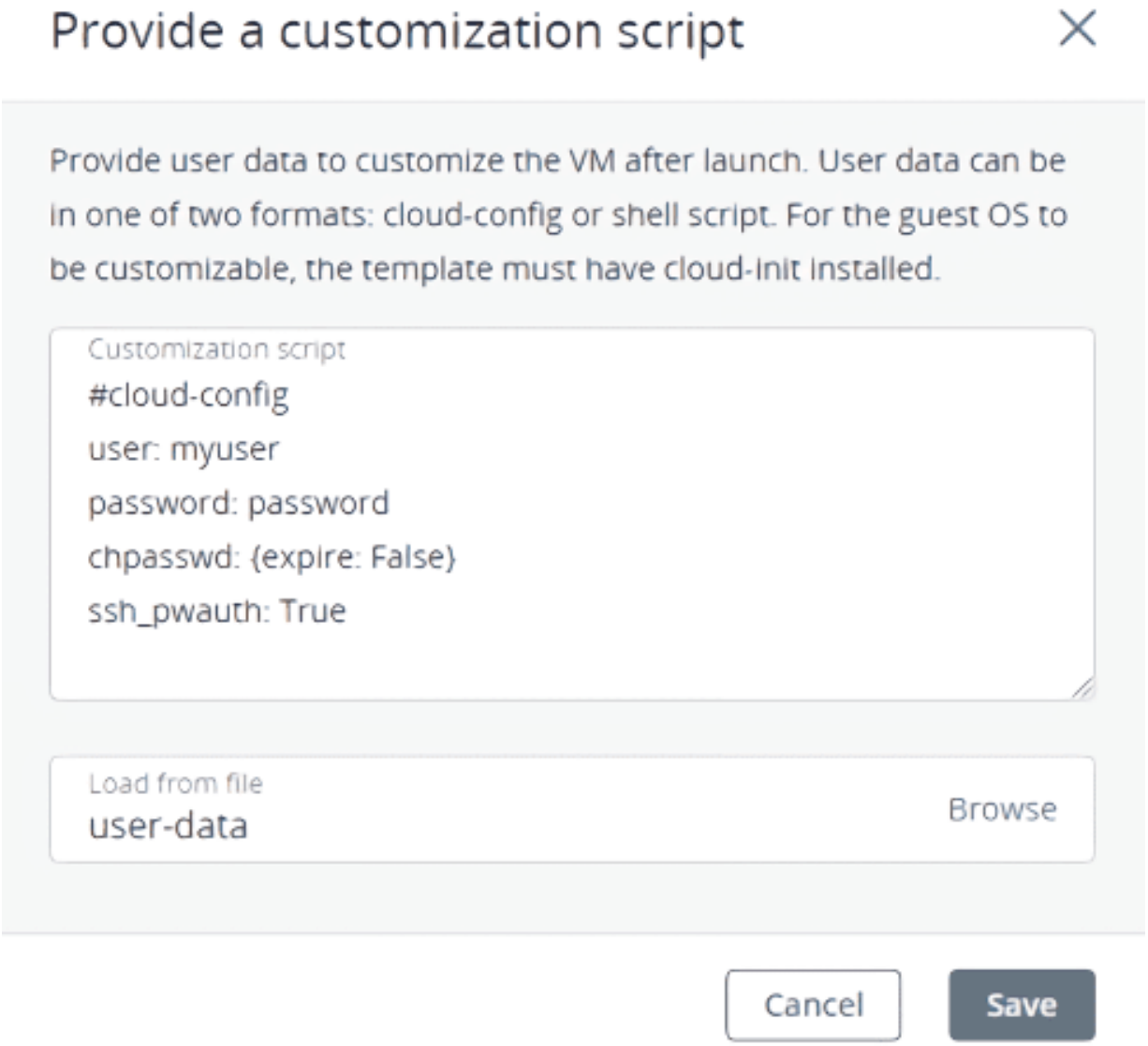
7. [Optional] If you have chosen to boot from a template or volume, which has cloud-init and OpenSSH installed:

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**.



- 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.



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>
```

8. [Optional] Enable CPU and RAM hot plug for the VM in **Advanced options**, to be able to change its flavor when the VM is running. You can also enable hot plug after the VM is created.

Note: If you do not see this option, CPU and RAM hot plug is disabled in your project. To enable it,

contact your system administrator.

9. [Optional] If you have chosen to boot from an ISO image, enable UEFI boot in **Advanced options**, to be able to boot the VM in the UEFI mode. This option cannot be configured after the VM is created.

   Note: You cannot configure UEFI boot if you have selected a template as the VM boot media. If your template has UEFI boot enabled, the option is automatically enabled for the VM, and vice versa.

10. 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. For VMs with UEFI boot enabled, open the VNC console, and then press any key to boot from the chosen ISO image. Virtual machines created from a template or a boot volume already have a preinstalled guest OS.

4.3 Connecting to Virtual Machines

Virtuozzo Cloud provides cloud images from OS vendors. By default, all cloud images have password login disabled in favor of key login.

4.3.1 Connecting to Virtual Machine via the VNC Console

Select a VM, and then click **Console** on its 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* on page 33).

4.3.2 Connecting to Virtual Machine 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.

4.3.3 Images and Cloud Usernames

Here is a list of images and cloud usernames:

Name	Cloud username
AlmaLinux-9	almalinux
AlmaLinux-8	almalinux
CentOS-9	centos
CentOS-8	centos
CentOS-7	centos
CentOS-6	centos
cirros	cirros
Debian-10	debian
Debian-9	debian
RockyLinux-9	rockylinux
RockyLinux-8	rockylinux
Ubuntu-22.04	ubuntu
Ubuntu-20.04	ubuntu
Ubuntu-18.04	ubuntu
Ubuntu-16.04	ubuntu
Windows Server 2022	Administrator
Windows Server 2019	Administrator

Note: Windows images do not have the SSH daemon installed. Set the administrator password on the first launch of the virtual machine.

4.4 Managing Virtual Machine Power State

Prerequisites:

- Virtual machines are created, as described in *Creating Virtual Machines* on page 12.

4.4.1 Managing the Power State of a Virtual Machine

Click the virtual machine or the ellipsis button next to it to see the full list of actions available for the current state.

- 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**.

4.5 Reconfiguring Virtual Machines

Once you create a virtual machine, you can manage its CPU and RAM resources, as well as network interfaces and volumes.

Prerequisites:

- Virtual machines are created, as described in *Creating Virtual Machines* on page 12.

4.5.1 Changing Virtual Machine Resources

You can change amount of CPU and RAM resources used by a virtual machine by applying another flavor to it. To be able to resize a running VM, you need to enable CPU and RAM hot plug for it first. You can change the hot plug settings for both new and existing VMs.

A running virtual machine has a resize limit, which defines the maximum number of vCPUs and the maximum amount of RAM you can allocate to the VM. The resize limit on vCPUs is static and equal to 64 for all VMs. The resize limit on RAM, on the contrary, is dynamic and depends on the amount of RAM a running VM is currently using. This limit is updated on a VM startup, and its values are listed in the table below.

Current RAM size, in GiB	**RAM size limit, in GiB**
1-4	16
5-8	32
9-16	64
17-32	128
33-64	256
65-128	512
129-256	1024

For example, you can resize a running VM with a flavor that has 16 GiB to a flavor with 256 GiB in two iterations:

1. Resize the VM to a flavor with 64 GiB.
2. Restart the VM to update the RAM size limit.
3. Resize the VM to a flavor with 256 GiB.

Limitations:

- You cannot change the flavor for shelved VMs. To resize such a VM, unshelve it first.
- You cannot decrease the number of CPUs and the amount of RAM for running VMs.
- [For all Linux guests] If a VM has no guest tools installed, new cores may be offline after CPU hot plugging. You can verify which CPU cores are online by using the command:

  ```
  # cat /sys/devices/system/cpu/online
  ```

  To activate offline CPU cores, run:

```
# echo 1 > /sys/devices/system/cpu/cpu<cpu_number>/online
```

Prerequisites:

- Before changing a flavor, ensure that the node hosting the VM has at least as much free CPU and RAM resources as the new VM size. For example, to resize a VM to the **large** flavor, the host must have at least 4 vCPUs and 8 GiB of RAM free.
- CPU and RAM hot plug is enabled by the system administrator.
- Before resizing a running VM, ensure that the guest operating system supports CPU and RAM hot plug (refer to *Supported Guest Operating Systems* on page 12). Note that otherwise the guest operating system may become unstable after a resize. To increase CPU or RAM resources for such a guest operating system, you need to stop the virtual machine first.
- Before resizing a running VM, ensure that the guest operating system has the latest updates installed.

4.5.1.1 Enabling or Disabling CPU and RAM Hot Plug for Virtual Machine

1. On the **Virtual machines** screen, ensure that the required virtual machine in the “Shut down” state, and then click it.
2. On the **Overview** tab, click the pencil icon in the **CPU and RAM hot plug** field.

   Note: If you do not see this field, CPU and RAM hot plug is disabled in your project. To enable it, contact your system administrator.

   ---

3. Select or clear the **Enable hot plug** check box, and then click the tick icon to save the changes.

With CPU and RAM hot plug enabled, you can change the flavor of a running VM.

4.5.1.2 Changing Virtual Machine Flavor

1. On the **Virtual machines** screen, click the required virtual machine.
2. On the **Overview** tab, click the pencil icon in the **Flavor** field.
3. In the **Flavor** window, select a new flavor, and then click **Done**.

4.5.2 Configuring Network Interfaces of Virtual Machines

You can add new network interfaces to your virtual machines, edit IP addresses and security groups for the existing interfaces, and remove network interfaces by detaching them.

Limitations:

- You cannot manage network interfaces of shelved VMs.
- A VM that is connected to a dual-stack network always receives an IPv6 address, if the IPv6 subnet is in the SLAAC or DHCPv6 stateless mode.

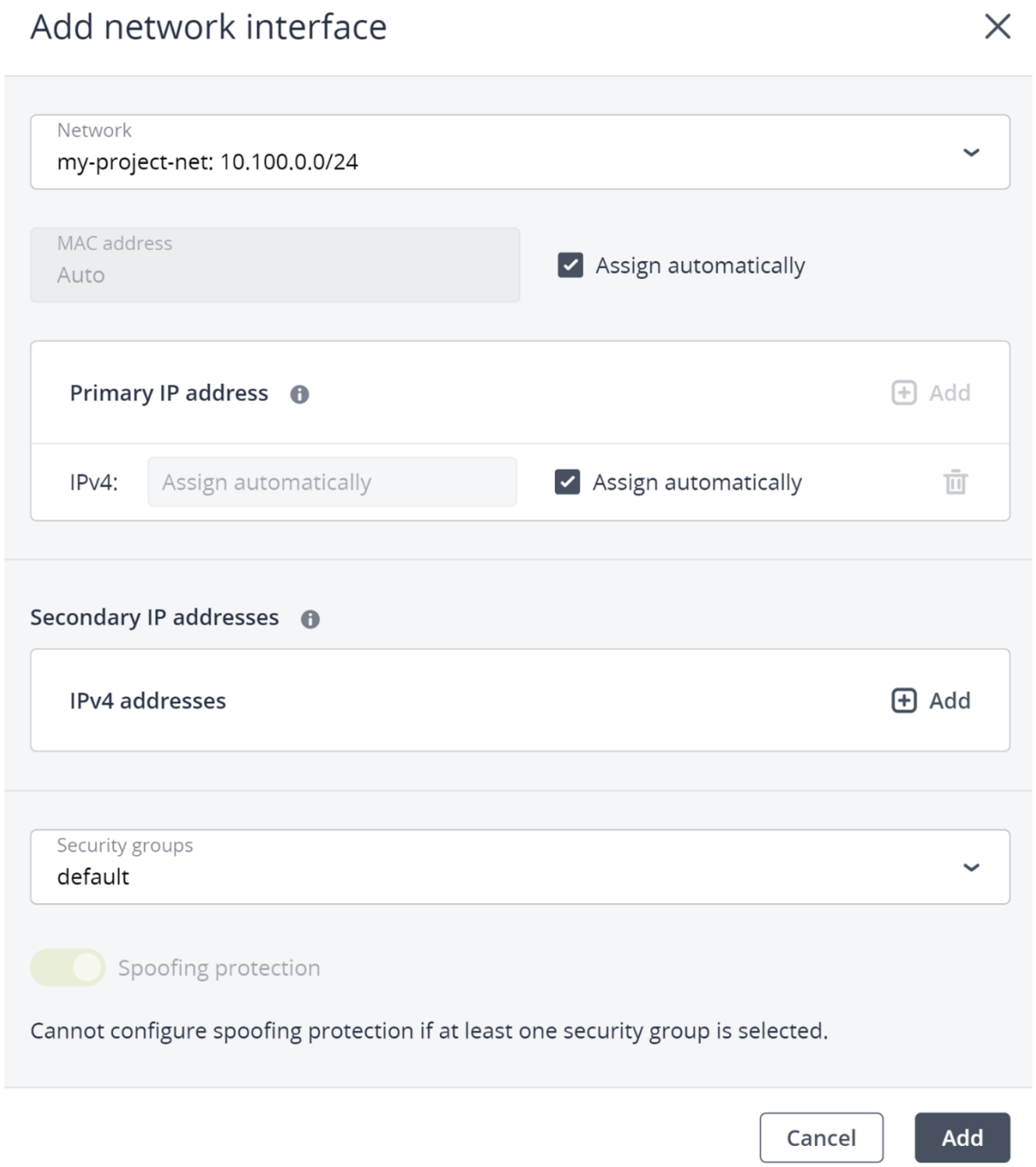
4.5.2.1 Connecting Virtual Machine to Private Network

1. On the **Virtual machines** screen, click the required virtual machine.
2. On the **Overview** tab, click **Edit** in the **Network interfaces** section.
3. In the **Network interfaces** window, click **Add** to attach a network interface.
4. In the **Add network interface** window:
   4.1. Select a compute network to connect to. By default, MAC and primary IP addresses are assigned automatically. To specify them manually, clear the **Assign automatically** checkboxes and ensure that free IP and MAC are selected.
   4.2. A secondary IP can be used by applications inside the VM.

   Note: Virtuozzo DHCP does not configure additional IP for an interface inside the VMs. Therefore any additional IP must be configured manually inside the VM guest.

   4.3. A security group is a port firewall rules list; fault opens all connections. The **default** security group is selected by default. Create other rules on the **Security group** tab if required. Please refer to *Managing Security Group Rules* on page 36.
   4.4. Spoofing protection is enabled by default. It is a security feature that blocks outgoing traffic with source MAC and IP addresses that differs from what was defined above. Disabling the spoofing protections also means disabling any firewall control. Do not disable the spoofing protection. If required, deselect the security group.

4.5. Click **Add**.



5. Click **Done** to finish editing VM network interfaces and save your changes.

4.5.2.2 Editing Network Interface of Virtual Machine

1. On the **Virtual machines** screen, click the required virtual machine.
2. On the **Overview** tab, click **Edit** in the **Network interfaces** section.
3. In the **Network interfaces** window, click the ellipsis button next to the interface you want to edit, and then click **Edit**.
4. In the **Edit network interface** window, modify the network interface parameters as follows:
   - Change the primary IP address. To update the address inside the VM guest OS, restart the network interface.
   - Add or remove secondary IP addresses.
   - Modify security groups assigned to the VM.

   After updating the required parameters, click **Save**.
5. Click **Done** to finish editing VM network interfaces and save your changes.

4.5.2.3 Detaching Network Interface from Virtual Machine

1. On the **Virtual machines** screen, click the required virtual machine.
2. On the **Overview** tab, click **Edit** in the **Network interfaces** section.
3. In the **Network interfaces** window, click the ellipsis button next to the interface you want to detach, and then click **Remove**.
4. Click **Done** to finish editing VM network interfaces and save your changes.

4.5.3 Configuring Virtual Machine Volumes

You can add new volumes to your virtual machines, attach existing volumes, and detach unneeded volumes from virtual machines.

Limitations:

- You cannot change, detach, or delete the boot volume.
- You can only attach and detach non-boot volumes.

- You cannot manage volumes of shelved VMs.

Prerequisites:

To be able to use volumes attached to VMs, they must be initialized inside the guest OS by standard means.

4.5.3.1 Attaching Volume to Virtual Machine

1. On the **Virtual machines** screen, click the required virtual machine.
2. On the **Overview** tab, click the pencil icon in the **Disks** field.
3. In the **Volumes** window:
   - Click **Attach** to attach an existing volume, and then select the volume in the **Attach** volume window.
   - Click **Add** to create a new volume, and then specify the volume name, size, and storage policy. The created volume will be automatically added to the VM disks.
4. Click **Done** to finish editing VM disks and save your changes.

4.5.3.2 Detaching Volume from Virtual Machine

1. On the **Virtual machines** screen, click the required virtual machine.
2. On the **Overview** tab, click the pencil icon in the **Disks** field.
3. In the **Volumes** window:
   - Click **Detach** to detach a volume from a stopped virtual machine.
   - Click **Force detach** to detach a volume from a running virtual machine.

   Note: There is a risk of data loss.

   ---

4. Click **Done** to finish editing VM disks and save your changes.

4.6 Monitoring Virtual Machines

Prerequisites:

- Virtual machines are created, as described in *Creating Virtual Machines* on page 12.

4.6.1 Monitoring 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.

4.7 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.

Prerequisites:

- Virtual machines are created, as described in *Creating Virtual Machines* on page 12.

4.7.1 Shelving Virtual Machine

1. Click the desired virtual machine.
2. If the VM is stopped, click **Shelve** on its right pane.
3. If the VM is running or suspended, click **Shut down** or **Power off** on its right pane, and then select **Shelve virtual machine** in the confirmation window.

4.7.2 Spawning Shelved VM on Node with Enough Resources to Host It

1. Click a shelved virtual machine.
2. On the VM right pane, click **Unshelve**.

4.8 Rescuing Virtual Machines

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.

Limitations:

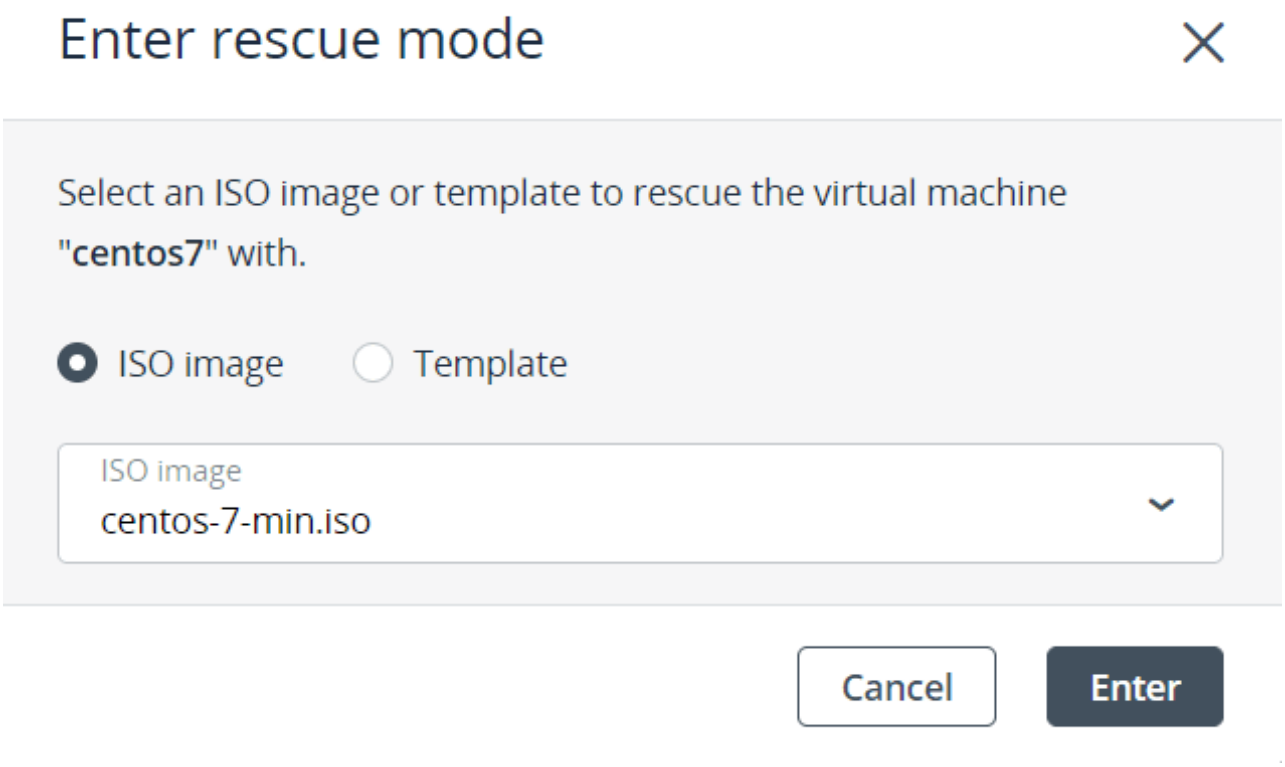
- You can send a VM to the rescue mode only if its current status is “Active” or “Shut down”.
- There are only three actions available for the VM in the rescue mode: **Console**, **Exit rescue mode**, and **Delete**.
- If a rescue image has cloud-init installed, then the VM booted from it can be accessed with the same SSH key that was used for its creation.

Prerequisites:

- Virtual machines are created, as described in *Creating Virtual Machines* on page 12.

4.8.1 Putting Virtual Machine to the Rescue Mode

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”.

4.8.2 Returning Virtual Machine to Normal Operation

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.

4.8.3 Exiting the Rescue Mode for Windows VM

There might be an issue of exiting the rescue mode for a Windows VM. If in the rescue mode you set the original system disk online, its ID becomes the same as that of the rescue disk. Then, when you try to exit the rescue mode, the boot loader cannot find the proper boot disk. To resolve the ID conflict, follow the steps:

1. With the VM in the rescue mode, open the **Disk Management** window and note the numbers of the original system disk (offline) and the rescue disk (online). Set the original system disk to **Online**.
2. To edit the boot configuration, enter the following command in the **Command Prompt** window:

   ```
   > bcdedit /store <the original system disk name>:\boot\bcd
   ```

3. Review the output and check that the rescue disk is the target for objects in the output (`partition=<the rescue disk name>`).

   If the objects do not point to drive C, fix it with the following commands:

   ```
   > bcdedit /store <the original system disk name>:\boot\bcd \
   /set {default} osdevice partition=<the rescue disk name>:
   > bcdedit /store <the original system disk name>:\boot\bcd \
   /set {default} device partition=<the rescue disk name>:
   > bcdedit /store <the original system disk name>:\boot\bcd \
   /set {bootmgr} device partition=<the rescue disk name>:
   > bcdedit /store <the original system disk name>:\boot\bcd \
   /set {memdiag} device partition=<the rescue disk name>:
   ```

4. To view the available disks, enter the following commands in the command line:

   ```
   > DISKPART
   > LIST DISK
   ```

   Match the disk number and name to those displayed in the **Disk Management** window.

5. To get the ID of the rescue disk, run the following commands:

   ```
   > SELECT DISK <the rescue disk number>
   > UNIQUEID DISK
   ```

   Record the disk ID, you will need it later.

6. Change this ID by using the following command:

```
> UNIQUEID DISK id=<any hex value of 8 characters>
```

   Make sure that the value has changed with the `UNIQUEID DISK` command.

7. Assign the ID that you recorded previously to the original system disk:

```
> SELECT DISK <the original system disk number>
> UNIQUEID DISK id=<the recorded disk ID>
```

   Make sure that the value has changed with the `UNIQUEID DISK` command.

You should now be able to exit the rescue mode.

4.9 Troubleshooting Virtual Machines

If a virtual machine fails to deploy, review 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 virtual machine is in the error state, examine the VM history in the **History** tab on the VM right pane. The event log will contain all of the VM management operations performed by users in the user or command-line interface. You can expand each log entry to view operation details by clicking the arrow icon next to it. The details include the operation name, date and time, status, initiator, and request ID.

If a virtual machine is stuck in a failed or transitional state, reset the VM to its last stable state (active, shut down or shelved):

1. Click the stuck VM.
2. On the VM right pane, click **Reset state**.

If a virtual machine fails to boot, examine the VM console log by clicking **Download console log** on the VM right pane.

4.10 Deleting Virtual Machines

Limitations:

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

Prerequisites:

- Virtual machines are created, as described in *Creating Virtual Machines* on page 12.

4.10.1 Removing One Virtual Machine

1. Click the ellipsis button next to a VM you want to delete, and then click **Delete**.
2. Click **Delete** in the confirmation window.

4.10.2 Removing Multiple Virtual Machines

1. Select the check boxes next to VMs you want to delete.
2. Over the VM list, click **Delete**.
3. Click **Delete** in the confirmation window.

CHAPTER 5

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.

Limitations:

- You can manage only IPv4 security group rules.

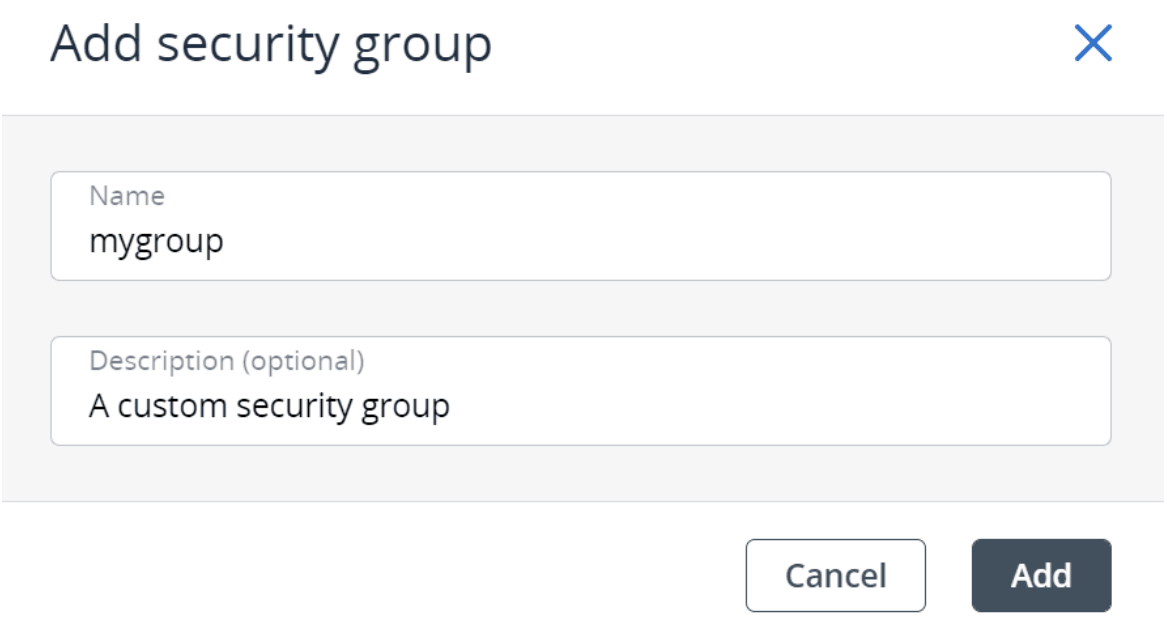
5.1 Creating and Deleting Security Groups

Limitations:

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

5.1.1 Creating Security Group

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**.



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

5.1.2 Deleting Security Group

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.

5.2 Managing Security Group Rules

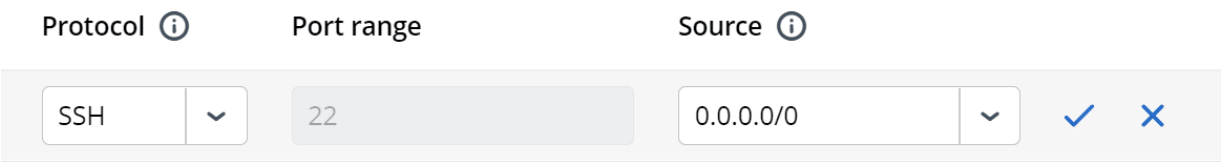
You can modify security groups by adding and removing rules. Editing rules is not available. If you need to change the existing rule, remove it and recreate with the required parameters.

Prerequisites:

- You have a security group created, as described in *Creating and Deleting Security Groups* on page 35.

5.2.1 Adding Rule to Security Group

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:
   3.1. Select a protocol from the list or enter a number from 0 to 255.
   3.2. Enter a single port or a port range. Some protocols already have a predefined port range. For example, the port for SSH is 22.
   3.3. Select a predefined subnet CIDR or an existing security group.



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.

5.2.2 Removing Rule from Security Group

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.

5.3 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.

Limitations:

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

5.3.1 Viewing Virtual Machines Assigned to Security Group

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.

5.3.2 Assigning Security Group to Virtual Machine

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**.

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

CHAPTER 6

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:

Kubernetes version	Underlying OS	Container runtime	Network plugin
v1.21.3, v1.22.2	Fedora 36 CoreOS	Docker 20.10.12	Flannel with VXLAN
v1.23.5, v1.24.3		containerd 1.6.6	

Limitations:

- Kubernetes versions 1.15.x–1.20.x are no longer supported. Kubernetes clusters created with these versions are marked with the **Deprecated** tag.
- Kubernetes cluster certificates are issued for five years. To renew the certificates, use the openstack coe ca rotate command, as described in the OpenStack documentation.

6.1 Creating and Deleting Kubernetes Clusters

Limitations:

- Only users that have access to the corresponding project can perform operations with Kubernetes clusters.

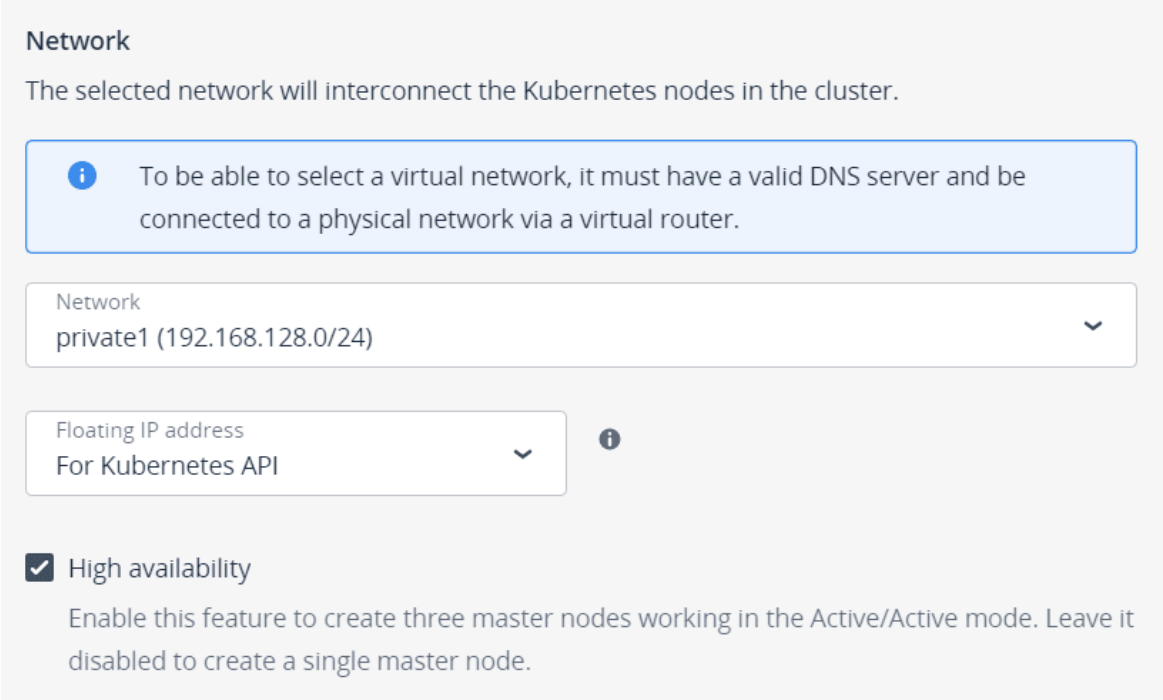
Prerequisites:

- The Kubernetes-as-a-service component is installed by a system administrator. It can be deployed along with the compute cluster or later.
- You have 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 is added. It will be installed on both the master and worker nodes.
- You have enough resources for all of the Kubernetes nodes, taking their flavors into account.
- 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

6.1.1 Creating Kubernetes Cluster

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. Enter the cluster name, and then select a Kubernetes version and an SSH key.
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).

   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.

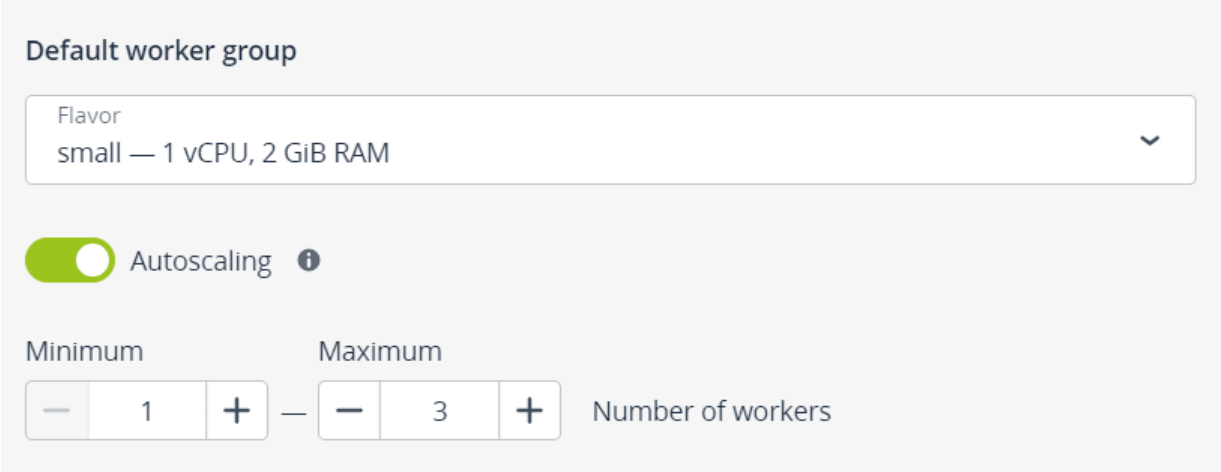


4. In the **Master node** section, select a flavor for the master node. For production clusters, it is strongly recommended to use a flavor with at least 2 vCPUs and 8 GiB of RAM.
5. Optionally, enable **Integrated monitoring** to automatically deploy the cluster-wide monitoring solution, which includes the following components: Prometheus, Alertmanager, and Grafana.

Note: This feature is experimental and not supported in production environments.

---

6. In the **Container volume** section, select a storage policy, and then enter the size for volumes on both master and worker nodes.
7. In the **Default worker group** section, select a flavor for each worker, and then decide whether you want to allow automatic scaling of the worker group:
   - With **Autoscaling** enabled, the number of workers will be automatically increased if there are pods stuck in the pending state due to insufficient resources, and reduced if there are workers with no pods running on them. For scaling of the worker group, set its minimum and maximum size.
   - With **Autoscaling** disabled, the number of worker nodes that you set will be permanent.



8. In the **Labels** section, enter labels that will be used to specify supplementary parameters for this Kubernetes cluster in the `key=value` format. For example: `selinux_mode=permissive`. Currently, only the `selinux` label is supported. You can use other labels at your own risk. To see the full list of supported labels, refer to the OpenStack documentation.
9. 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. You can also access the Kubernetes master and worker nodes via SSH, by using the assigned SSH key and the user name **core**.

6.1.2 Deleting Kubernetes Cluster

Click the required Kubernetes cluster on the **Kubernetes clusters** screen and click **Delete**. The master and worker VMs will be deleted along with their volumes

6.2 Managing Kubernetes Worker Groups

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.

Limitations:

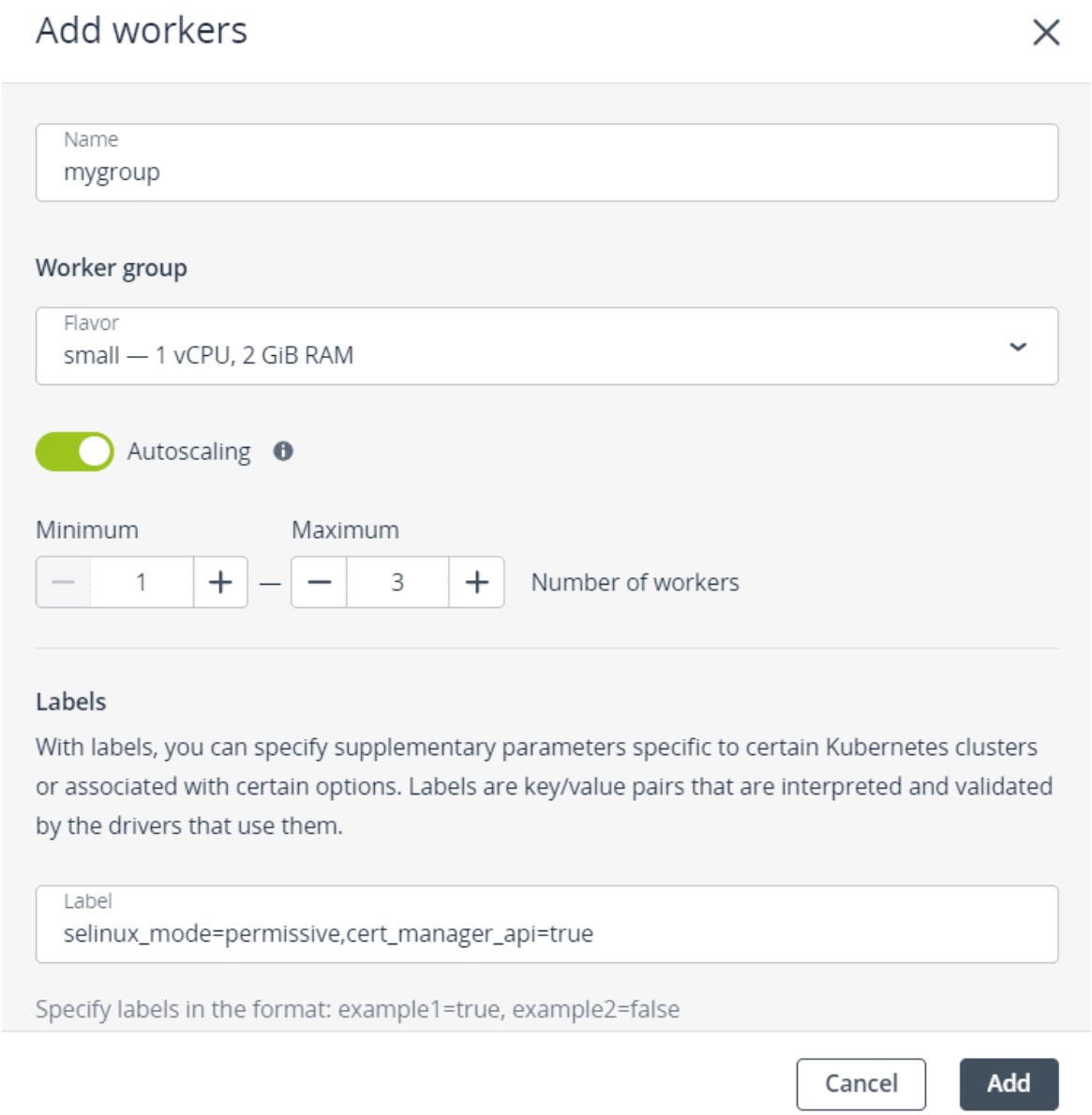
- Worker groups are not available for Kubernetes version 1.15.x.
- The default worker group cannot be deleted.

Prerequisites:

- A Kubernetes cluster is created, as described in *Creating and Deleting Kubernetes Clusters* on page 39.

6.2.1 Adding Worker Group

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, specify a name for the group.
5. In the **Worker group** section, select a flavor for each worker, and then decide whether you want to allow automatic scaling of the worker group:
    - With **Autoscaling** enabled, the number of workers will be automatically increased if there are pods stuck in the pending state due to insufficient resources, and reduced if there are workers with no pods running on them. For scaling of the worker group, set its minimum and maximum size.
    - With **Autoscaling** disabled, the number of worker nodes that you set will be permanent.
6. In the **Labels** section, enter labels that will be used to specify supplementary parameters for this Kubernetes cluster in the `key=value` format. For example: `selinux_mode=permissive`. Currently, only the `selinux` label is supported. You can use other labels at your own risk. To see the full list of supported labels, refer to the OpenStack documentation.
7. Click **Add**.



When the worker group is created, you can assign pods to these worker nodes, as explained in *Assigning Kubernetes Pods to Specific Nodes* on page 56.

6.2.2 Editing the Number of Workers in Group

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**.

3. In the **Edit workers** window, enable or disable **Autoscaling**, or change the number of workers in the group.
4. Click **Save**.

6.2.3 Deleting 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.

6.3 Updating Kubernetes Clusters

When a new Kubernetes version becomes available, you can update your Kubernetes cluster to it. An update is non-disruptive for Kubernetes worker nodes, which means that these nodes are updated one by one, with the data availability unaffected. The Kubernetes API will be unavailable during an update, unless high availability is enabled for the master node.

Limitations:

- You cannot update Kubernetes clusters with version 1.15.x to newer versions.
- You cannot manage Kubernetes clusters in the self-service panel during an update.

Prerequisites:

- A Kubernetes cluster is created, as described in *Creating and Deleting Kubernetes Clusters* on page 39.

6.3.1 Updating Kubernetes Cluster

1. Click a Kubernetes cluster that is marked with the **Update available** tag.
2. On the Kubernetes cluster pane, click **Update** in the **Kubernetes version** field.
3. In the **Update** window, select a Kubernetes version to update to and follow the provided link to read about API resources that are deprecated or obsoleted in the selected version. Then, click **Update**.
4. In the confirmation window, click **Confirm**. The update process will start.

Note: Do not manage Kubernetes virtual machines during the update as it may lead to disruption of the

update process and cluster inoperability.

6.4 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.

6.4.1 Creating Storage Classes

In Virtuozzo Hybrid Cloud, 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.

6.4.1.1 Creating 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.

6.4.2 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.

Prerequisites:

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

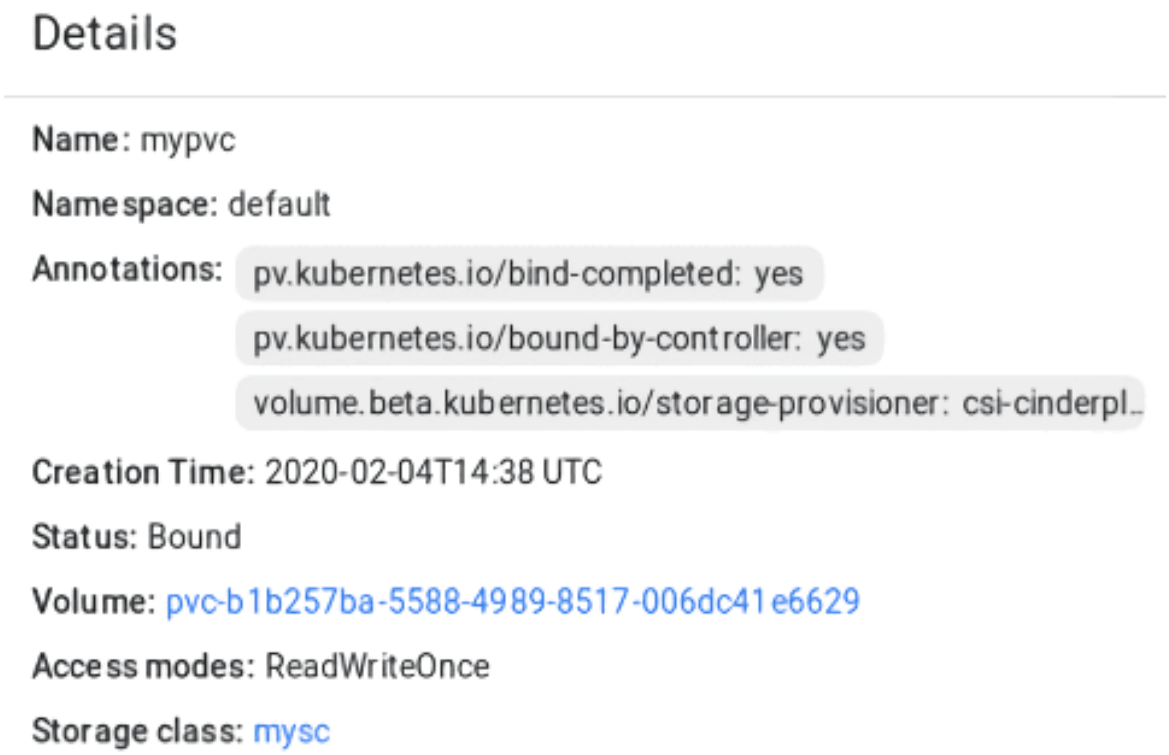
6.4.2.1 Provisioning PV to Pod Dynamically

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* on page 46.
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.



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
```

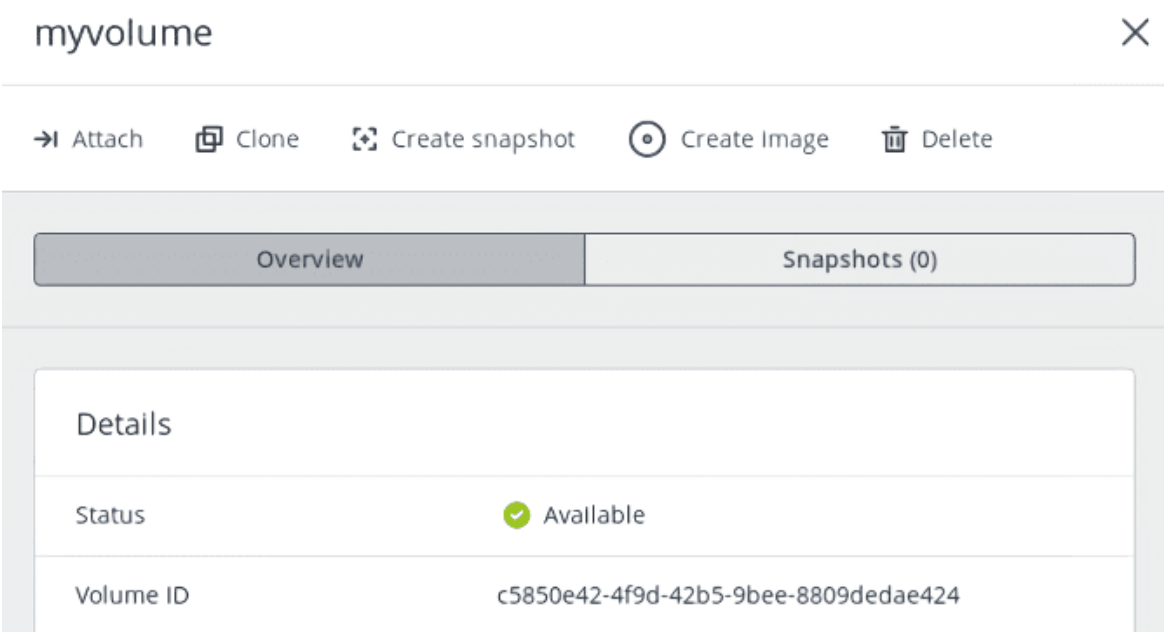
This configuration file describes the pod `nginx` that uses the persistent volume claim `mypvc`. The persistent volume bound to the claim will be accessible at `/var/lib/www/html` inside the `nginx` container.

6.4.3 Statically Provisioning Persistent Volumes

You can mount existing compute volumes to pods using static provisioning of persistent volumes.

6.4.3.1 Mounting Compute Volume

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* on page 46.
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:
```

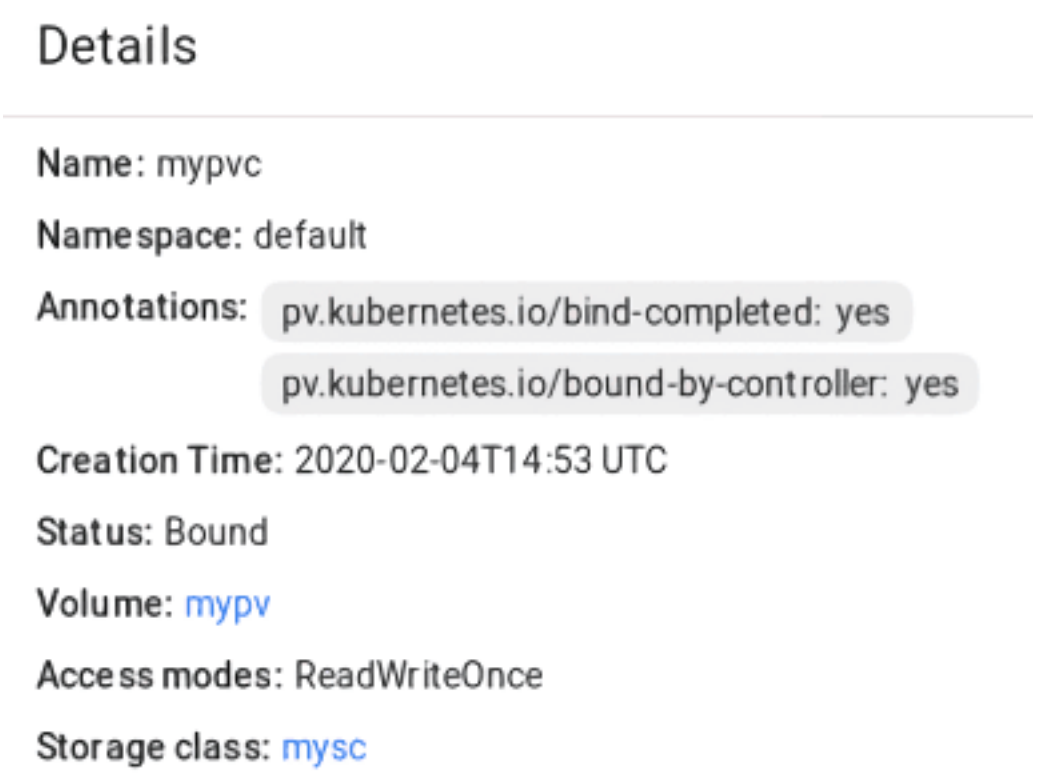
```
    driver: cinder.csi.openstack.org
    fsType: ext4
    volumeHandle: c5850e42-4f9d-42b5-9bee-8809dedae424
  persistentVolumeReclaimPolicy: Delete
  storageClassName: mysc
```

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:

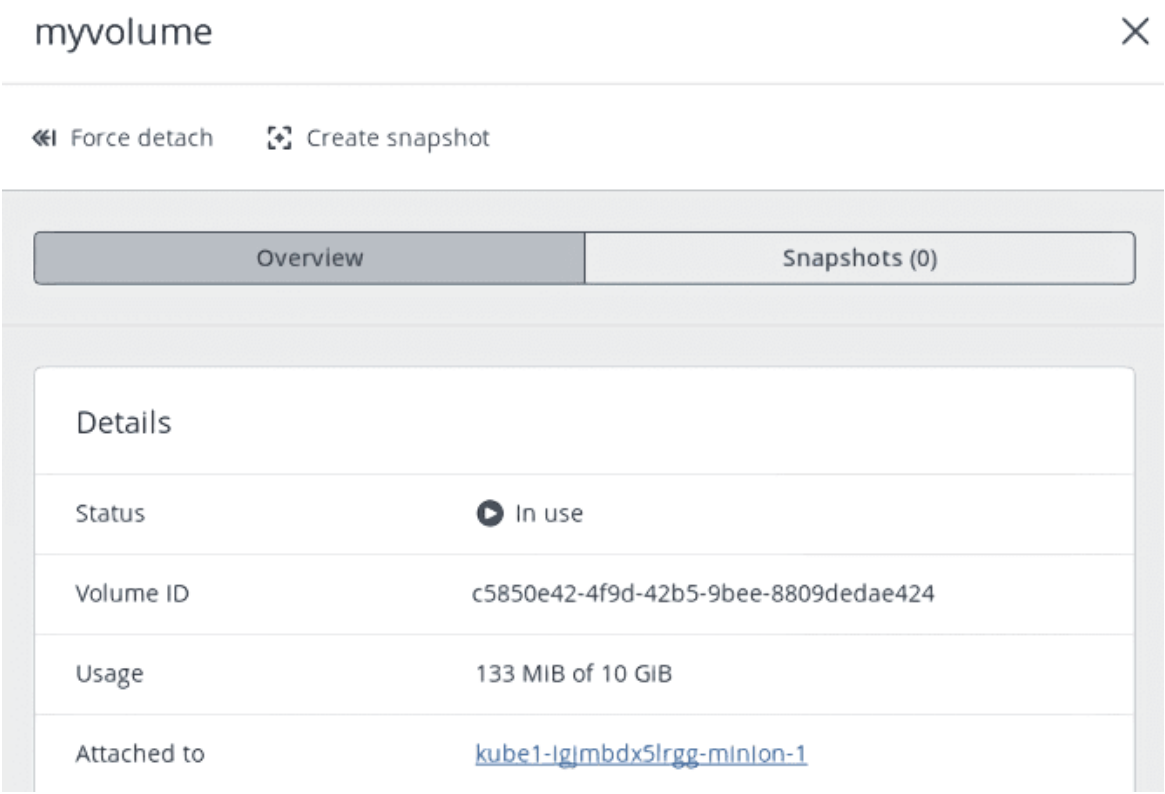
```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: mysc
```

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



6. Create a pod and specify the PVC as its volume. Use the example from Step 4 in *Dynamically Provisioning Persistent Volumes* on page 46.

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



6.4.4 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.

6.4.4.1 Terminating 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.

6.5 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.

Prerequisites:

- To be able to assign a specific floating IP address to an external load balancer during its deployment, this floating IP address must be created in advance, as described in *Managing Floating IP Addresses* on page 115.

6.5.1 Creating Service with External Load Balancer

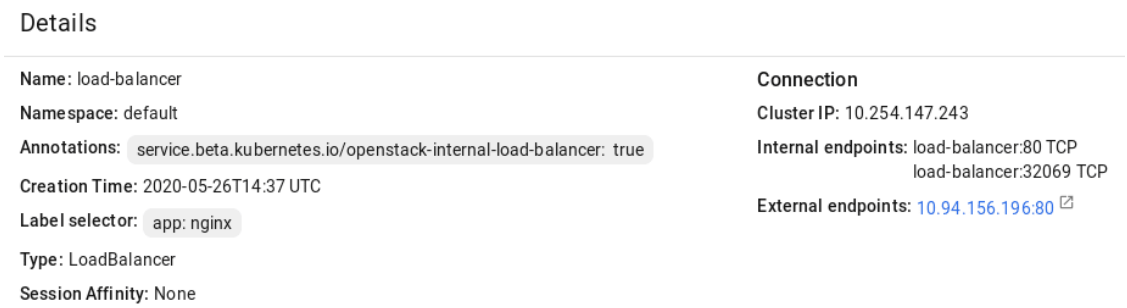
1. Access the Kubernetes cluster via the dashboard. Click **Kubernetes access** for instructions.
2. On the Kubernetes dashboard, create a deployment and service of the **LoadBalancer** type. To do it, click **+ Create** and specify a YAML file that defines these objects. For example:
   - If you have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



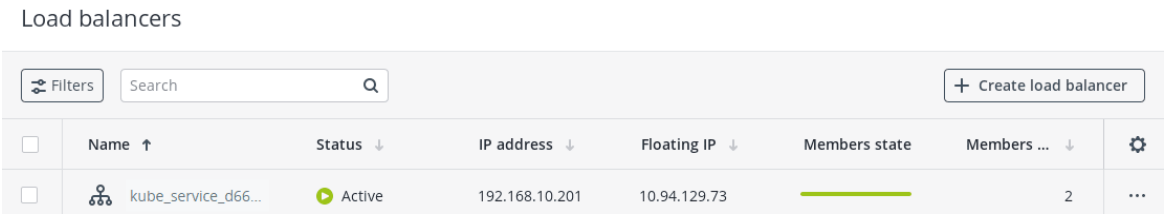
- If you have deployed the Kubernetes cluster in a virtual network linked to a physical one via a virtual router, you can use the YAML file above without the `annotations` section for the `load-balancer` service. The created load balancer will receive a floating IP address from the physical network and can be accessed at this external endpoint. To use a specific floating IP address, create it in the self-service panel in advance, and then specify it with the `loadBalancerIP` parameter:

```
<...>
---
kind: Service
apiVersion: v1
metadata:
  name: load-balancer
spec:
  selector:
    app: nginx
```

```
type: LoadBalancer
loadBalancerIP: 10.10.10.100
ports:
- port: 80
  targetPort: 80
  protocol: TCP
```

- If you want to choose whether to create highly available load balancers for your service or not, you can make use of load balancer flavors. To specify a flavor for a load balancer add `loadbalancer.openstack.org/flavor-id: <flavor-id>` to the `annotations` section. The flavor ID can be obtained from your system administrator.

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



6.6 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.

6.6.1 Creating Pod That Will Be Scheduled on Specific Node

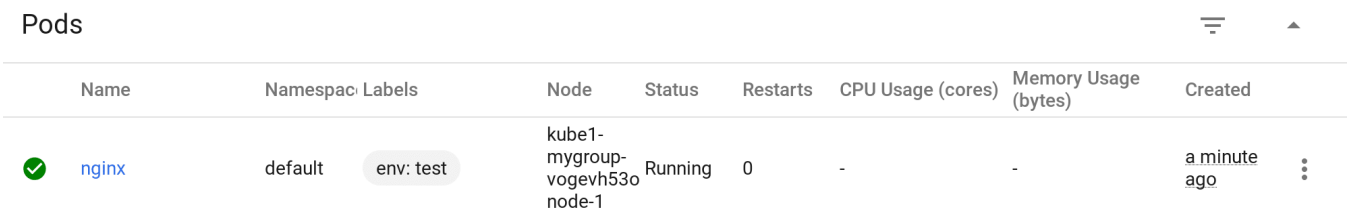
Click **+ Create** on the Kubernetes dashboard and specify a YAML file that defines this object. For example:

```
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.



6.7 Monitoring Kubernetes Clusters

Note: This feature is experimental and not supported in production environments.

If you have enabled integrated monitoring during your Kubernetes cluster deployment, that means that the cluster has the `monitoring_enabled=true` label and the following components installed:

- Prometheus for data collection, storage, and search:
  - `node-exporter` exposes various server-level and OS-level metrics.
  - `kube-state-metrics` generates metrics on the state of Kubernetes objects.
- Alertmanager for alarm aggregation, processing, and dispatch.
- Grafana server for metrics visualization.

For instructions on how to create and configure Alertmanager and Prometheus instances, refer to the kube-prometheus documentation.

The Grafana server is accessible from within a Kubernetes cluster at the **magnum-grafana.kube-system.svc.cluster.local** DNS name and TCP port 80.

The metrics on the state of Kubernetes objects are exported at the **/metrics** HTTP endpoint on the listening

port: **magnum-kube-state-metrics.kube-system.svc.cluster.local:8080/metrics**. The metrics can be consumed either by Prometheus itself or by a scraper that is able to scrape a Prometheus client endpoint. For the list of exposed metrics, refer to kube-state-metrics documentation.

Prerequisites:

- A Kubernetes cluster with enabled integrated monitoring is created, as described in *Creating and Deleting Kubernetes Clusters* on page 39.

6.7.1 Accessing the Kubernetes Grafana Dashboards

1. On the **Kubernetes clusters** screen, click a Kubernetes cluster.
2. On the cluster right pane, click **Download kubeconfig**. The .kubeconfig file will be downloaded to your client machine.
3. On your client machine, install and set up the kubectl tool, to be able to run commands against Kubernetes clusters, as described in the official documentation.
4. Specify the path to your Kubernetes configuration file in the `KUBECONFIG` environment variable:

```
# export KUBECONFIG=<path_to_kubeconfig>
```

5. Check that the `kube-prometheus` stack is installed:

```
# kubectl --namespace kube-system get pods -l "release=magnum"
NAME                                                  READY   STATUS    RESTARTS   AGE
magnum-kube-prometheus-sta-operator-85f757c5dc-ckllb  1/1     Running   0          3d17h
magnum-kube-state-metrics-5cc46cbc5f-tclcv            1/1     Running   0          3d17h
magnum-prometheus-node-exporter-99kfc                 1/1     Running   0          3d3h
magnum-prometheus-node-exporter-gwgzr                 1/1     Running   0          3d17h
magnum-prometheus-node-exporter-q2pm2                 1/1     Running   0          3d17h
magnum-prometheus-node-exporter-sqsl7                 1/1     Running   0          2d22h
```

6. Obtain the password of the `admin` user:

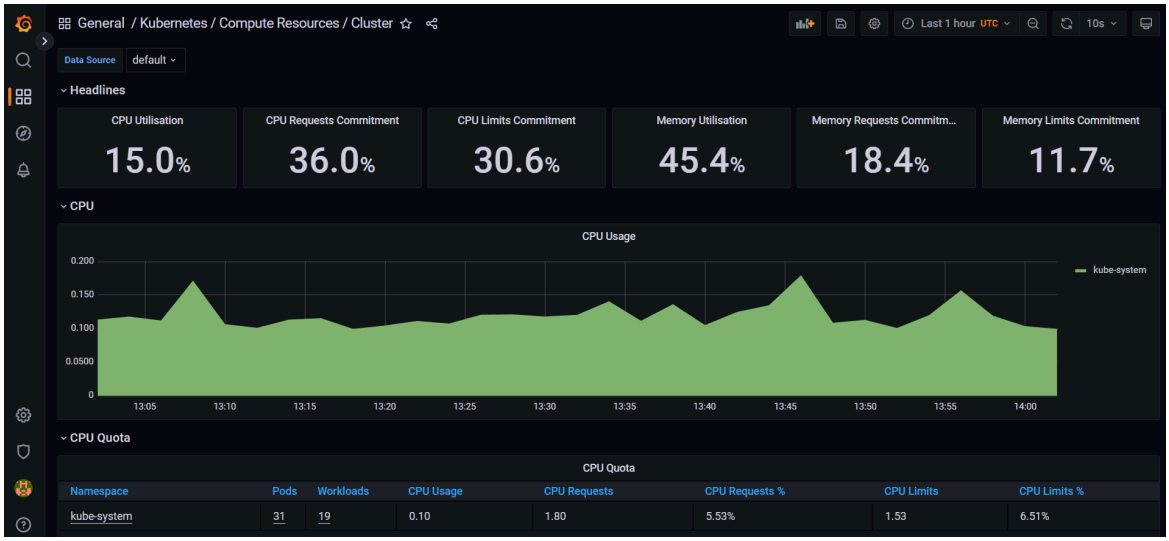
```
# kubectl get secret --namespace kube-system magnum-grafana \
-o jsonpath="{.data.admin-password}" | base64 --decode ; echo
```

7. Configure the port forwarding for the Grafana pod:

```
# kubectl --namespace kube-system port-forward service/magnum-grafana 3000:80
```

8. Log in to http://localhost:3000 under the admin user by specifying its username and password obtained in step 6.

9. In the left menu, click **Dashboards** > **Browse**, and then select the dashboard you want to view.



6.7.2 Accessing the Prometheus User Interface

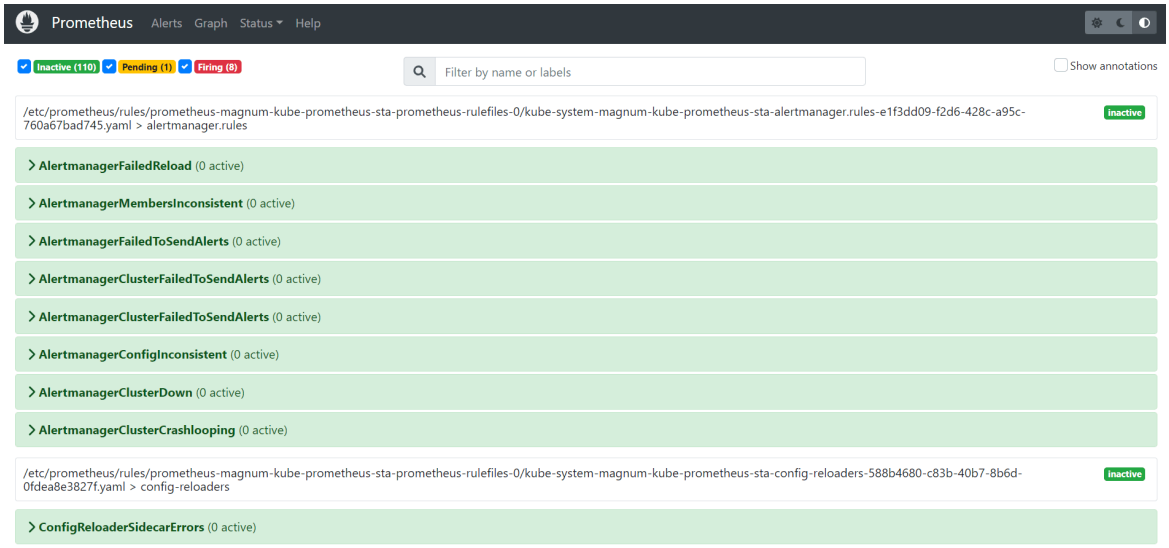
1. On the **Kubernetes clusters** screen, click a Kubernetes cluster.
2. On the cluster right pane, click **Download kubeconfig**. The `.kubeconfig` file will be downloaded to your client machine.
3. On your client machine, install and set up the `kubectl` tool, to be able to run commands against Kubernetes clusters, as described in the official documentation.
4. Specify the path to your Kubernetes configuration file in the `KUBECONFIG` environment variable:

```
# export KUBECONFIG=<path_to_kubeconfig>
```

5. Configure the port forwarding for the Prometheus pod:

```
# kubectl --namespace kube-system port-forward service/magnum-kube-prometheus-sta-prometheus 9
```

6. Visit http://localhost:9090/graph to use the Prometheus expression browser and to graph expressions. You can also navigate to http://localhost:9090/metrics to view the list of exported metrics, or http://localhost:9090/alerts to view the alerting rules.



6.7.3 Accessing the Alertmanager User Interface

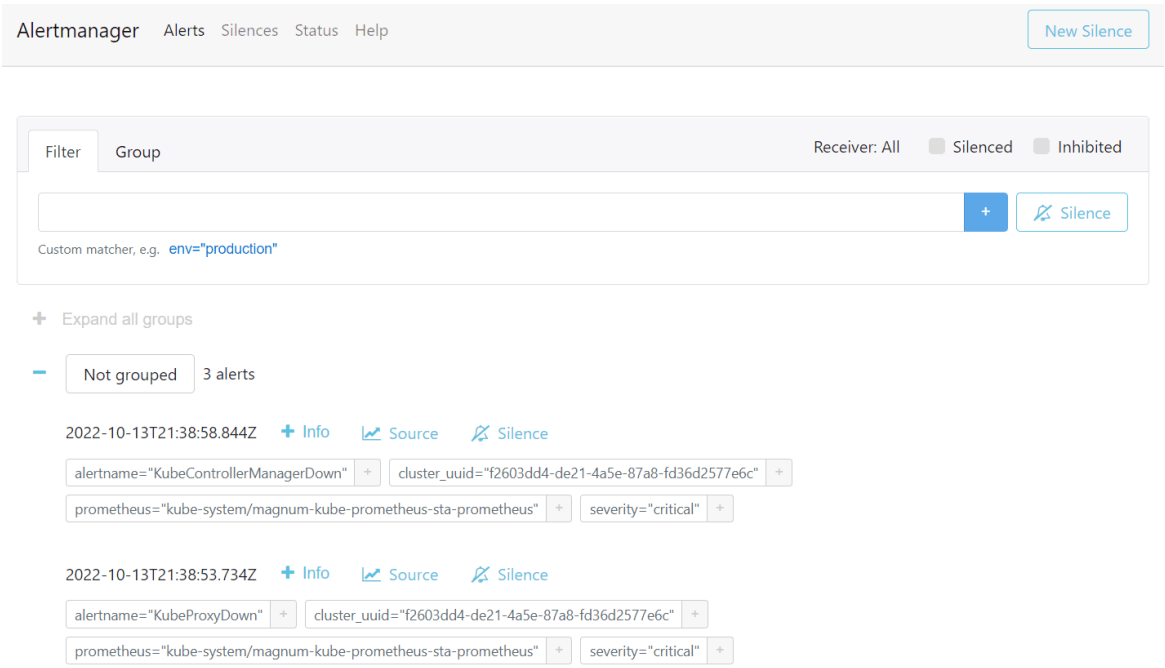
1. On the **Kubernetes clusters** screen, click a Kubernetes cluster.
2. On the cluster right pane, click **Download kubeconfig**. The `.kubeconfig` file will be downloaded to your client machine.
3. On your client machine, install and set up the `kubectl` tool, to be able to run commands against Kubernetes clusters, as described in the official documentation.
4. Specify the path to your Kubernetes configuration file in the `KUBECONFIG` environment variable:

```
# export KUBECONFIG=<path_to_kubeconfig>
```

5. Configure the port forwarding for the Alertmanager pod:

```
# kubectl --namespace kube-system port-forward service/magnum-kube-prometheus-sta-alertmanager
```

6. Visit http://localhost:9093 to access the Alertmanager user interface.


Alertmanager
Alerts
Silences
Status
Help
New Silence
Filter
Group
Receiver: All
Silenced
Inhibited
Silence
Custom matcher, e.g. env="production"
Expand all groups
Not grouped
3 alerts
2022-10-13T21:38:58.844Z
Info
Source
Silence
alertname="KubeControllerManagerDown"
cluster_uuid="f2603dd4-de21-4a5e-87a8-fd36d2577e6c"
prometheus="kube-system/magnum-kube-prometheus-sta-prometheus"
severity="critical"
2022-10-13T21:38:53.734Z
Info
Source
Silence
alertname="KubeProxyDown"
cluster_uuid="f2603dd4-de21-4a5e-87a8-fd36d2577e6c"
prometheus="kube-system/magnum-kube-prometheus-sta-prometheus"
severity="critical"

CHAPTER 7

Managing Images

Virtuozzo Hybrid Cloud 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. You can upload an ISO image to the compute cluster.
- A template is a ready boot volume in the QCOW2 format with an installed operating system and applications. Many OS vendors offer templates of their operating systems under the name “cloud images”. You can upload a cloud image from the OS official repository or prepare your own template in the compute cluster.

Prerequisites:

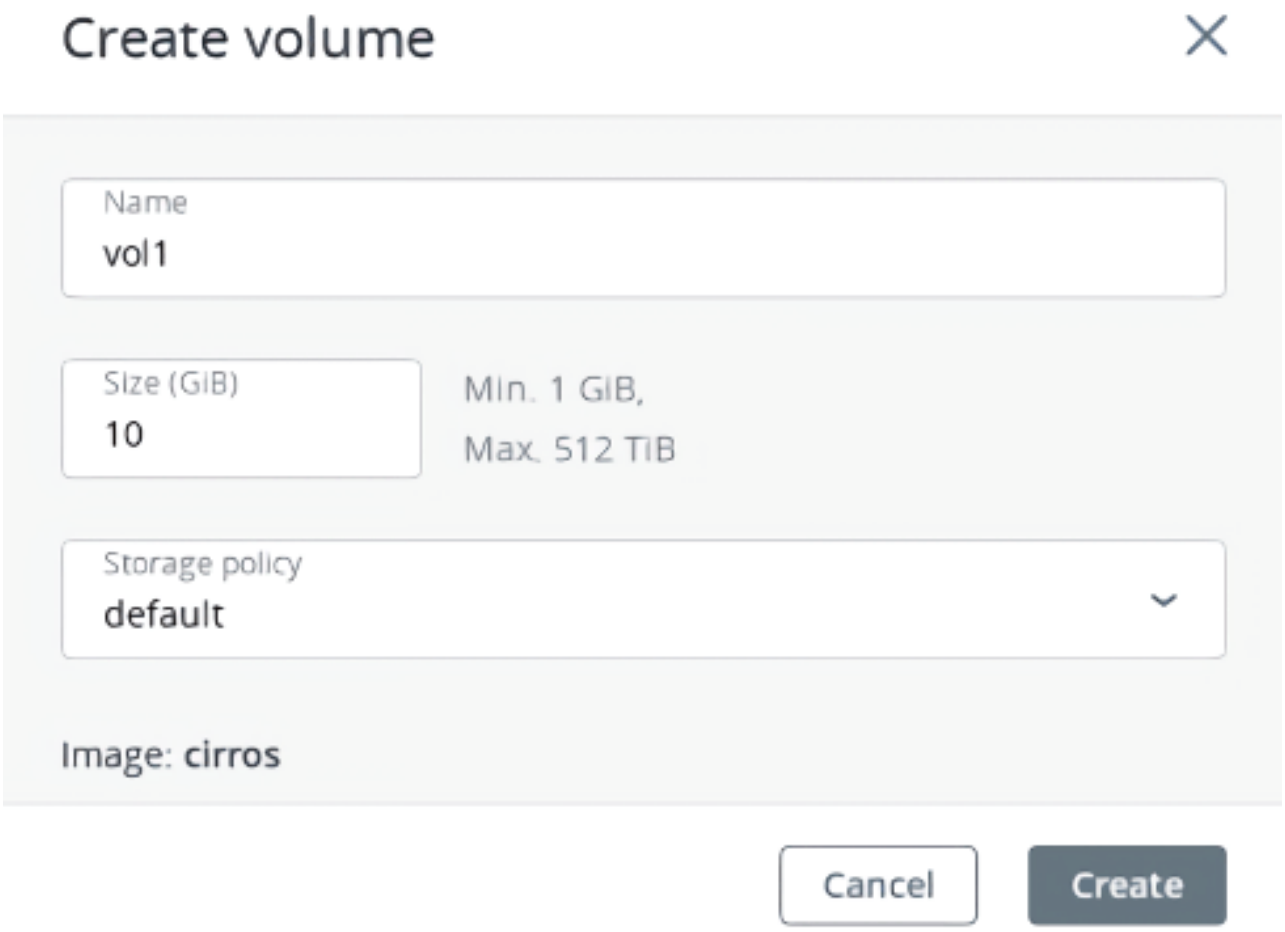
- Knowledge of the supported guest operating systems listed in *Supported Guest Operating Systems* on page 12.

7.1 Creating Volumes from Image

You can create volumes from both ISO images and templates.

7.1.1 Making Volume from Image

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.



4. Click **Create**.

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

CHAPTER 8

Managing Volumes

A volume in Virtuozzo Hybrid Cloud 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.

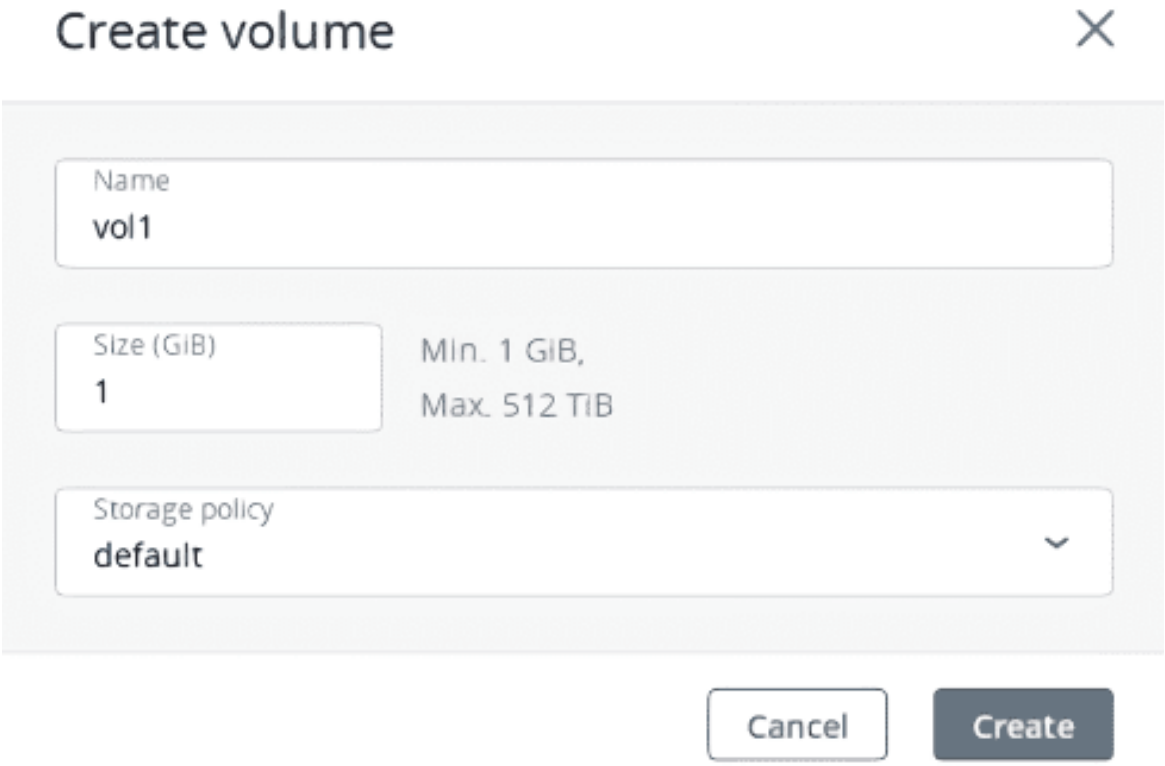
8.1 Creating and Deleting Volumes

Limitations:

- A volume is removed along with all of its snapshots.

8.1.1 Creating Volume

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



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

8.1.2 Removing Volume

1. On the **Volumes** tab, check the status of the volume you want to remove.
2. If the status is “In use”, click the volume, and then click **Force detach**.
3. If the status is “Available”, click the volume, and then click **Delete**.

8.2 Attaching and Detaching Volumes

Limitations:

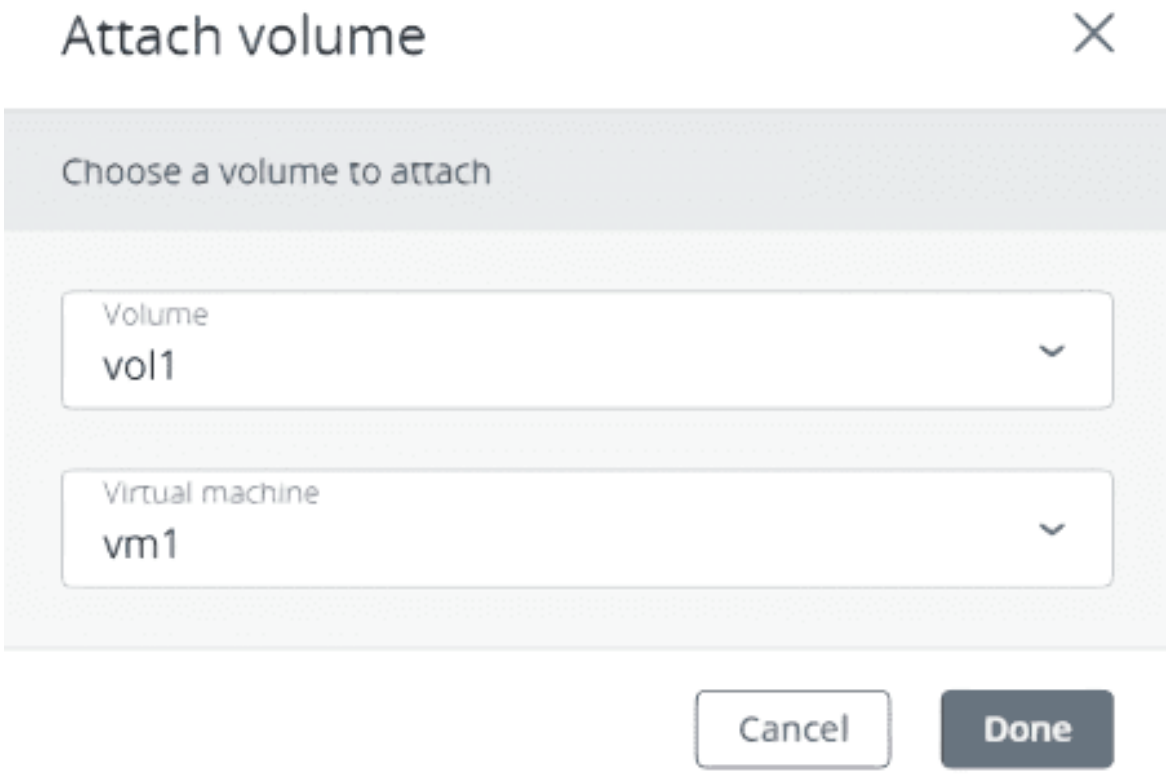
- You can only attach and detach non-boot volumes.

Prerequisites:

- A volume is created, as described in *Creating and Deleting Volumes* on page 64.
- To be able to use volumes attached to VMs, they must be initialized inside the guest OS by standard means.

8.2.1 Attaching Volume to Virtual Machine

1. On the **Volumes** screen, click an unused volume.
2. On the volume right pane, click **Attach**.
3. In the **Attach volume** window, select the VM from the drop-down list, and then click **Done**.



8.2.2 Detaching Volume from Virtual Machine

1. On the **Volumes** screen, click a volume that is in use.
2. If the VM is stopped, click **Detach** on the volume right pane.

3. If the VM is running, click **Force detach** on the volume right pane.

Note: There is a risk of data loss.

8.3 Resizing Volumes

You can change volume size only by increasing it. Volumes can be extended for both running (online resizing) and stopped (offline resizing) virtual machines. Online volume resizing allows users to avoid downtime and enables scaling VM storage capacity on the fly without service interruption.

Limitations:

1. You cannot shrink volumes.
2. During volume resizing, the file system inside the guest OS is not extended.
3. If you revert a volume to a snapshot that was taken before the volume extension, the new volume size will be retained.

Prerequisites:

- A volume is created, as described in *Creating and Deleting Volumes* on page 64.

8.3.1 Extending Volume

1. On the **Volumes** screen, click a volume.
2. Click the pencil icon in the **Size** field.
3. Enter the desired volume capacity, and then click the tick icon.

After the volume is extended, you will need to re-partition the disk inside the guest OS to allocate the added disk space.

8.4 Changing the Storage Policy for Volumes

You can manage compute volume redundancy and performance by changing the storage policy applied to the volume. The storage policy can be changed for volumes attached to both running and stopped virtual machines.

Limitations:

- Only storage policies enabled by project quotas will be available for selection.

Prerequisites:

- A volume is created, as described in *Creating and Deleting Volumes* on page 64.

8.4.1 Changing the Storage Policy for Volumes

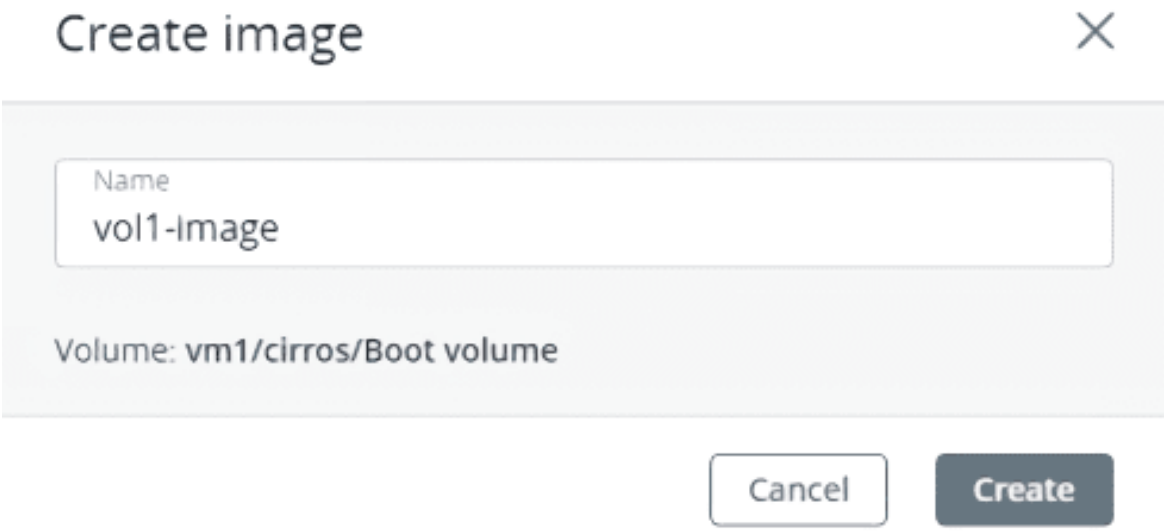
1. On the **Volumes** screen, click a volume.
2. Click the pencil icon in the **Storage policy** field.
3. Select a new storage policy, and then click the tick icon. You can choose only between storage policies with the same redundancy type.

8.5 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.

8.5.1 Creating Template from Boot Volume

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**.



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

8.6 Cloning Volumes

Limitations:

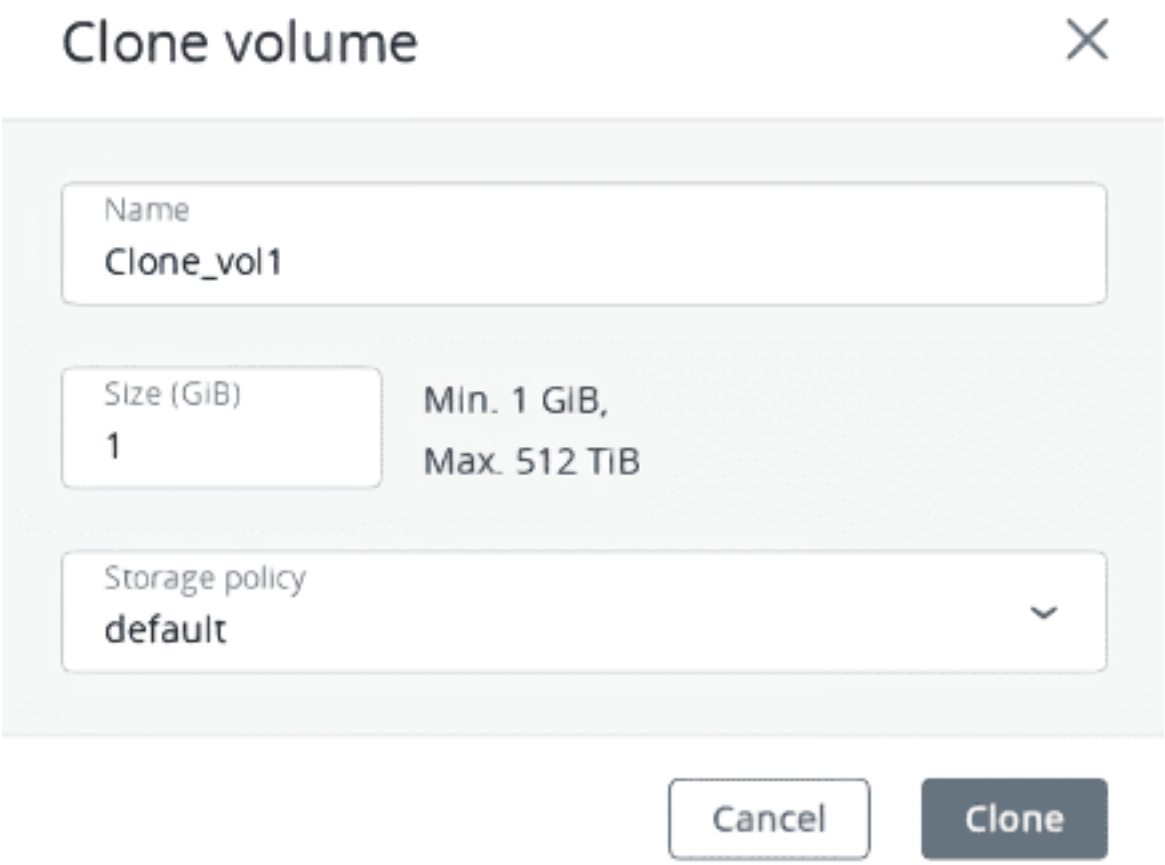
- You can clone volumes that are not attached to VMs or attached to stopped VMs.

Prerequisites:

- A volume is created, as described in *Creating and Deleting Volumes* on page 64.

8.6.1 Cloning Volume

1. On the **Volumes** screen, click a volume.
2. On the volume right pane, click **Clone**.
3. In the **Clone volume** window, specify a volume name, size, and storage policy. Click **Clone**.



8.7 Managing Volume Snapshots

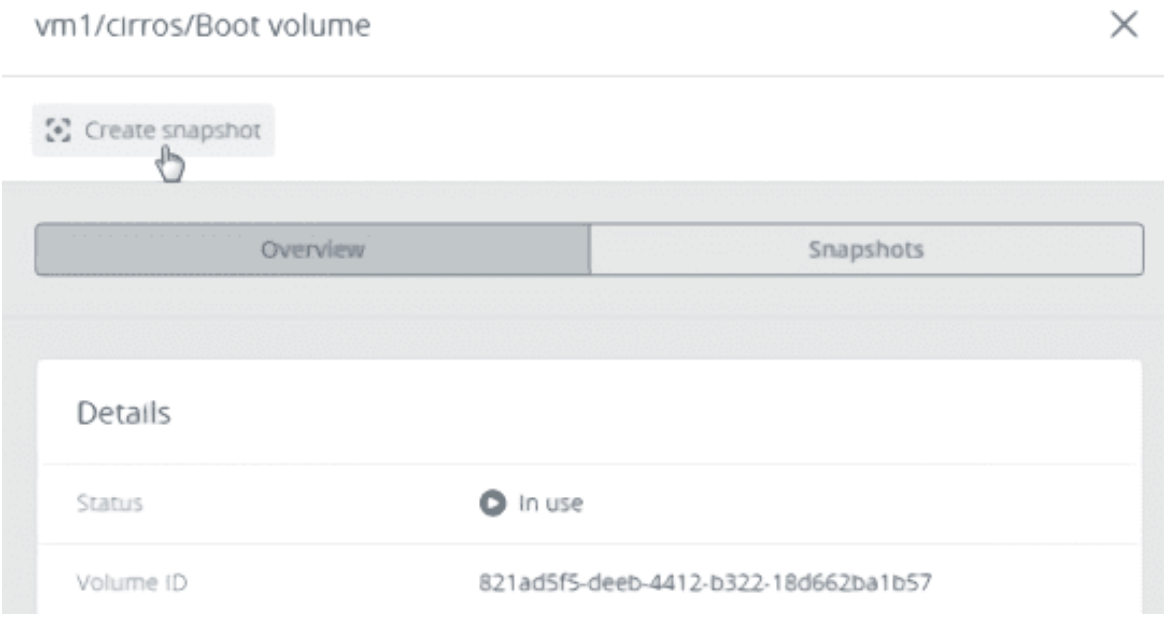
You can save the current state of a VM file system or user data by creating a snapshot of a volume. A snapshot of a boot volume may be useful, for example, before updating VM software. If anything goes wrong, you will be able to revert the VM to a working state at any time. A snapshot of a data volume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

Prerequisites:

- To create a consistent snapshot of a running VM's volume, the guest tools must be installed in the VM. The QEMU guest agent included in the guest tools image automatically quiesces the filesystem during snapshotting.

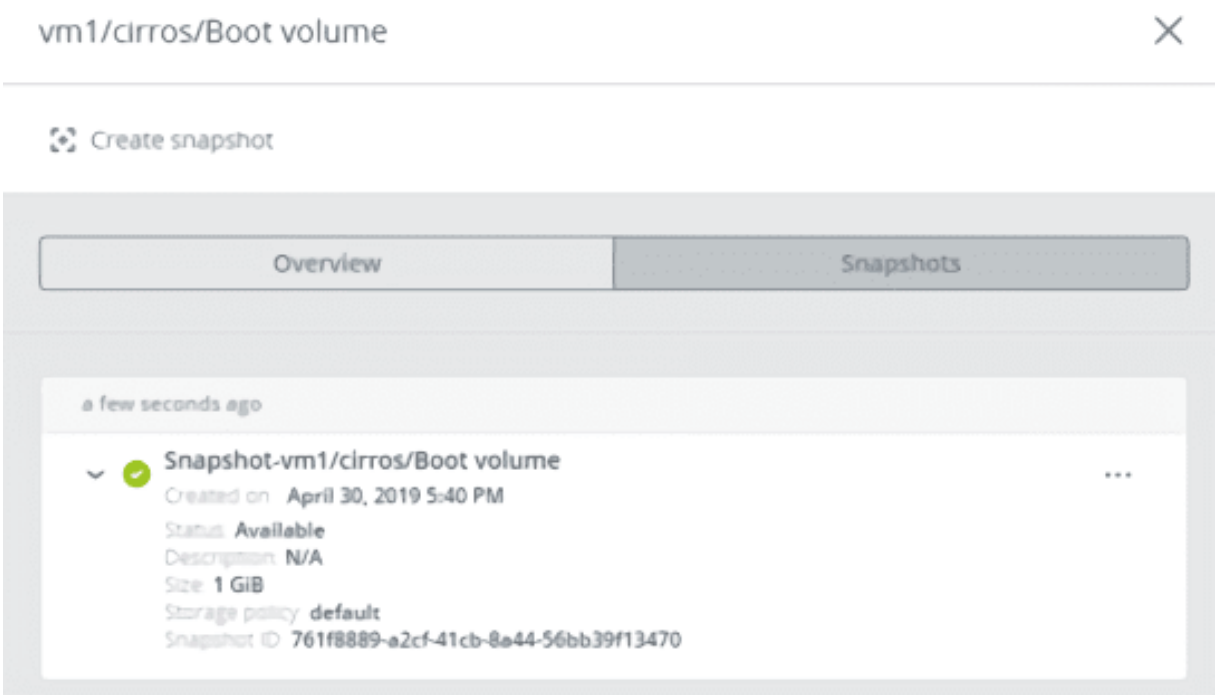
8.7.1 Creating Snapshot of Volume

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



8.7.2 Managing Volume Snapshot

Select a volume and open the **Snapshots** tab on its 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.

Note: 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.

8.8 Transferring Volumes Between Projects

There is no direct way to migrate a virtual machine between different projects. However, you can transfer the VM boot volume, and then create a new VM from it. You can transfer both boot and non-boot volumes to projects within different domains.

Limitations:

- You can only transfer volumes with the “Available” status.
- Transferring volumes that have snapshots breaks the snapshots.

Prerequisites:

- Access to the compute API depends on your provider’s settings. You need to obtain from your provider the instruction how to connect to the API.
- You have login credentials for the source and destination projects.
- If you want to transfer a boot volume that is attached to a VM, clone this volume first, as described in *Cloning Volumes* on page 69.
- If you want to transfer a non-boot volume that is attached to a VM, detach it first, as described in *Attaching and Detaching Volumes* on page 65.

8.8.1 Transferring Volume Between Two Projects

1. Log in to the source project by changing the environment variables to the project credentials. For example:

```
export OS_PROJECT_DOMAIN_NAME=domain1
export OS_USER_DOMAIN_NAME=domain1
export OS_PROJECT_NAME=project1
export OS_USERNAME=user1
export OS_PASSWORD=password
```

2. List all volumes within your project to find out the ID of the volume you want to transfer:

```
# openstack --insecure volume list
+--------------------------------------+------------------+-----------+------+
| ID                                   | Name             | Status    | Size |
+--------------------------------------+------------------+-----------+------+
| 2c8386fa-331b-4ba8-9e4c-de690969a4c8 | win10/Boot volume | available |   64 |
+--------------------------------------+------------------+-----------+------+
```

3. Create a transfer request by specifying the ID of the chosen volume. For example:

```
# openstack --insecure volume transfer request create c0d4cf0e-48e3-417d-b6fc-f1fb36571c5f
+------------+--------------------------------------+
| Field      | Value                                |
+------------+--------------------------------------+
| auth_key   | 75fcf37d56f40182                     |
| created_at | 2022-04-27T09:00:11.776511           |
| id         | b9b835a3-ed41-489a-9552-483fae33c549 |
| name       | None                                 |
| volume_id  | c0d4cf0e-48e3-417d-b6fc-f1fb36571c5f |
+------------+--------------------------------------+
```

   Save the request id and auth-key from the command output, to accept the transfer in the other project.

4. Log in to the destination project by changing the environment variables to the project credentials. For example:

```
export OS_PROJECT_DOMAIN_NAME=domain1
export OS_USER_DOMAIN_NAME=domain1
export OS_PROJECT_NAME=project2
export OS_USERNAME=user2
export OS_PASSWORD=password
```

5. Accept the transfer request by specifying the request ID and authorization key. For example:

```
# openstack --insecure volume transfer request accept --auth-key 75fcf37d56f40182 \
b9b835a3-ed41-489a-9552-483fae33c549
```

Once the volume is moved to the other project, you can create a virtual machine from it, as described in *Creating Virtual Machines* on page 12.

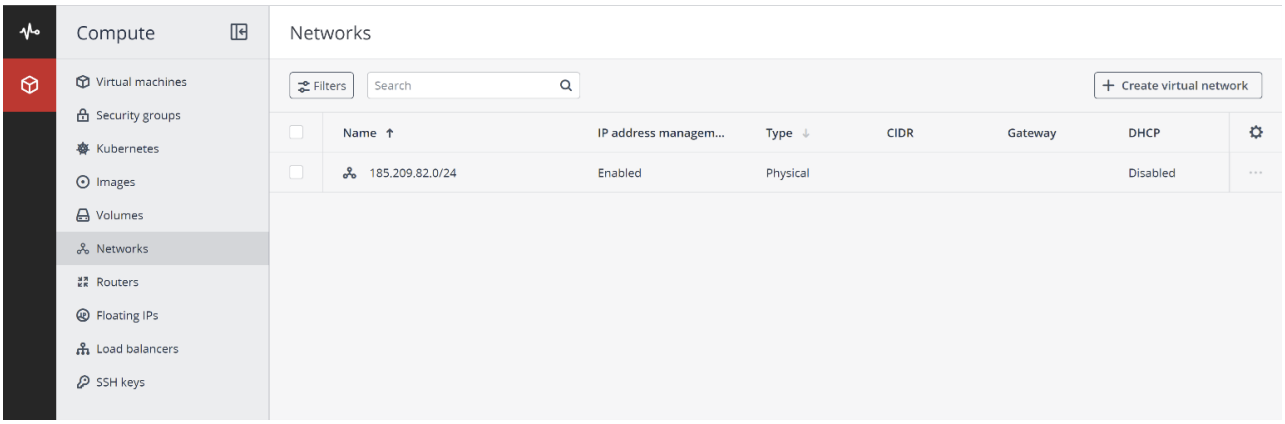
CHAPTER 9

Managing Virtual Private Networks

You can create, edit, and delete a virtual network.

9.1 Creating Virtual Private Network

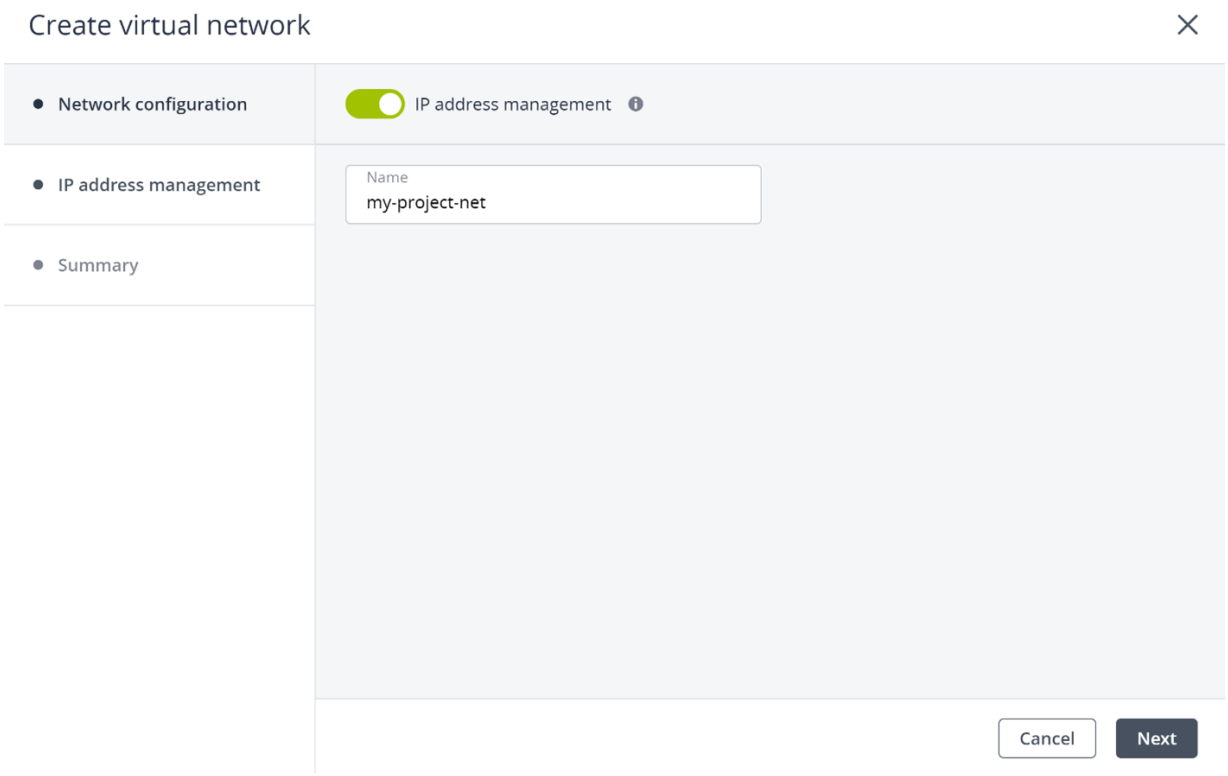
1. On the **Networks** screen, click **+ Create virtual network**.



2. On the **Network configuration** step, do the following:

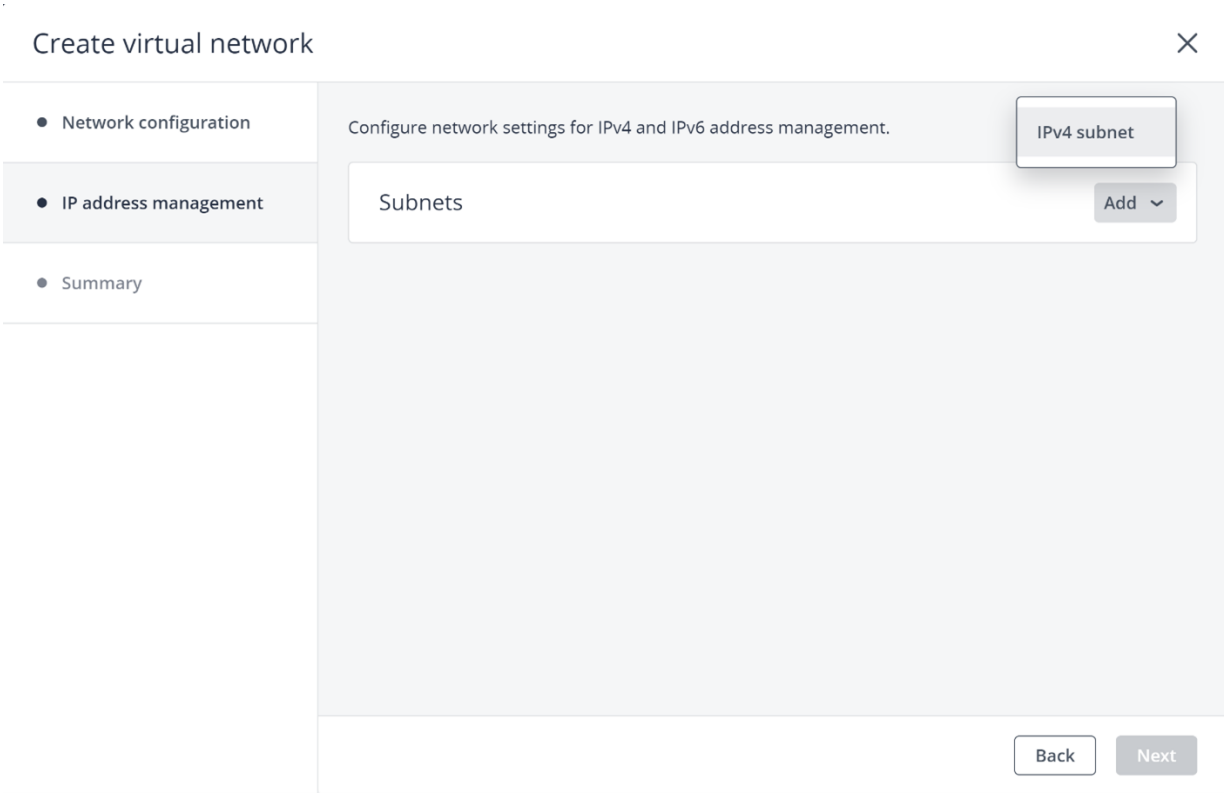
   2.1. Turn on the **IP address management** toggle to provide Internet access for this network.

   2.2. Specify the name of the network and click **Next**.



3. When IP address management is enabled, you will move on to the **IP address management** step:

    3.1. In the **Subnets** section, click **Add** and select **IPv4 subnet**.



3.2. In the **Add IPv4 subnet** window, specify the network's IPv4 address range. The CIDR must be of some private range. For example, 10.100.0.0/24.

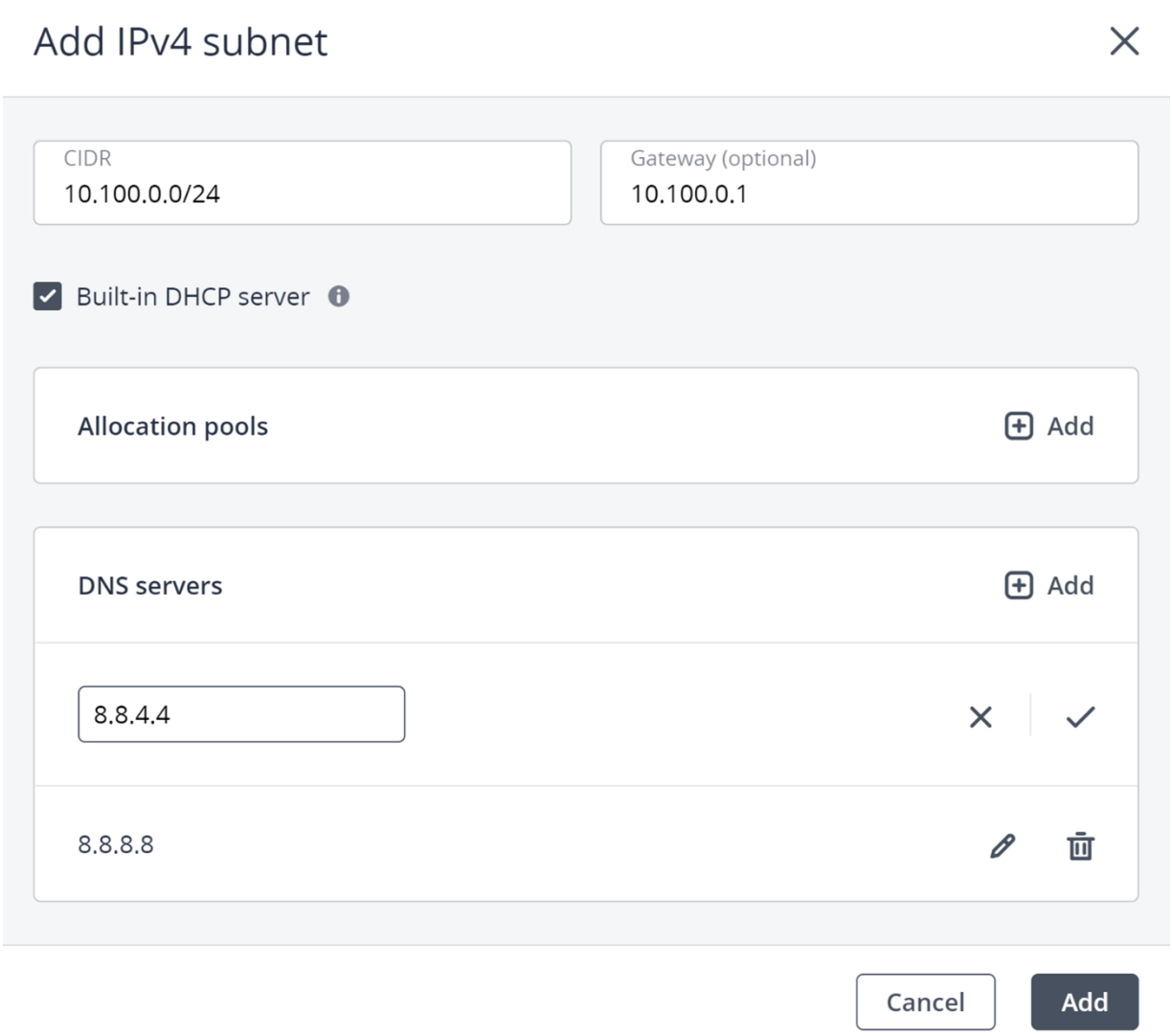
3.3. Specify the gateway, which is a placeholder IP for the virtual router. It can be any IP within the CIDR range—for example, 10.10.0.1.

3.4. Select the **Built-in DHCP server** checkbox to enable delivering the IP for VMs. The DHCP server will take the first two IPs from the allocation pool.

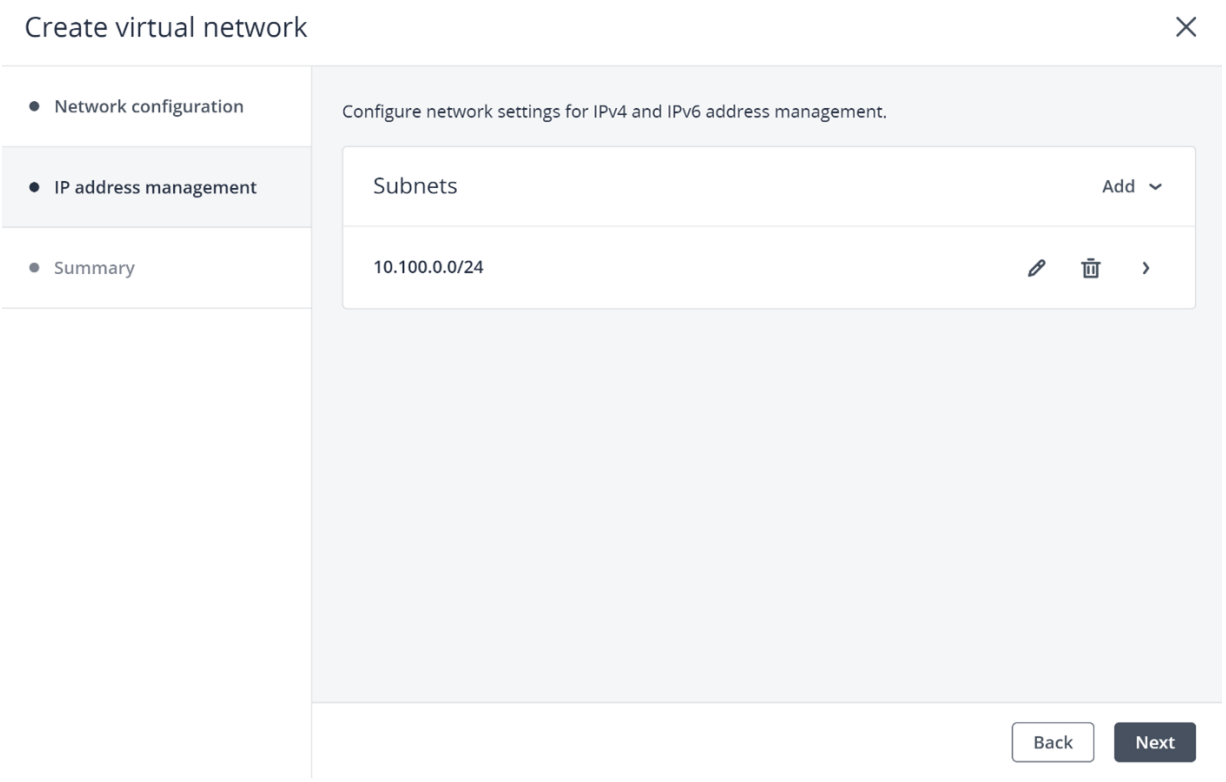
3.5. Specify one or more allocation pools. The allocation pool is an optional configuration to set only needed IP ranges within the CIDR range. When not provided, the allocation pool is equal to the CIDR range. Typically, you do not need to configure it, only if you want to exclude some IPs from being issued by providing narrowed pool range.

3.6. Specify DNS servers. They must point to some existing DNS services.

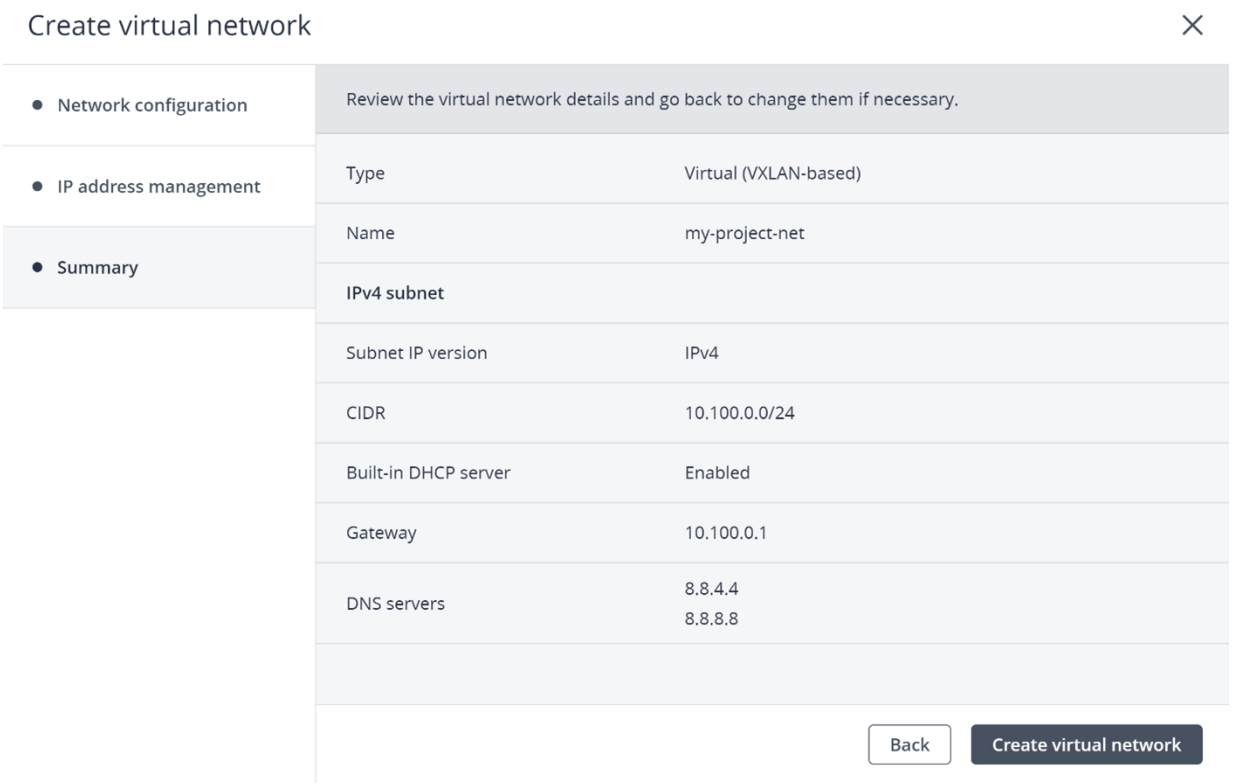
3.7. Click the **Add** button.



3.8. Click **Next**.



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



9.2 Editing Parameters of Virtual Network

1. On the **Networks** screen, click the required network.
2. On the network right pane, click the pencil icon next to the network name or IPv4 subnet.
3. Make changes and save them.

9.3 Deleting Compute Network

Click the ellipsis icon next to the required network, and then click **Delete**. To remove multiple compute networks at once, select them, and then click **Delete**.

CHAPTER 10

Preparing Network

Virtuozzo Hybrid Cloud is a compute service with a web management panel. This page describes the typical network configuration needed to connect to the Internet and to expose compute resources to the public.

Each VHC customer gets their resources grouped and isolated in the cloud tenant we call the **project**. Each project has routed access to the public network (by default, **185.209.82.0/24**).

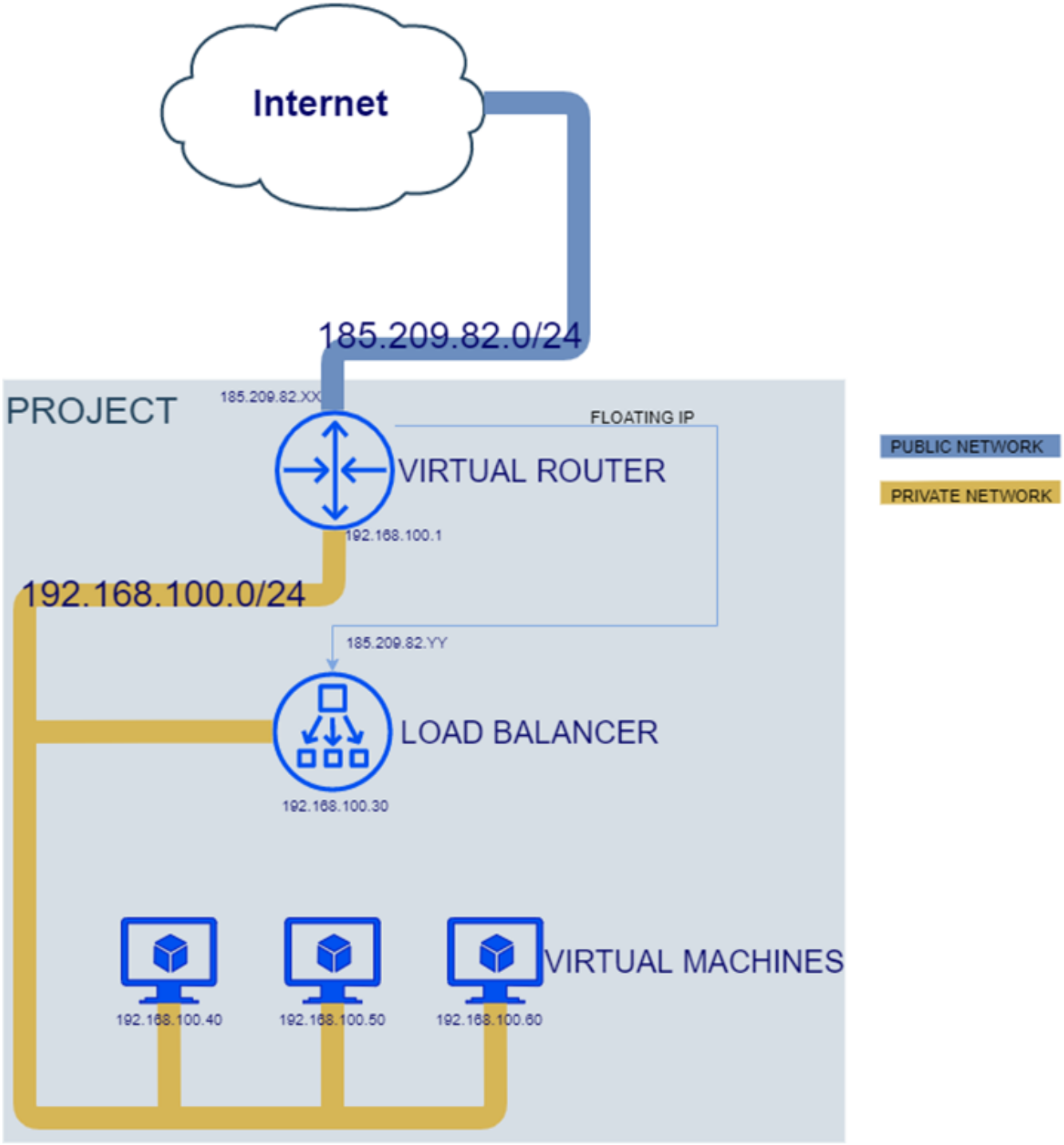
Creating many **virtual private networks** inside every project is possible. Each private network can be a private subnet of any private class: A, B, or C, including subranges of /24 or smaller. For example:

- **10.0.0.0/8 IP addresses**: 10.0.0.0–10.255.255.255
- **172.16.0.0/12 IP addresses**: 172.16.0.0–172.31.255.255
- **192.168.0.0/16 IP addresses**: 192.168.0.0–192.168.255.255

A **virtual router** connects one or multiple private networks to one public network. The virtual router can route traffic between private networks and perform source address translation of private IPs into public to enable internet access for the private networks. In addition, the virtual router can perform destination network address translation to expose a private IP as public.

A **floating IP** is the feature of a virtual router to expose a private IP as a public IP. It binds one virtual machine's private network port to one public network IP.

Exposing multiple private ports via single floating IP with a **Load Balancer** feature is also possible. The load balancer is a particular virtual instance with HAproxy that redirects network traffic to multiple members according to the balancing policy.

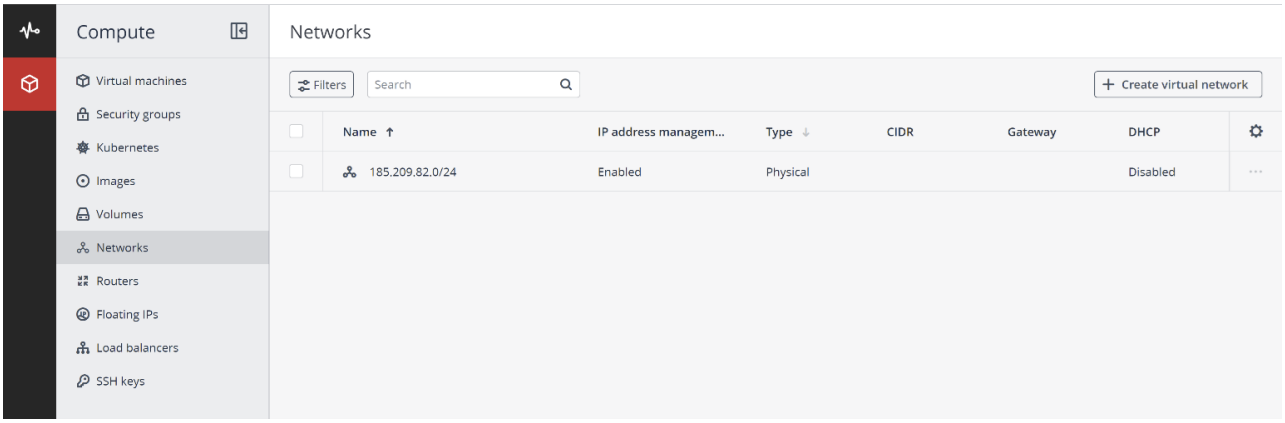

Internet
185.209.82.0/24
PROJECT
185.209.82.XX
FLOATING IP
VIRTUAL ROUTER
192.168.100.1
192.168.100.0/24
185.209.82.YY
LOAD BALANCER
192.168.100.30
VIRTUAL MACHINES
192.168.100.40
192.168.100.50
192.168.100.60
PUBLIC NETWORK
PRIVATE NETWORK

10.1 Preparing Project Networking to Create Virtual Machine with Access to the Internet

A typical project's network must consist of a virtual private network and a virtual router that connects it to the public network. To do it, you must create a virtual private network, virtual router, and virtual machine (refer to *Creating Virtual Machines* on page 12). Then connect your virtual machine to your private network.

10.1.1 Creating Virtual Private Network

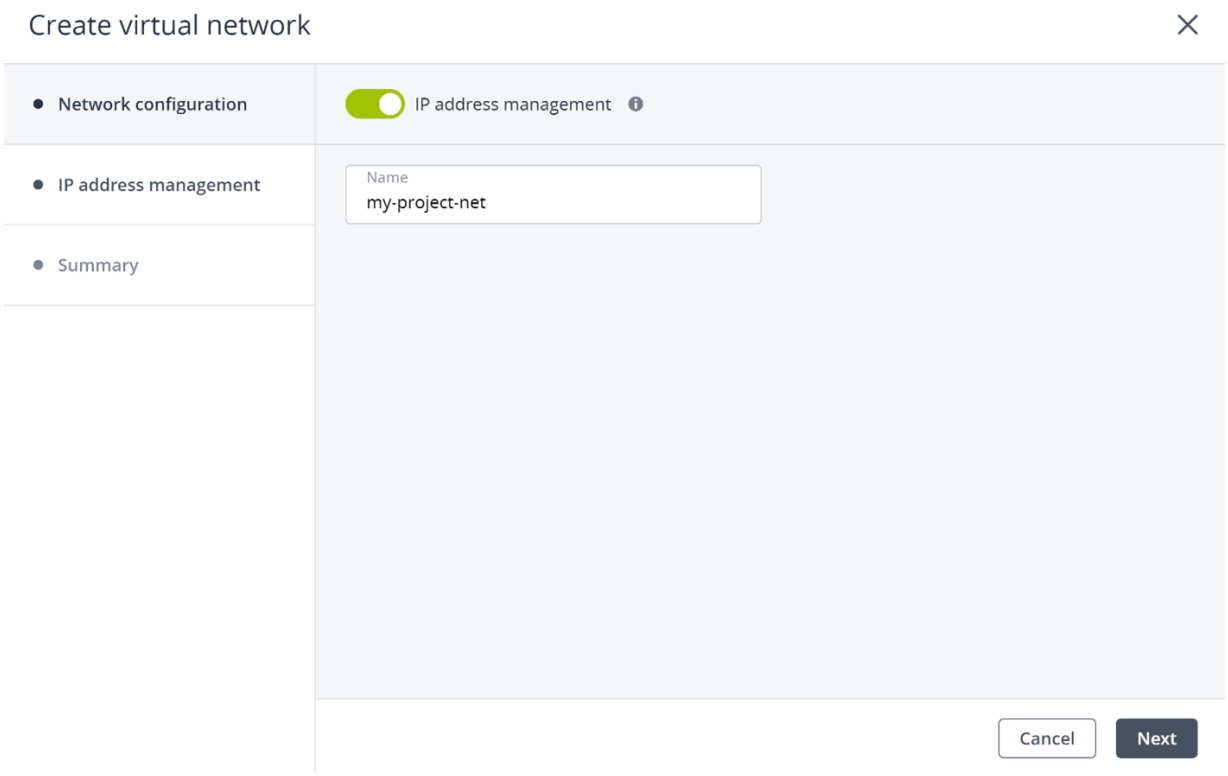
1. On the **Networks** screen, click **+ Create virtual network**.



2. On the **Network configuration** step, do the following:

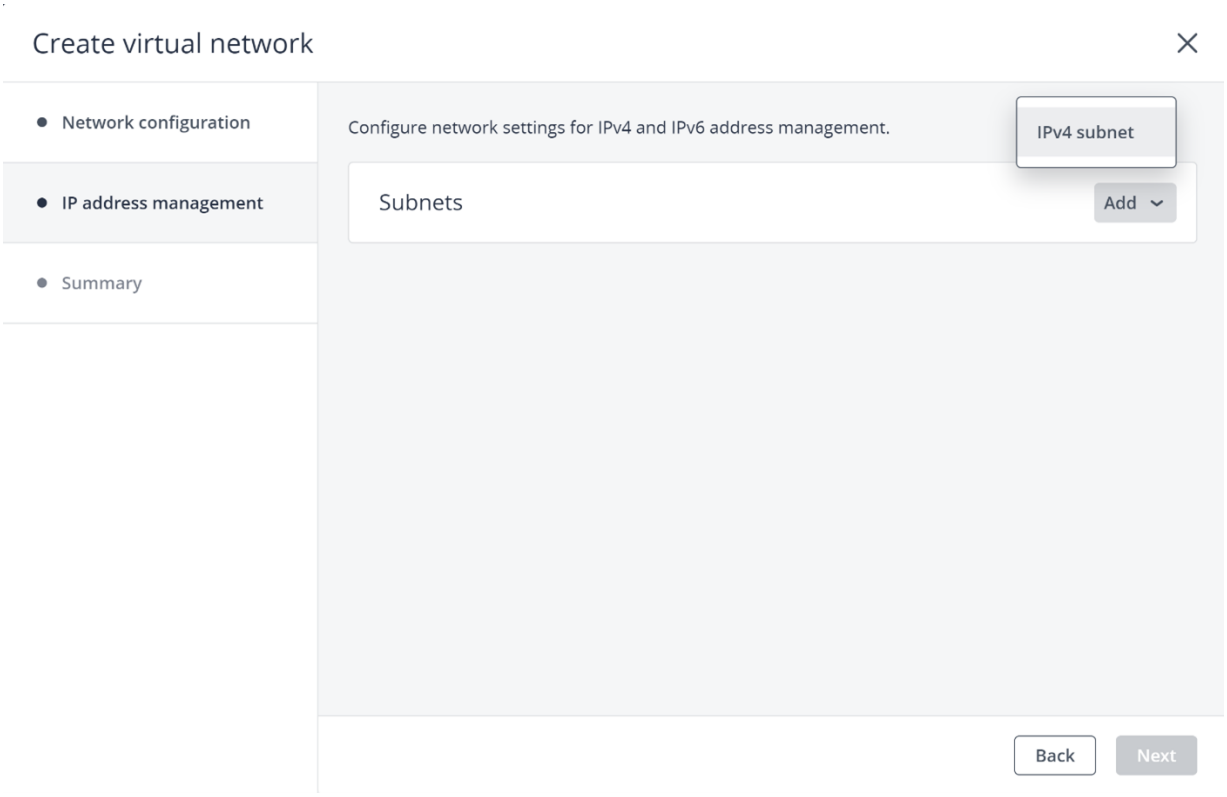
   2.1. Turn on the **IP address management** toggle to provide Internet access for this network.

   2.2. Specify the name of the network and click **Next**.



3. When IP address management is enabled, you will move on to the **IP address management** step:

   3.1. In the **Subnets** section, click **Add** and select **IPv4 subnet**.



3.2. In the **Add IPv4 subnet** window, specify the network's IPv4 address range. The CIDR must be of some private range. For example, 10.100.0.0/24.

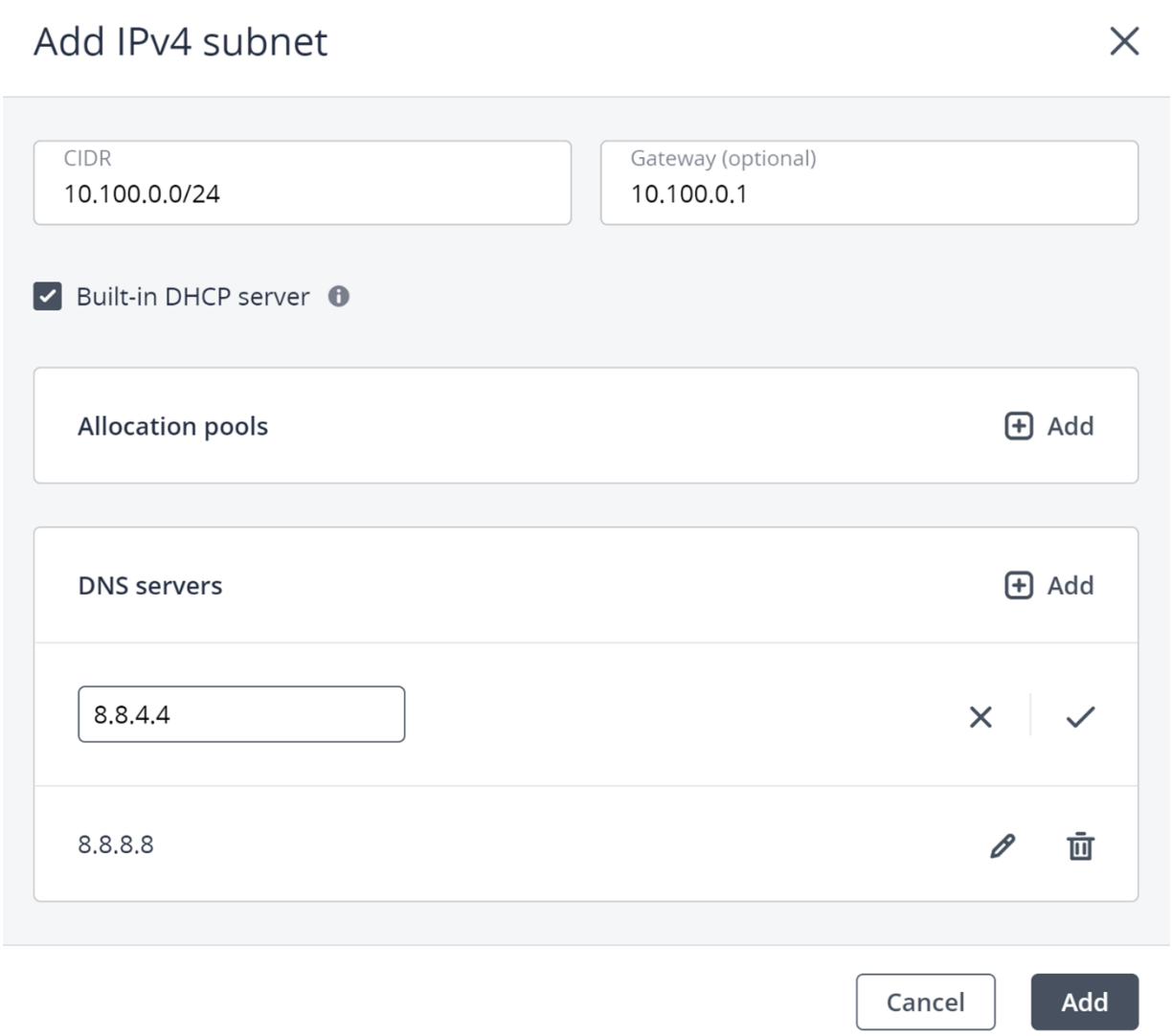
3.3. Specify the gateway, which is a placeholder IP for the virtual router. It can be any IP within the CIDR range—for example, 10.10.0.1.

3.4. Select the **Built-in DHCP server** checkbox to enable delivering the IP for VMs. The DHCP server will take the first two IPs from the allocation pool.

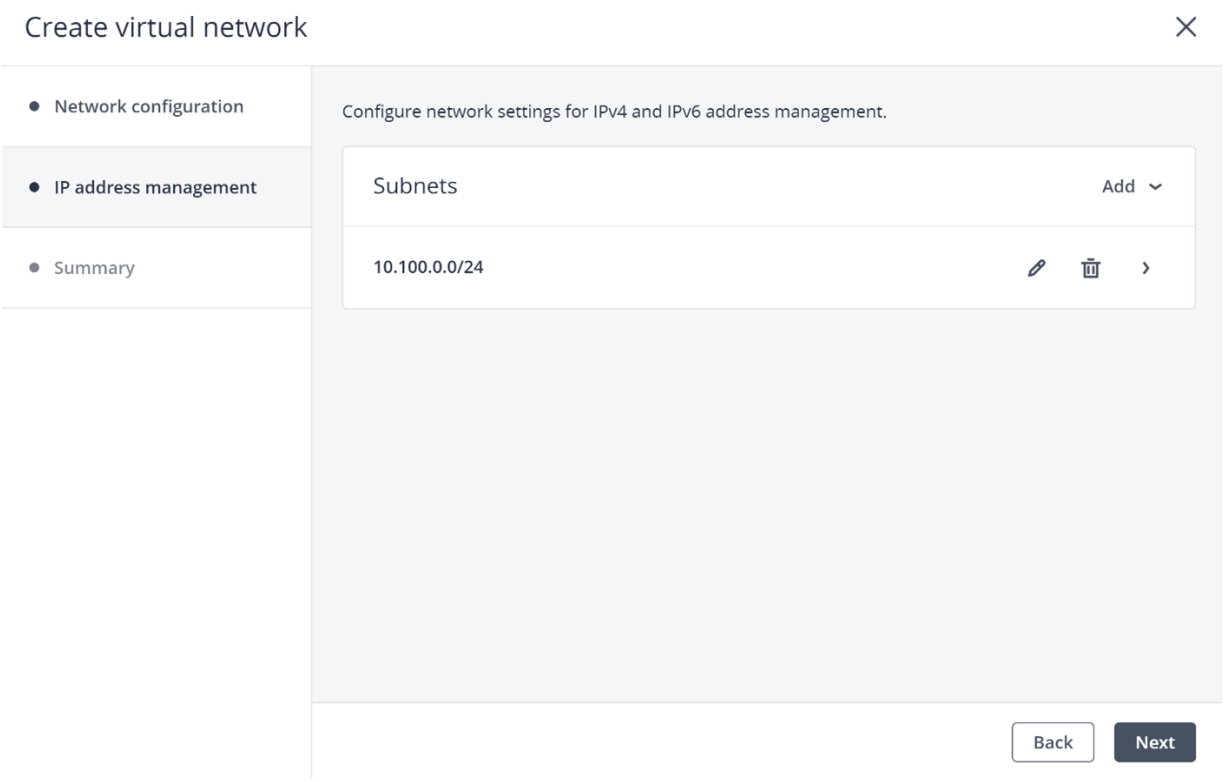
3.5. Specify one or more allocation pools. The allocation pool is an optional configuration to set only needed IP ranges within the CIDR range. When not provided, the allocation pool is equal to the CIDR range. Typically, you do not need to configure it, only if you want to exclude some IPs from being issued by providing narrowed pool range.

3.6. Specify DNS servers. They must point to some existing DNS services.

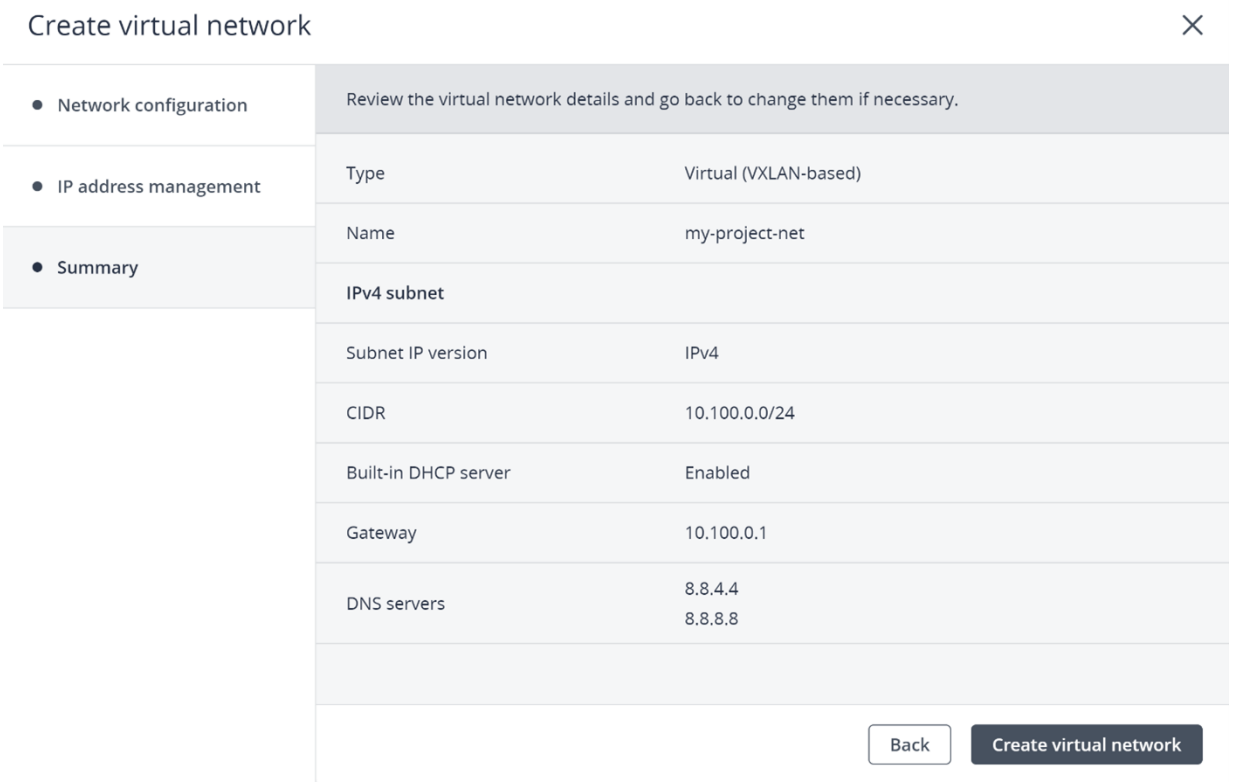
3.7. Click the **Add** button.



3.8. Click **Next**.

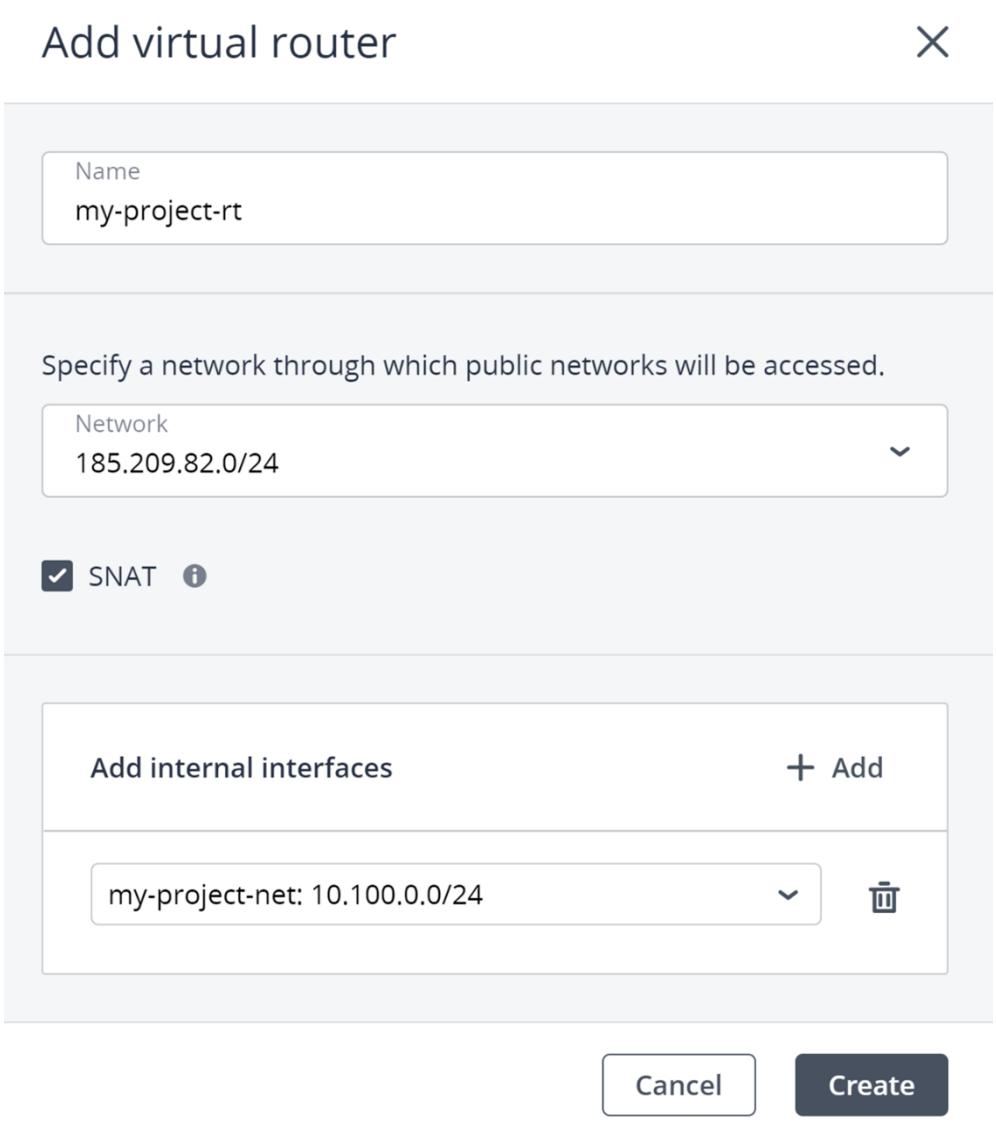


4. On the **Summary** step, review the configuration and click **Create virtual network**.



10.1.2 Creating Virtual Router

1. Navigate to the **Routers** screen and click **+ Add router**.
2. In the **Add virtual router** window:
   2.1. Specify the name of the virtual router.
   2.2. On the **Network** dropdown menu, select an available public network through which public networks will be accessed.
   2.3. Select the **SNAT** checkbox to allow VMs in the private network to communicate with the Internet.
   2.4. In the **Add internal interfaces** section, select the created private network (refer to *Creating Virtual Private Network* on page 75) as an internal interface for the router.
   2.5. Click **Create**.



10.1.3 Connecting Virtual Machine to Private Network

1. On the **Virtual machines** screen, click the required virtual machine.
2. On the **Overview** tab, click **Edit** in the **Network interfaces** section.
3. In the **Network interfaces** window, click **Add** to attach a network interface.
4. In the **Add network interface** window:
   4.1. Select a compute network to connect to. By default, MAC and primary IP addresses are assigned automatically. To specify them manually, clear the **Assign automatically** checkboxes and ensure

that free IP and MAC are selected.

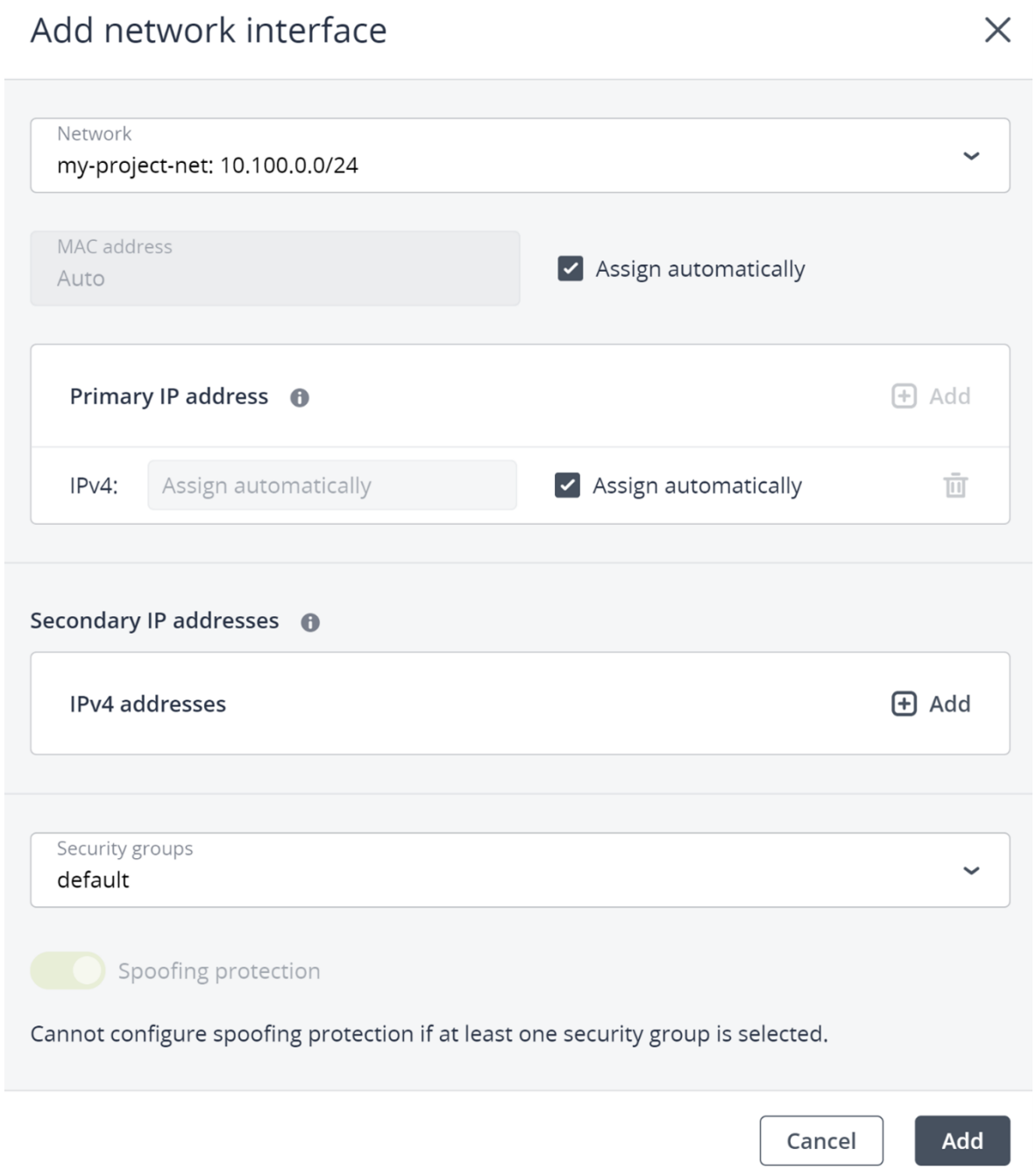
4.2. A secondary IP can be used by applications inside the VM.

Note: Virtuozzo DHCP does not configure additional IP for an interface inside the VMs. Therefore any additional IP must be configured manually inside the VM guest.

4.3. A security group is a port firewall rules list; fault opens all connections. The **default** security group is selected by default. Create other rules on the **Security group** tab if required. Please refer to *Managing Security Group Rules* on page 36.

4.4. Spoofing protection is enabled by default. It is a security feature that blocks outgoing traffic with source MAC and IP addresses that differs from what was defined above. Disabling the spoofing protections also means disabling any firewall control. Do not disable the spoofing protection. If required, deselect the security group.

4.5. Click **Add**.



5. Click **Done** to finish editing VM network interfaces and save your changes.

The resulting VM will be able to reach the Internet, and you can access it via the console from the self-service panel.

10.2 Exposing Virtual Machine to the Internet

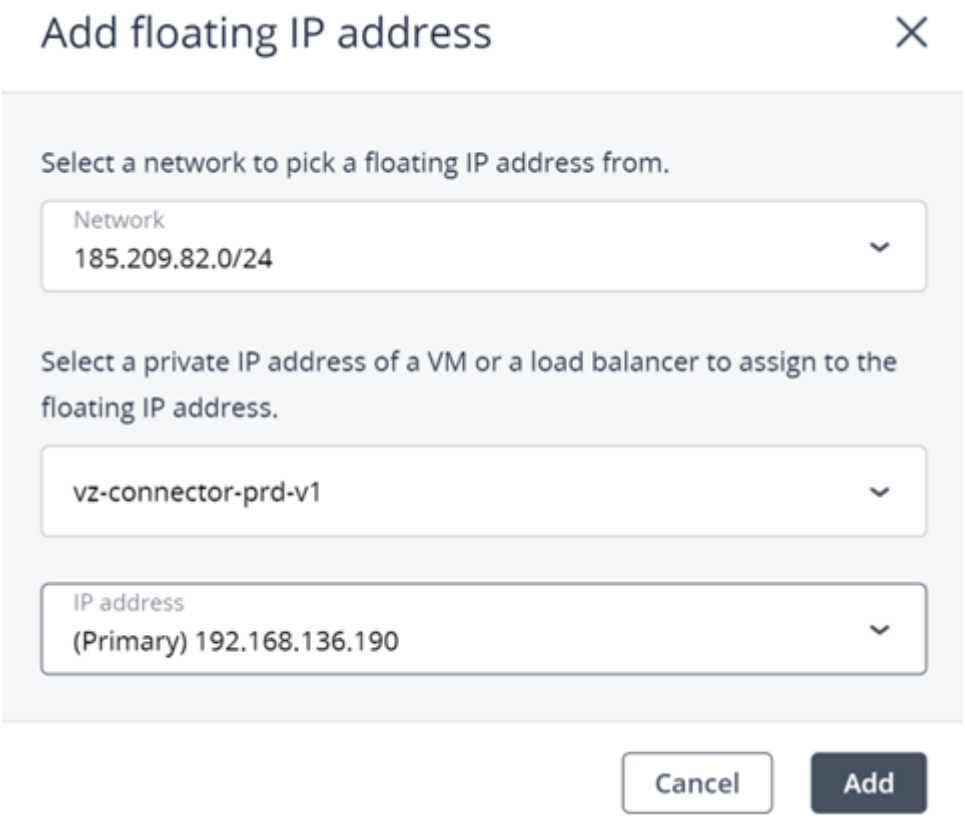
Prerequisites:

- You already have a private network and a router connected to it, and a virtual machine running with a network port.

Floating IP is an IP from the public range assigned to a VM's port in a private network.

To create and assign a floating IP address to a virtual machine:

1. On the **Floating IPs** screen, click **Add floating IP**.
2. In the **Add floating IP address** window, select a network to pick a floating IP from, VM, and port.
3. Click **Add**.



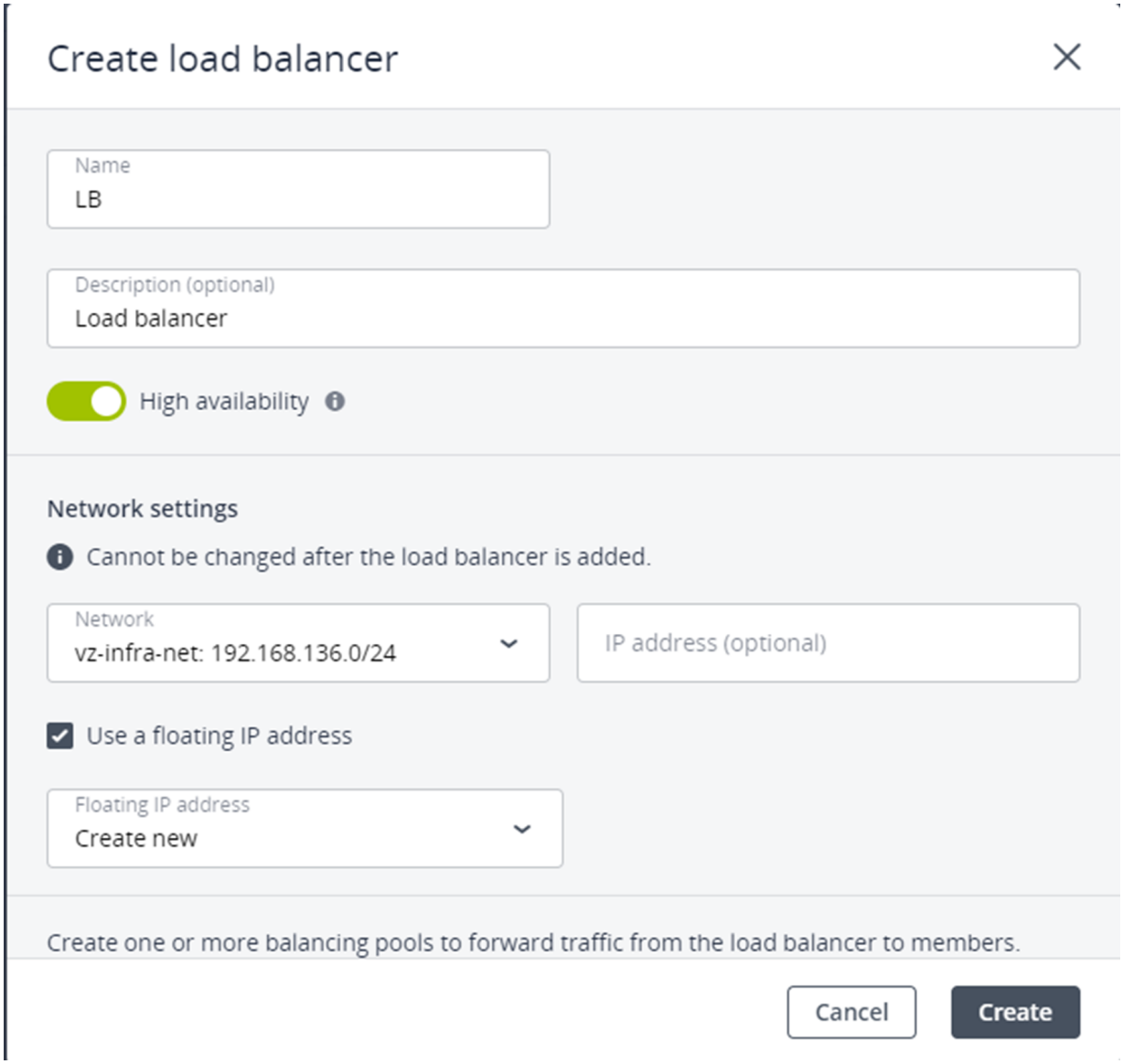
On the **Floating IPs** screen, you will see what public IP you received. This public IP will be bound to the project even if the virtual machine is destroyed. An unassigned floating IP is a floating IP not connected to any VM. It can be used later on by assigning a VM or a load balancer to it, or it can be deleted. In this case, this IP will be lost for the project.

10.3 Using Load Balancer to Expose Service Running on Multiple Virtual Machines

A load balancer is a particular virtual instance with configured HAproxy service that redirects specific traffic to the members' group.

So, if a floating IP is a way to expose one VM to the public, a load balancer is the tool to expose a service running on multiple virtual machines. If you want to expose an HTTP service running on port 8080 in two VMs into one public IP and port 80, do the following:

1. On the **Load balancers** screen, click **+ Create load balancer**.
2. In the **Create load balancer** window, do the following:
    2.1. Specify a name and, optionally, a description.
    2.2. High availability means using two instances of load balancers in the active-backup mode. If high availability is disabled, a single load balancer will be secured with the default platform high availability mode when a VM gets restarted on a new HW node in case of HW failure on the initial node.
3. In the **Network settings** section, select the network in which you have your service's VMs.
    3.1. Select the **Use a floating IP address** checkbox if you need to expose the service to the public, and then choose to use an available floating IP address or create a new one.



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

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

   4.1. In the **Forwarding rule** section:

      4.1.1. Select the protocol which is your service networking protocol, such as HTTP/HTTPS, TCP, or UDP.

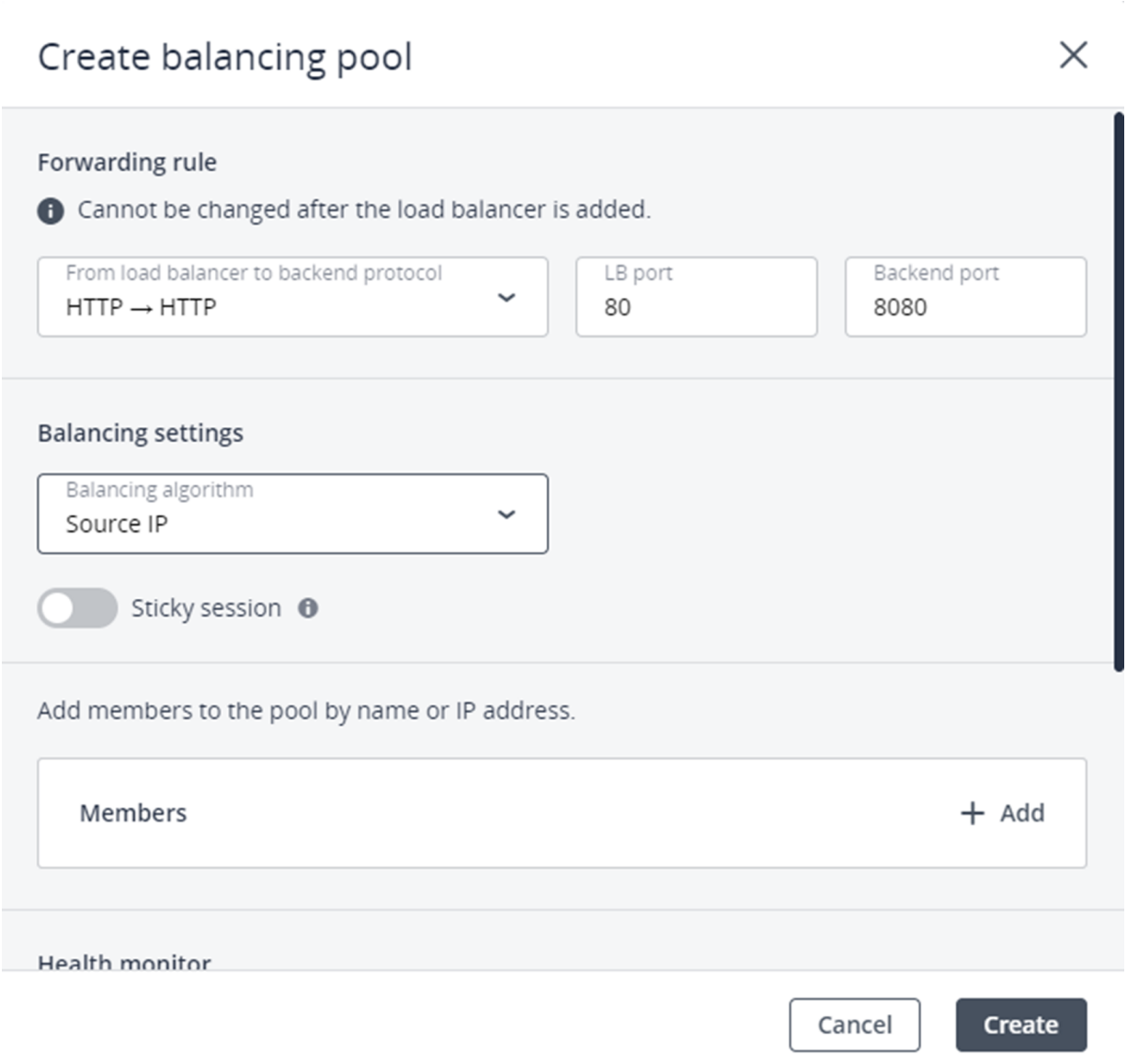
      4.1.2. Specify the LB port a front-facing port that you will use to connect from outside.

      4.1.3. Enter the back-end port, a service port on your virtual machines.

4.2. In the **Balancing settings** section, select the balancing algorithm that determines how data flow will be balanced between the back-end virtual machines:

- **Source IP algorithm**. It will guarantee that an external client (if its IP does not change) will be directed to the same back-end host.
- **Round-robin**. It will direct each packet or session (for session-level protocols) to different back-end hosts.

4.3. Turn on the **Sticky session** toggle to balance the session's level protocols, such as HTTP/HTTPS, to send the packets of the same session to the same back-end host.



5. Click **Create**.

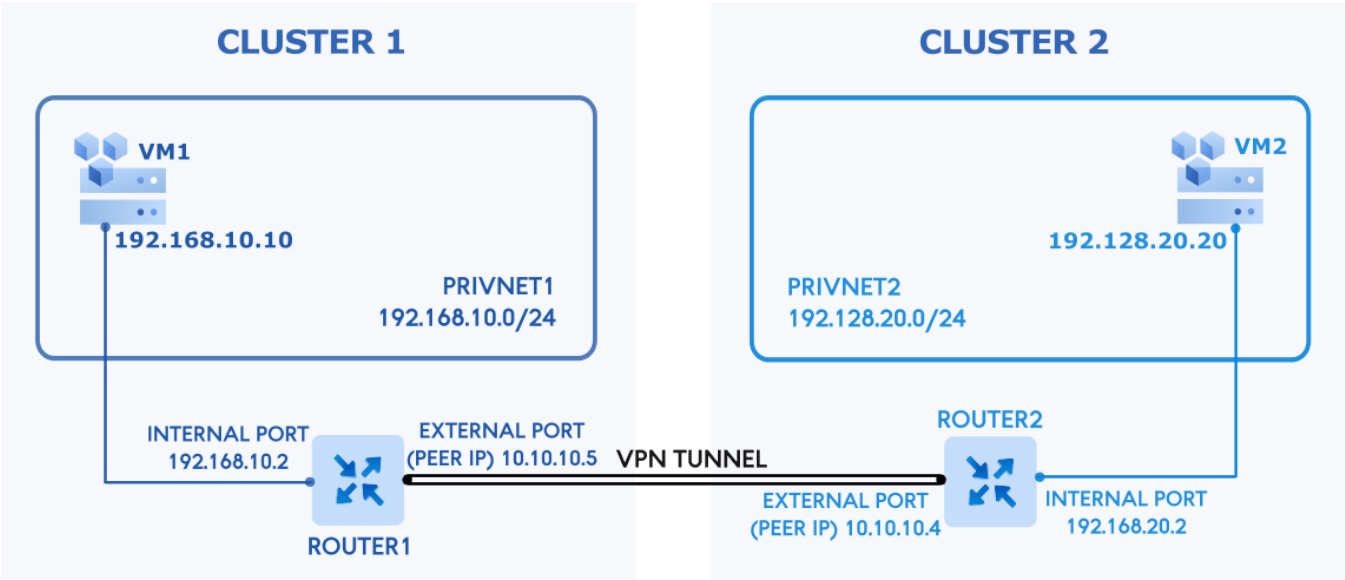
Once created, a load balancer exposes your service to the public.

CHAPTER 11

Managing VPN Connections

With Virtual Private Network (VPN) as a service, self-service users can extend virtual networks across public networks, such as the Internet. To connect two or more remote endpoints, VPNs use virtual connections tunneled through physical networks. To secure VPN communication, the traffic that flows between remote endpoints is encrypted. The VPN implementation uses the Internet Key Exchange (IKE) and IP Security (IPsec) protocols to establish secure VPN connections and is based on the strongSwan IPsec solution.

To better understand how a VPN works, consider the following example:

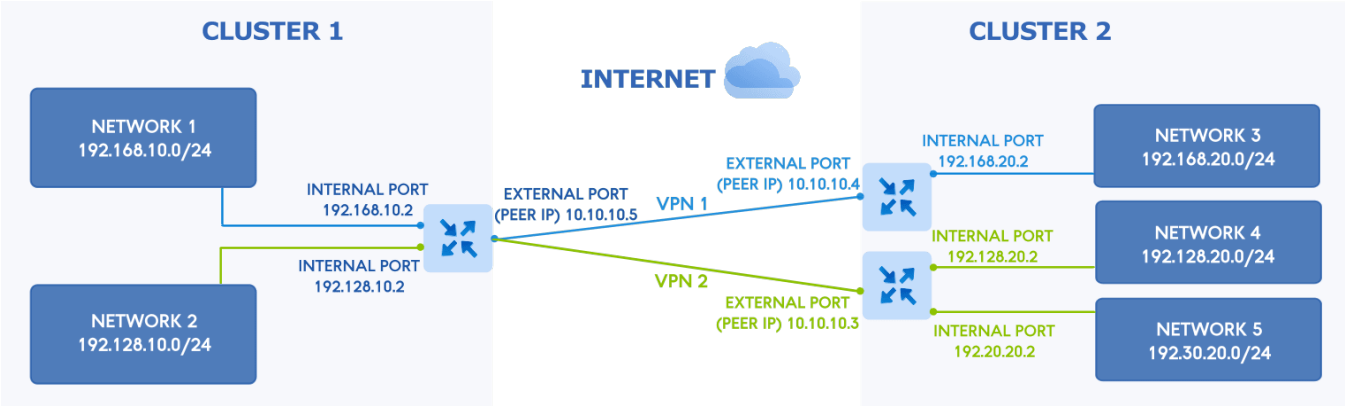


- In the **cluster 1**, the virtual machine **VM1** is connected to the virtual network **privnet1** (192.168.10.0/24) via the network interface with IP address 192.168.10.10. The network **privnet1** is exposed to public networks via the router **router1** with the external port 10.10.10.5.
- In the **cluster 2**, the virtual machine **VM2** is connected to the virtual network **privnet2** (192.168.20.0/24) via the network interface with IP address 192.168.20.20. The network **privnet2** is

exposed to public networks via the router **router2** with the external port 10.10.10.4.

- The VPN tunnel is created between the routers **router1** and **router2** that serve as VPN gateways, thus allowing mutual connectivity between the networks **privnet1** and **privnet2**.
- The virtual machines **VM1** and **VM2** are visible to each other at their private IP addresses. That is, **VM1** can access **VM2** at 192.168.20.20, and **VM2** can access **VM1** at 192.168.10.10.

For key exchange between communicating parties, two IKE versions are available: IKE version 1 (IKEv1) and IKE version 2 (IKEv2). IKEv2 is the latest version of the IKE protocol and it supports connecting multiple remote subnets.



In the example above:

- **VPN1** uses the IKEv1 and connects the network **network1** with the **network3**.
- **VPN2** uses the IKEv2 and connects the network **network2** with the two networks **network4** and **network5**.

11.1 Creating VPN Connections

Limitations:

- A virtual machine must have no floating IP addresses assigned to its private network interface. Otherwise, the VM traffic cannot be routed through a VPN tunnel.

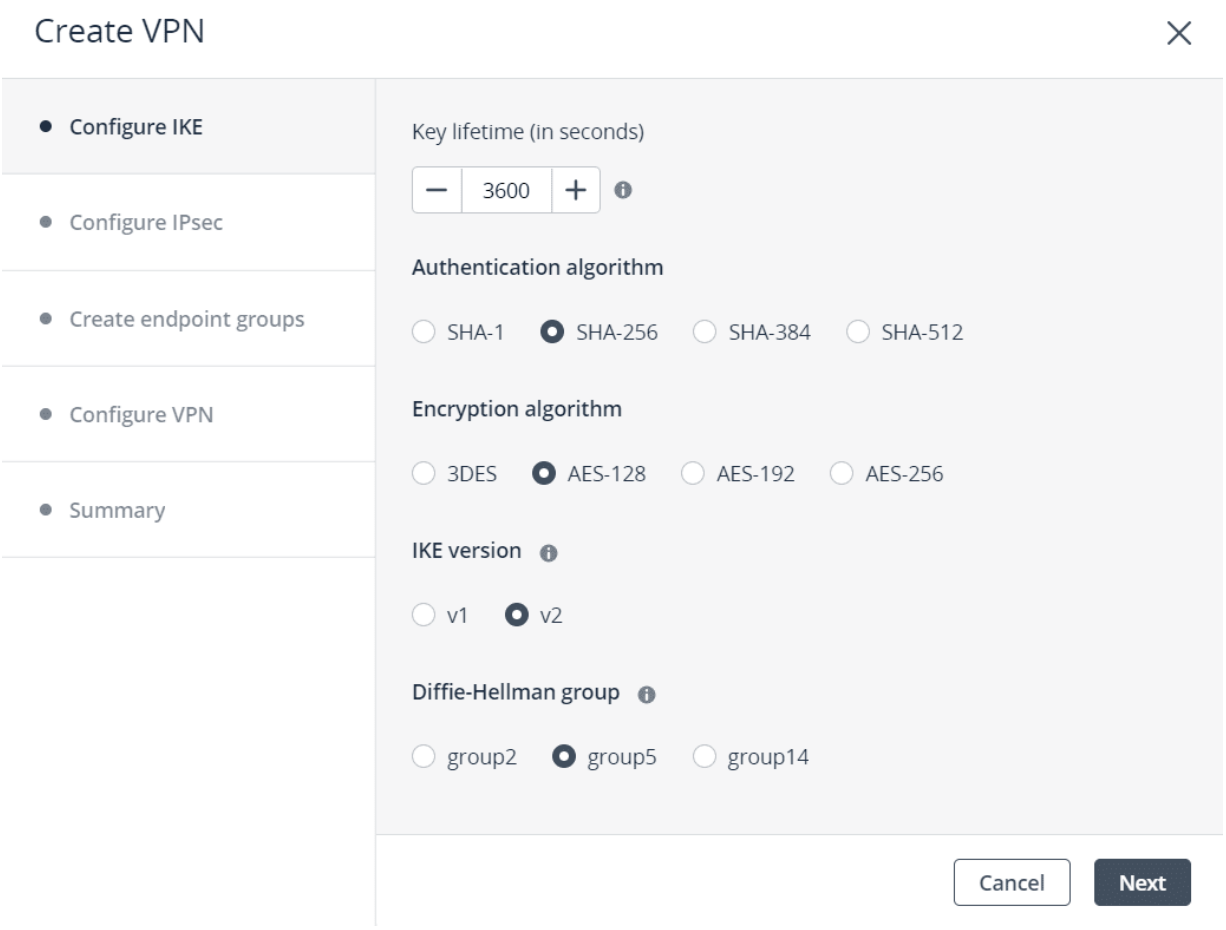
Prerequisites:

- You have a virtual router created, as described in *Managing Virtual Routers* on page 106.
- The virtual router connects the physical network with virtual networks that you want to be exposed.

- Networks that will be connected via a VPN tunnel must have non-overlapping IP ranges.

11.1.1 Creating VPN Connection

1. On the **VPN** screen, click **Create VPN**.
2. On the **Configure IKE** step, specify parameters for the IKE policy that will be used to establish a VPN connection. You can choose to use an existing IKE policy or create a new one. For the new IKE policy, do the following:
   2.1. Specify a custom name for the IKE policy.
   2.2. Specify the key lifetime, in seconds, that will define the rekeying interval. The IKE key lifetime must be greater than that of the IPsec key.
   2.3. Select the authentication algorithm that will be used to verify the data integrity and authenticity.
   2.4. Select the encryption algorithm that will be used to ensure that data is not viewable while in transit.
   2.5. Select the IKE version 1 or 2. Version 1 has limitations, for example, it does not support multiple subnets.
   2.6. Select the Diffie-Hellman (DH) group that will be used to build the encryption key for the key exchange process. Higher group numbers are more secure but require additional time for the key to compute.
   2.7. Click **Next**.



3. On the **Configure IPsec** step, specify parameters for the IPsec policy that will be used to encrypt the VPN traffic. You can choose to use an existing IPsec policy or create a new one. For the new IPsec policy, do the following:

    3.1. Specify a custom name for the IPsec policy.

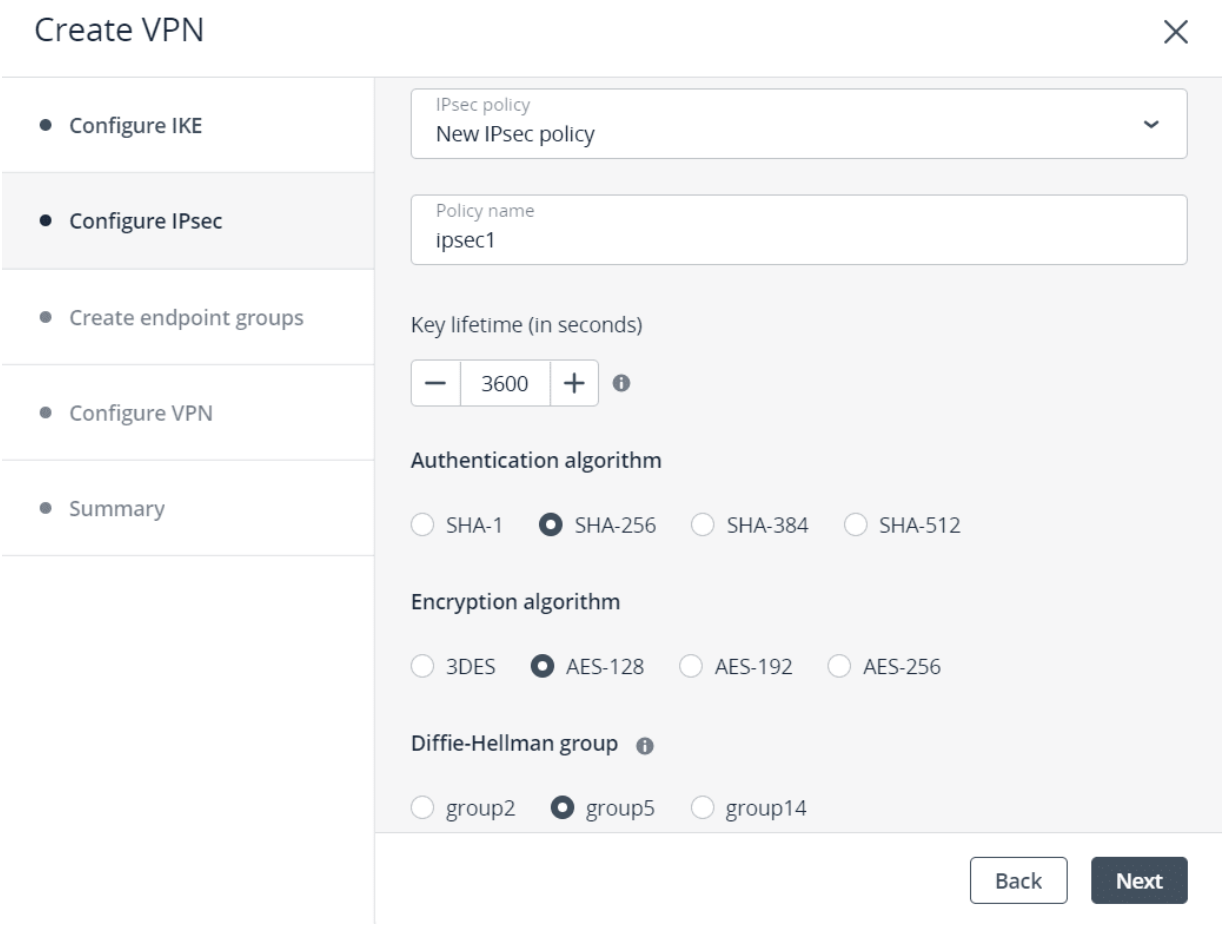
    3.2. Specify the key lifetime, in seconds, that will define the rekeying interval. The IPsec key lifetime must not be greater than that of the IKE key.

    3.3. Select the authentication algorithm that will be used to verify the data integrity and authenticity.

    3.4. Select the encryption algorithm that will be used to ensure that data is not viewable while in transit.

    3.5. Select the Diffie-Hellman (DH) group that will be used to build the encryption key for the key exchange process. Higher group numbers are more secure but require additional time for the key to compute.

    3.6. Click **Next**.

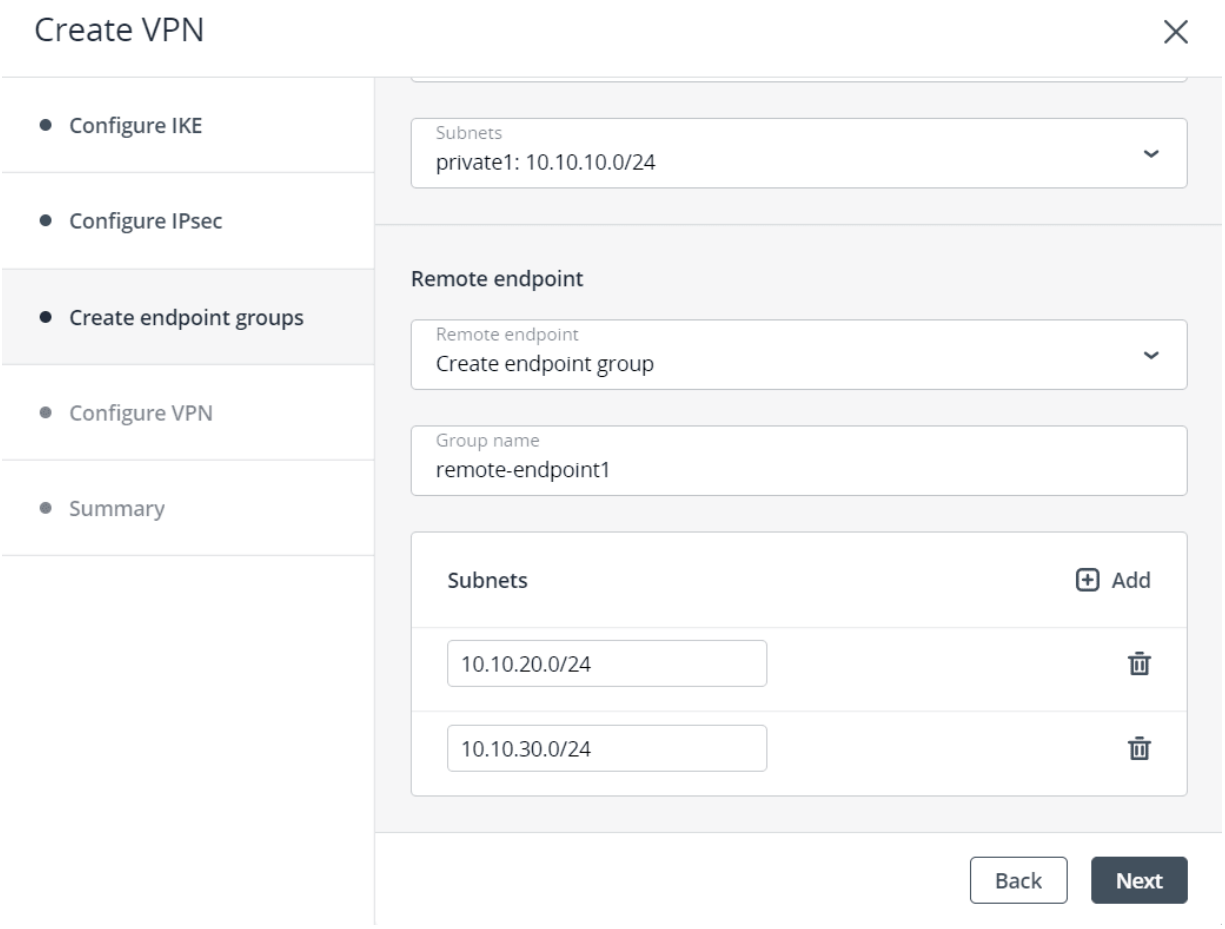


4. On the **Create endpoint groups** step, select a virtual router and specify local and remote subnets that will be connected by the VPN tunnel. You can choose to use existing local and remote endpoints, or create new ones. For the new endpoints, do the following:

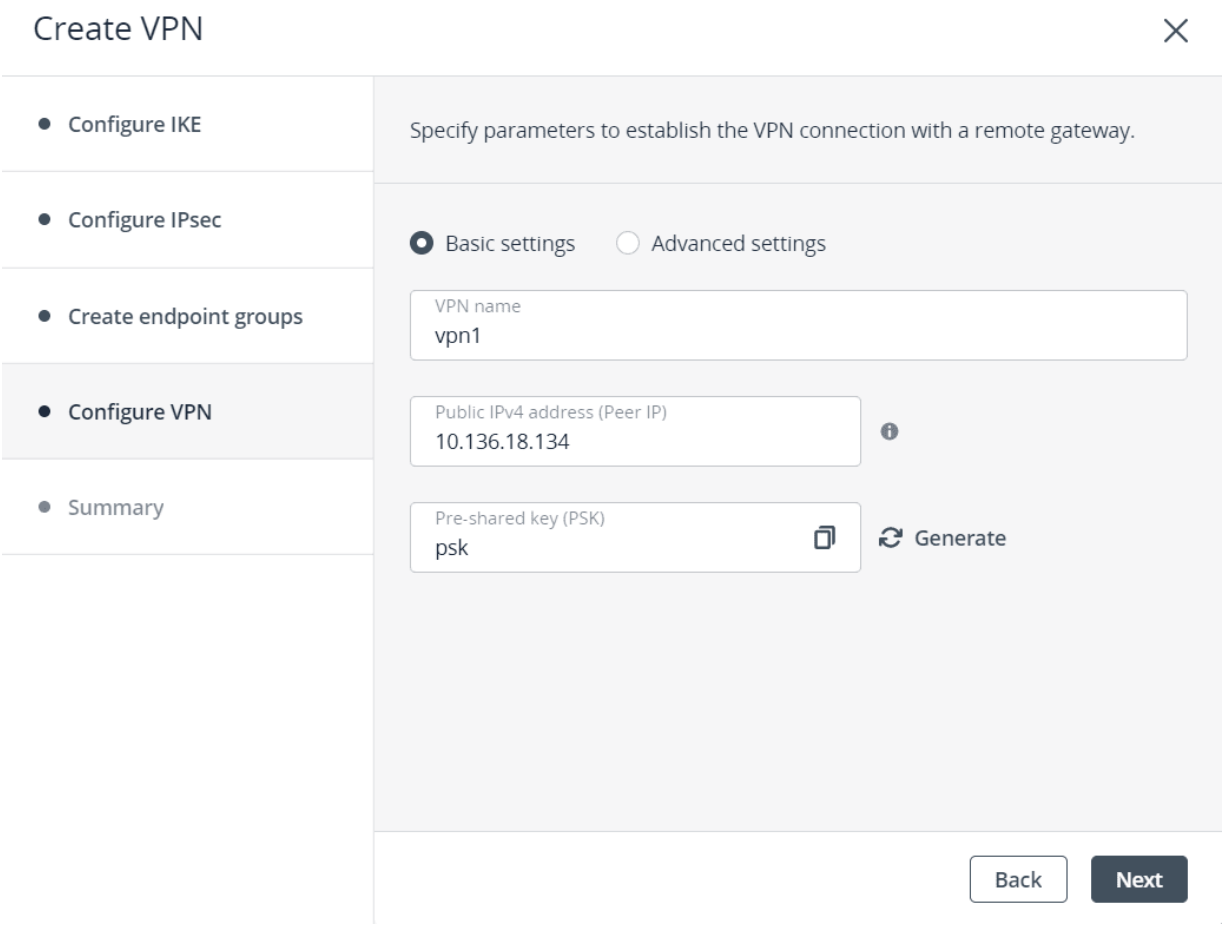
   4.1. Specify a custom name for the local endpoint, and then select local subnets.

   4.2. Specify a custom name for the remote endpoint, and then add remote subnets in the CIDR format.

   4.3. Click **Next**.



5. On the **Configure VPN** step, specify parameters to establish the VPN connection with a remote gateway:
   5.1. Specify a custom name for the VPN connection.
   5.2. Specify the public IPv4 address of the remote gateway, that is, peer IP address.
   5.3. Generate the pre-shared key that will be used for the peer authentication.
   5.4. [Optional] If necessary, you can also configure additional settings by selecting **Advanced settings** and specifying the following parameters:
      - The peer ID for authentication and the mode for establishing a connection.
      - The Dead Peer Detection (DPD) policy, interval, and timeout, in seconds.
   5.5. Click **Next**.



6. On the **Summary** step, review the configuration, and then click **Create**.

When the VPN connection is created, its status will change from “Pending creation” to “Down”. The connection will become active once the VPN tunnel is configured by the other VPN party and the IKE authorization is successful.

Important: The IKE and IPsec configuration must match for both communicating parties. Otherwise, the VPN connection between them will not be established.

11.2 Editing VPN Connections

After a VPN connection is created, you can change its endpoint groups and VPN settings at any time.

Limitations:

- You cannot change the virtual router and security policies used to establish a VPN connection.

Prerequisites:

- A VPN connection is created, as described in *Creating VPN Connections* on page 98.

11.2.1 Edit VPN Connection

1. On the **VPN** screen, click a VPN connection to modify.
2. On the connection right pane, click **Edit**.
3. In the **Edit VPN** window, configure local and remote endpoints, if required, and then click **Next**.
4. On the next step, change VPN parameters such as the VPN connection name, peer IP address, and PSK key. If necessary, you can also configure additional settings by selecting **Advanced settings** and editing the required parameters.
5. Click **Save** to apply your changes.

After you update the connection parameters, its status will change to “Down”. The connection will re-initiate once the parameters are similarly updated by the other VPN party.

Important: The IKE and IPsec configuration must match for both communicating parties. Otherwise, the VPN connection between them will not be established.

11.3 Restarting and Deleting VPN Connections

You can forcefully re-initiate a VPN connection by manually restarting it. When you delete a VPN connection, you also delete the IKE and IPsec policies and endpoint groups that were created during the VPN creation.

Prerequisites:

- A VPN connection is created, as described in *Creating VPN Connections* on page 98.

11.3.1 Restarting VPN Connection

1. On the **VPN** screen, click a VPN connection to restart.
2. On the connection right pane, click **Restart**.
3. Click **Restart VPN** in the confirmation window.

11.3.2 Deleting VPN connection

1. On the **VPN** screen, click a VPN connection to delete.
2. On the connection right pane, click **Delete**.
3. Click **Delete** in the confirmation window.

CHAPTER 12

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

Limitations:

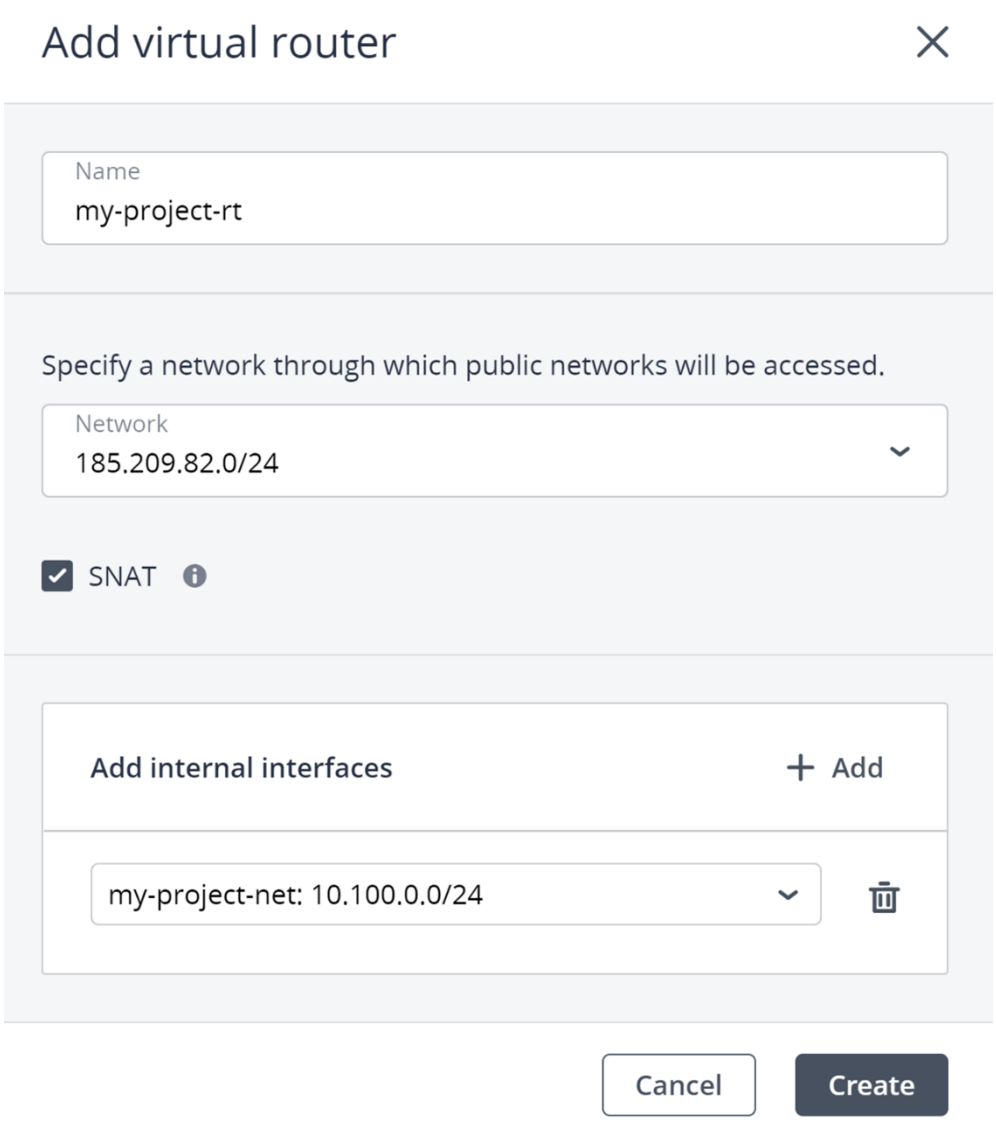
- A router can only connect networks that have IP management enabled.
- You can delete a virtual router if no floating IP addresses are associated with any network it is connected to.

Prerequisites:

- Compute networks are created, as described in *Managing Virtual Private Networks* on page 75.
- The compute networks that are to be connected to a router have a gateway specified.

To create a virtual router:

1. Navigate to the **Routers** screen and click **+ Add router**.
2. In the **Add virtual router** window:
    2.1. Specify the name of the virtual router.
    2.2. On the **Network** dropdown menu, select an available public network through which public networks will be accessed.
    2.3. Select the **SNAT** checkbox to allow VMs in the private network to communicate with the Internet.
    2.4. In the **Add internal interfaces** section, select the created private network (refer to *Creating Virtual Private Network* on page 75) as an internal interface for the router.
    2.5. Click **Create**.



12.1 Managing Router Interfaces

Prerequisites:

- You have a virtual router created, as described in *Managing Virtual Routers* on page 106.

12.1.1 Adding External Router Interface

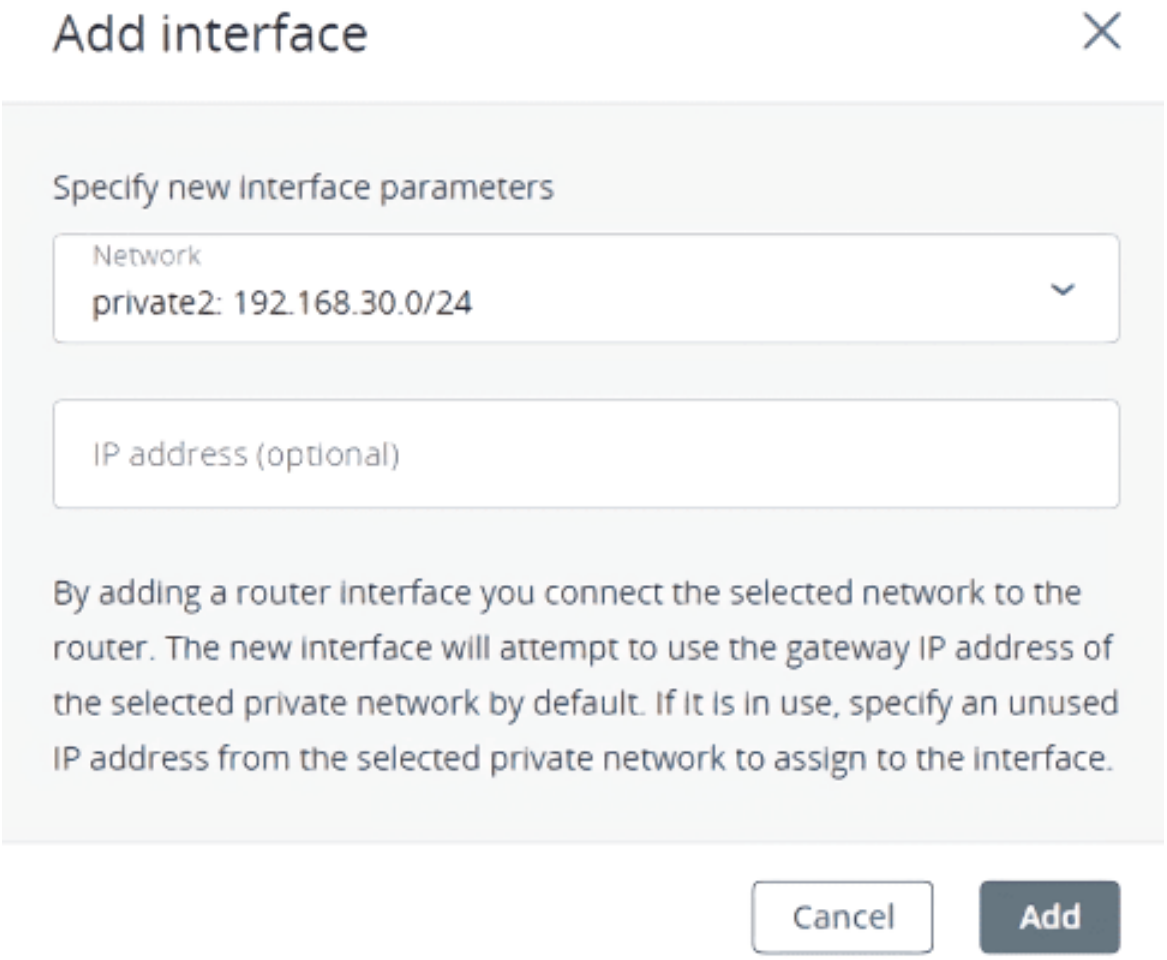
1. If you already have an external gateway, remove the existing one first.
2. On the **Routers** screen, click the router name to open the list of its interfaces.
3. Click **Add** on the toolbar, or click **Add interface** if there are no interfaces to show.
4. In the **Add interface** window, do the following:
   4.1. Select **External gateway**.
   4.2. 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.
   4.3. [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.

5. Click **Add**.

12.1.2 Adding Internal Router Interface

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.



4. Click **Add**.

12.1.3 Editing Router Interface Parameters

1. Click the ellipsis icon next to the interface, and then click **Edit**.
2. In the **Edit interface** window, change the IP address.
3. For an external interface, enable or disable SNAT on it.
4. Click **Save** to save your changes.

12.1.4 Removing Router Interface

1. Select the interface you want to remove.
2. Click the ellipsis icon next to it, and then click **Delete**.

12.2 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.

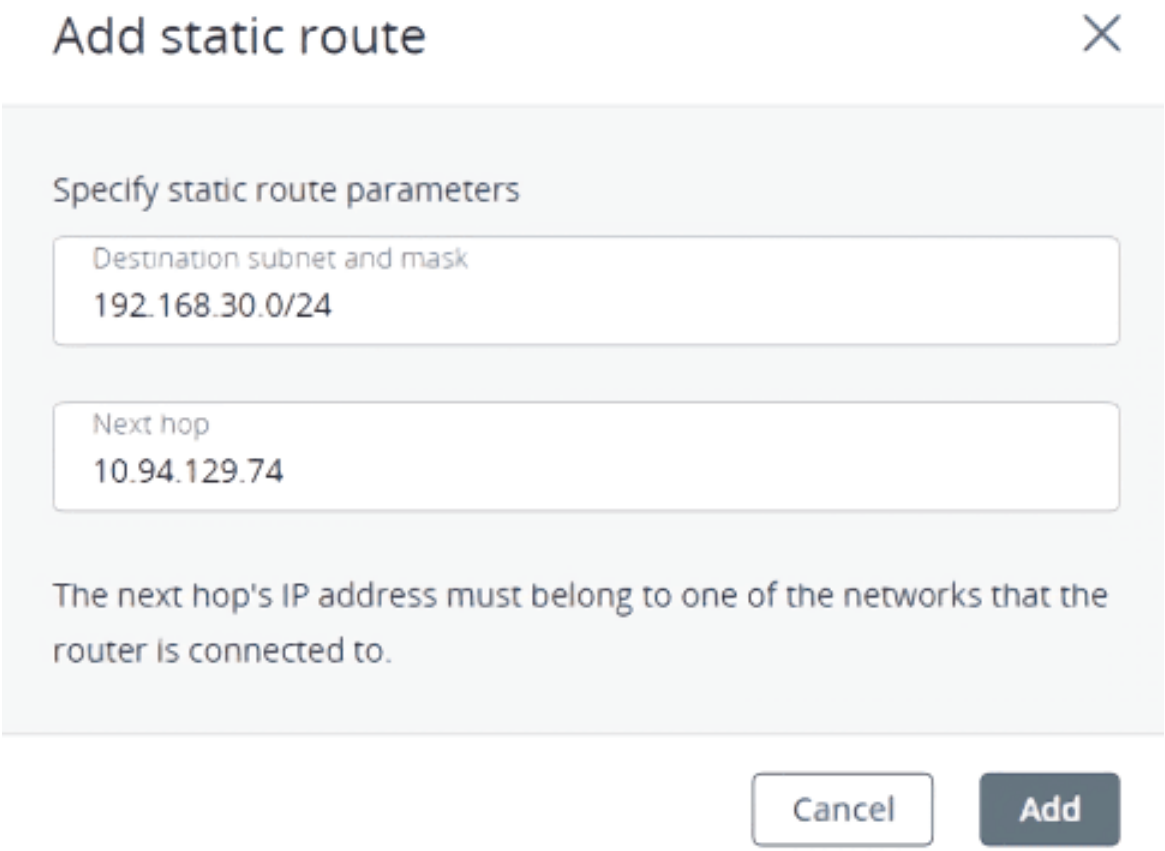
To be able to access **VM2** from **VM1**, you need to add a static route for **router1**, specifying the CIDR of **private2**, that is 192.168.30.0/24, as the destination subnet and the external gateway IP address of **router2**, that is 10.94.129.74, as the next hop IP address. In this case, when an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

Prerequisites:

- You have a virtual router created, as described in *Managing Virtual Routers* on page 106.

12.2.1 Creating Static Route for Router

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.



3. Click **Add**.

12.2.2 Editing Static Route

1. Click the ellipsis icon next to the required static route, and then click **Edit**.
2. In the **Edit static route** window, change the desired parameters, and then click **Save**.

12.2.3 Removing Static Route

Click the ellipsis icon next to the static route you want to remove, and then click **Delete**.

CHAPTER 13

Managing Floating IP Addresses

A volume in Virtuozzo Hybrid Cloud 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.

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.

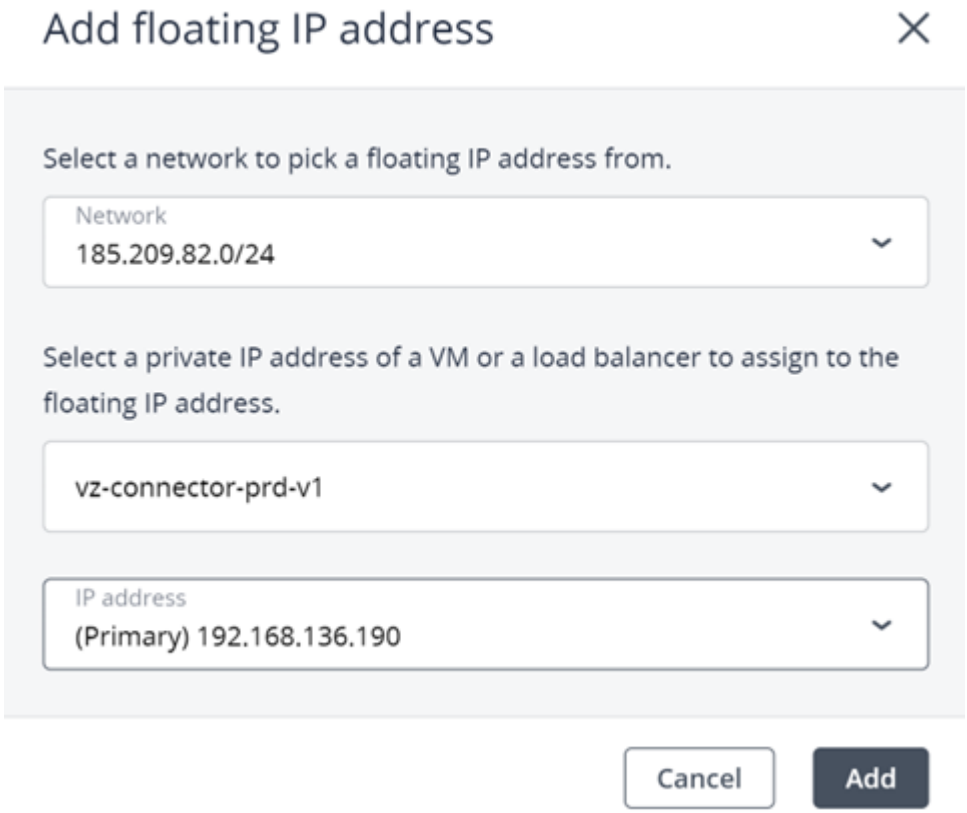
Prerequisites:

- You have a virtual router created, as described in *Managing Virtual Routers* on page 106.
- The virtual machine to assign a floating IP to has a fixed private IP address.
- The virtual router connects the physical network, from which a floating IP will be picked, with the VM's virtual network.

13.1 Creating Floating IP Address and Assigning It to Virtual Machine

1. On the **Floating IPs** screen, click **Add floating IP**.

2. In the **Add floating IP address** window, select a network to pick a floating IP from, VM, and port.
3. Click **Add**.



13.2 Reassigning Floating IP Address to Another Virtual Machine

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**.

13.3 Removing Floating IP Address

1. Unassign it from a virtual machine. Click the ellipsis icon next to the floating IP address, and then click **Unassign**.
2. Click the ellipsis icon again, and then select **Delete**.

CHAPTER 14

Managing Load Balancers

Virtuozzo Hybrid Cloud 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.

14.1 Creating Load Balancers

Limitations:

- The forwarding rule and protocol cannot be changed after the load balancer pool is added.
- If an IPv6 subnet where a load balancer will operate works in the SLAAC or DHCPv6 stateless mode, the load balancer will receive an IPv6 address automatically.

Prerequisites:

- A network where a load balancer will operate has IP management enabled.
- All VMs that will be added in balancing pools have fixed IP addresses.

14.1.1 Creating Load Balancer with Balancing Pools

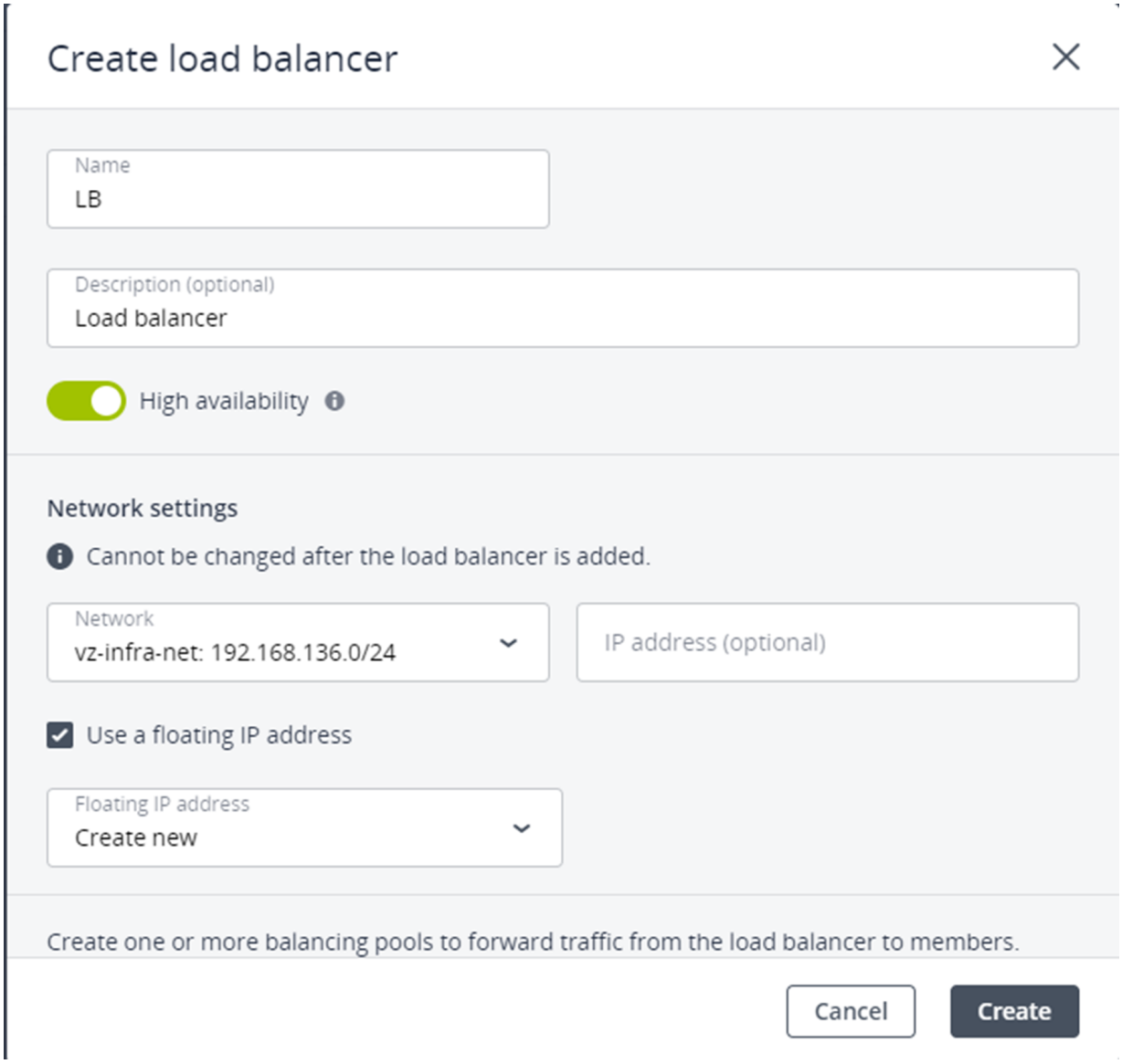
1. On the **Load balancers** screen, click **+ Create load balancer**.
2. In the **Create load balancer** window, do the following:

   2.1. Specify a name and, optionally, a description.

2.2. High availability means using two instances of load balancers in the active-backup mode. If high availability is disabled, a single load balancer will be secured with the default platform high availability mode when a VM gets restarted on a new HW node in case of HW failure on the initial node.

3. In the **Network settings** section, select the network in which you have your service's VMs.

3.1. Select the **Use a floating IP address** checkbox if you need to expose the service to the public, and then choose to use an available floating IP address or create a new one.



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

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

4.1. In the **Forwarding rule** section:

4.1.1. Select the protocol which is your service networking protocol, such as HTTP/HTTPS, TCP, or UDP.

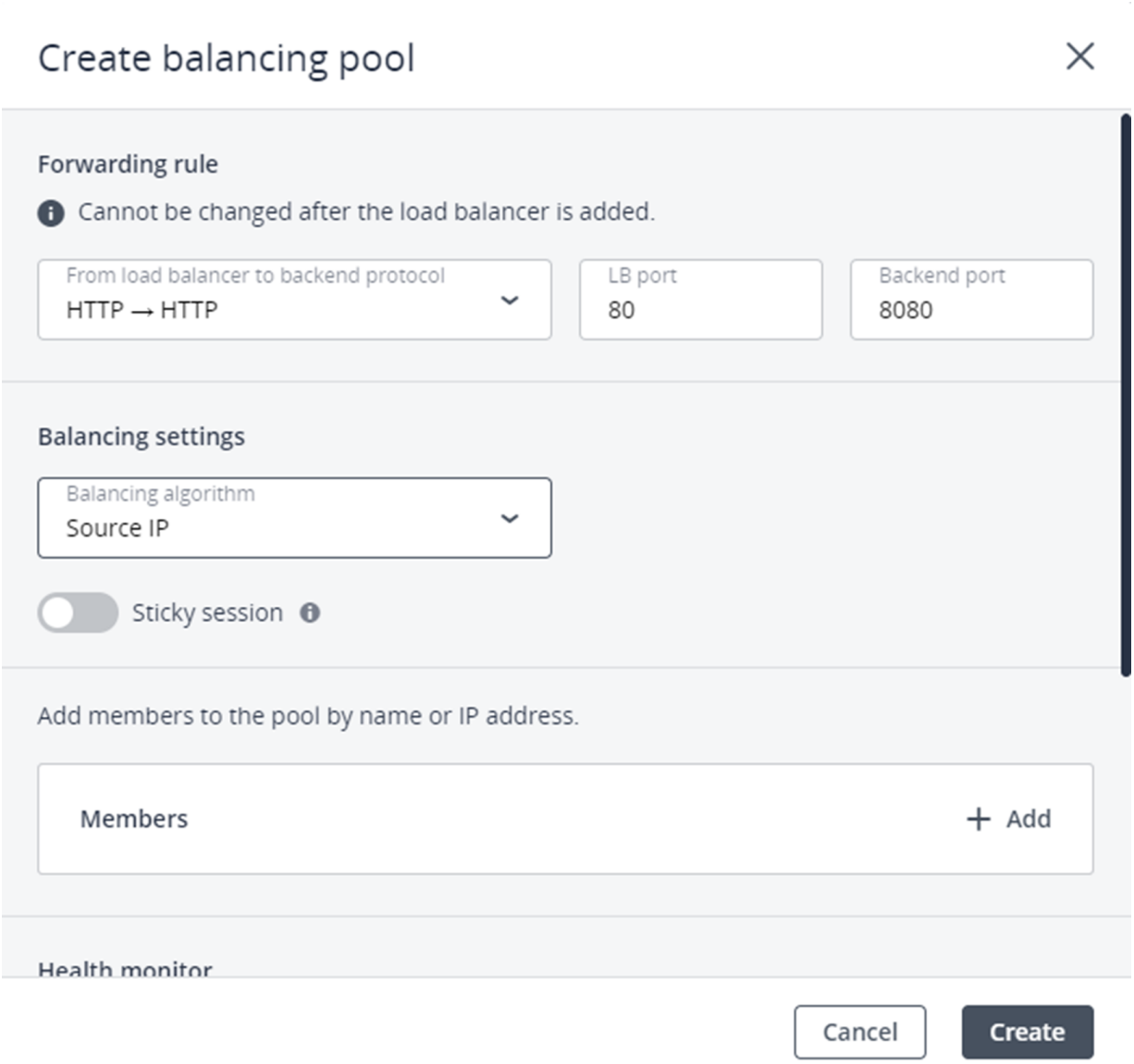
4.1.2. Specify the LB port a front-facing port that you will use to connect from outside.

4.1.3. Enter the back-end port, a service port on your virtual machines.

4.2. In the **Balancing settings** section, select the balancing algorithm that determines how data flow will be balanced between the back-end virtual machines:

- **Source IP algorithm**. It will guarantee that an external client (if its IP does not change) will be directed to the same back-end host.
- **Round-robin**. It will direct each packet or session (for session-level protocols) to different back-end hosts.

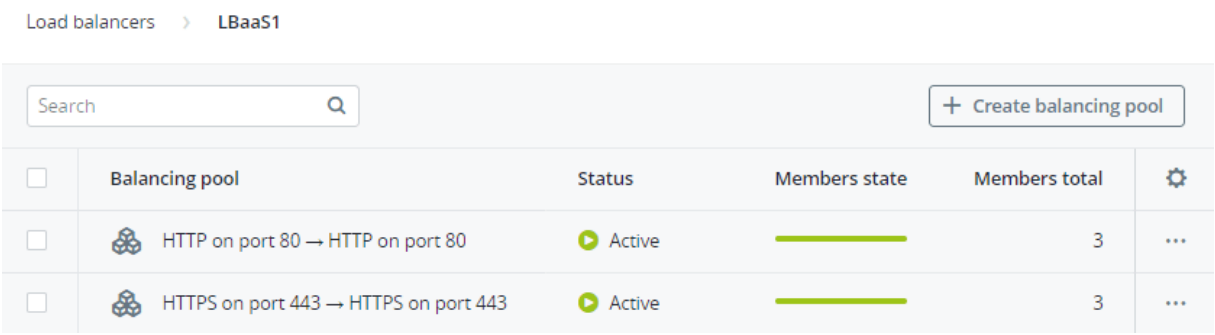
4.3. Turn on the **Sticky session** toggle to balance the session's level protocols, such as HTTP/HTTPS, to send the packets of the same session to the same back-end host.



5. Click **Create**.

14.2 Managing Balancing Pools

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



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.

Limitations:

- The forwarding rule and protocol cannot be changed after the load balancer pool is added.

Prerequisites:

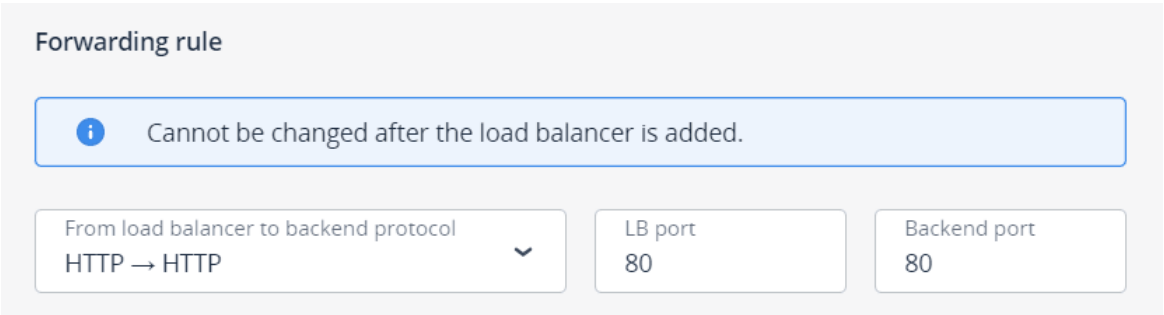
- All VMs that will be added in balancing pools have fixed IP addresses.

14.2.1 Add Another Balancing Pool to Load Balancer

1. Click the load balancer name, and then clikc **Create balancing pool**.
2. 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.

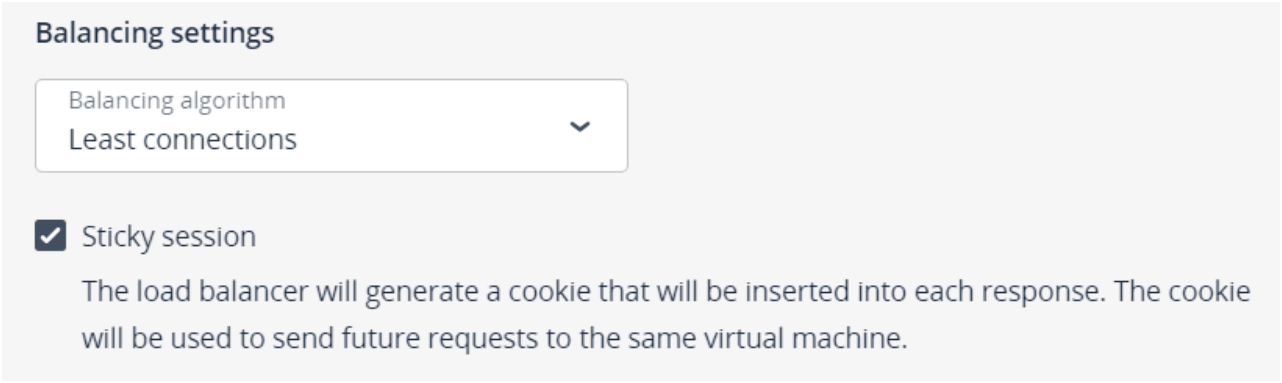


3. In the **Balancing settings** section, select the balancing algorithm:

   - **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.

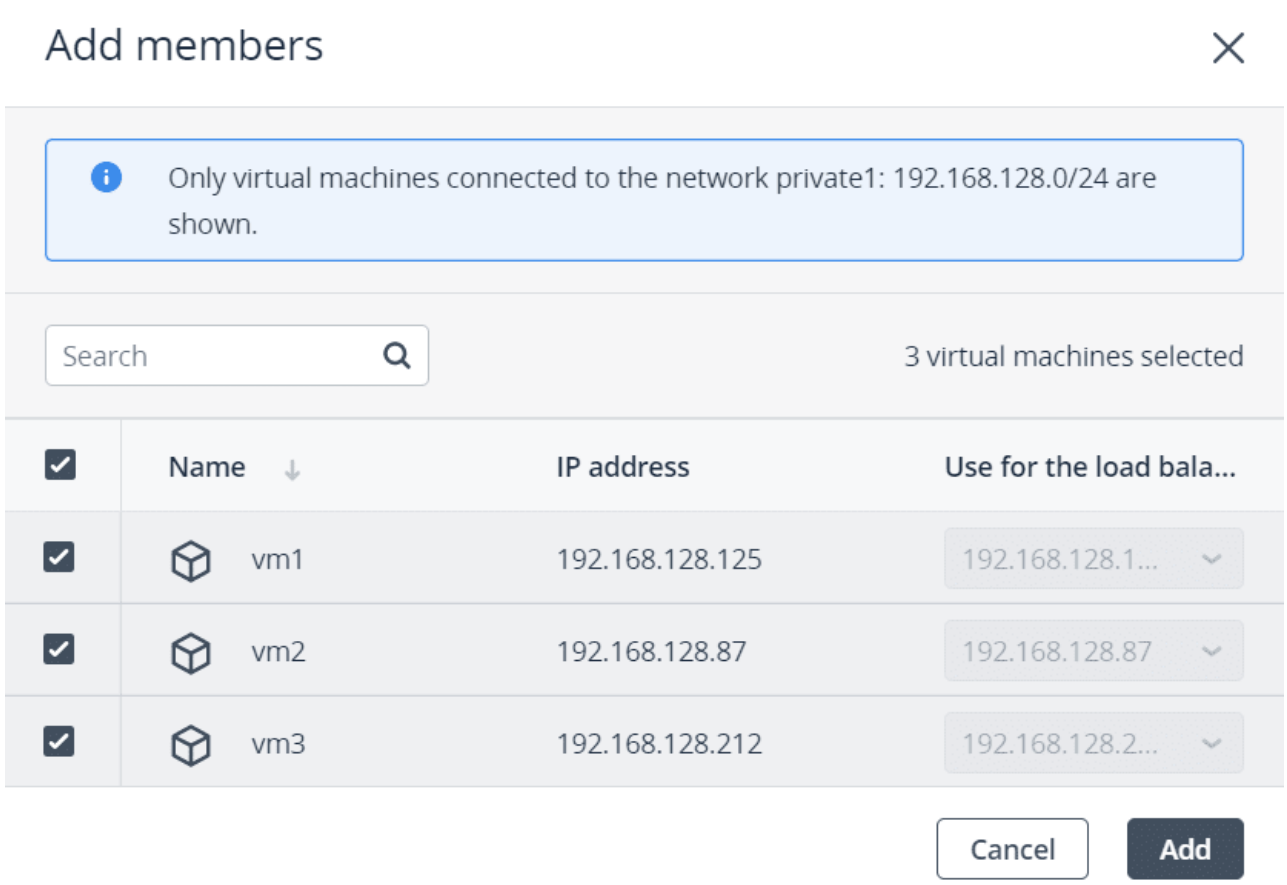
   Note: This option is not available in the SSL passthrough mode.

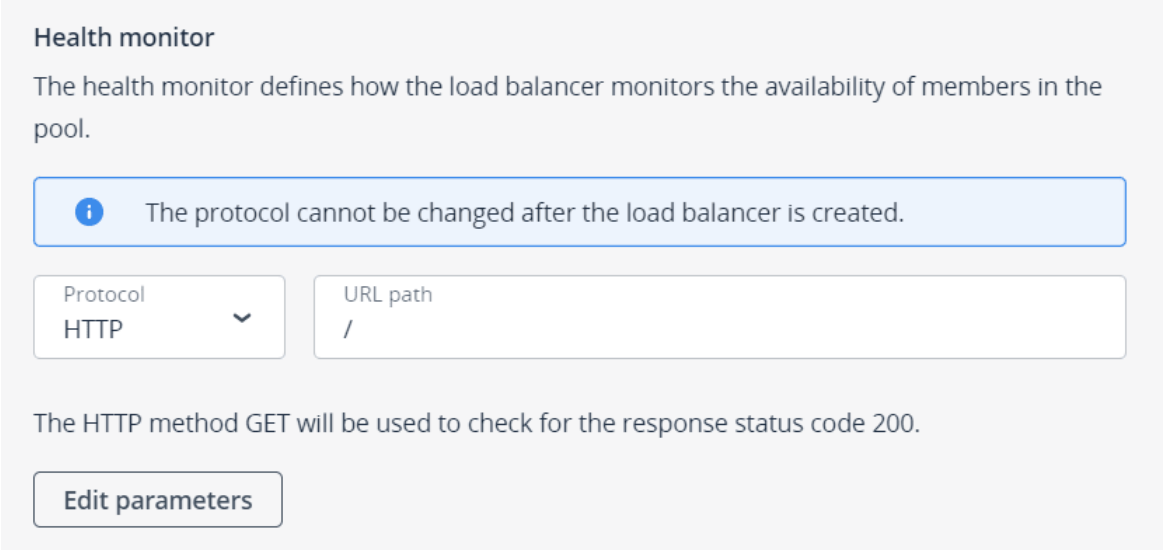
4. In the **Members** section, add members, that is, virtual machines, to the balancing pool by clicking **Add**. Each VM can be included to multiple balancing pools.

   In the **Add members** window that opens, select the desired VMs, and then click **Add**.

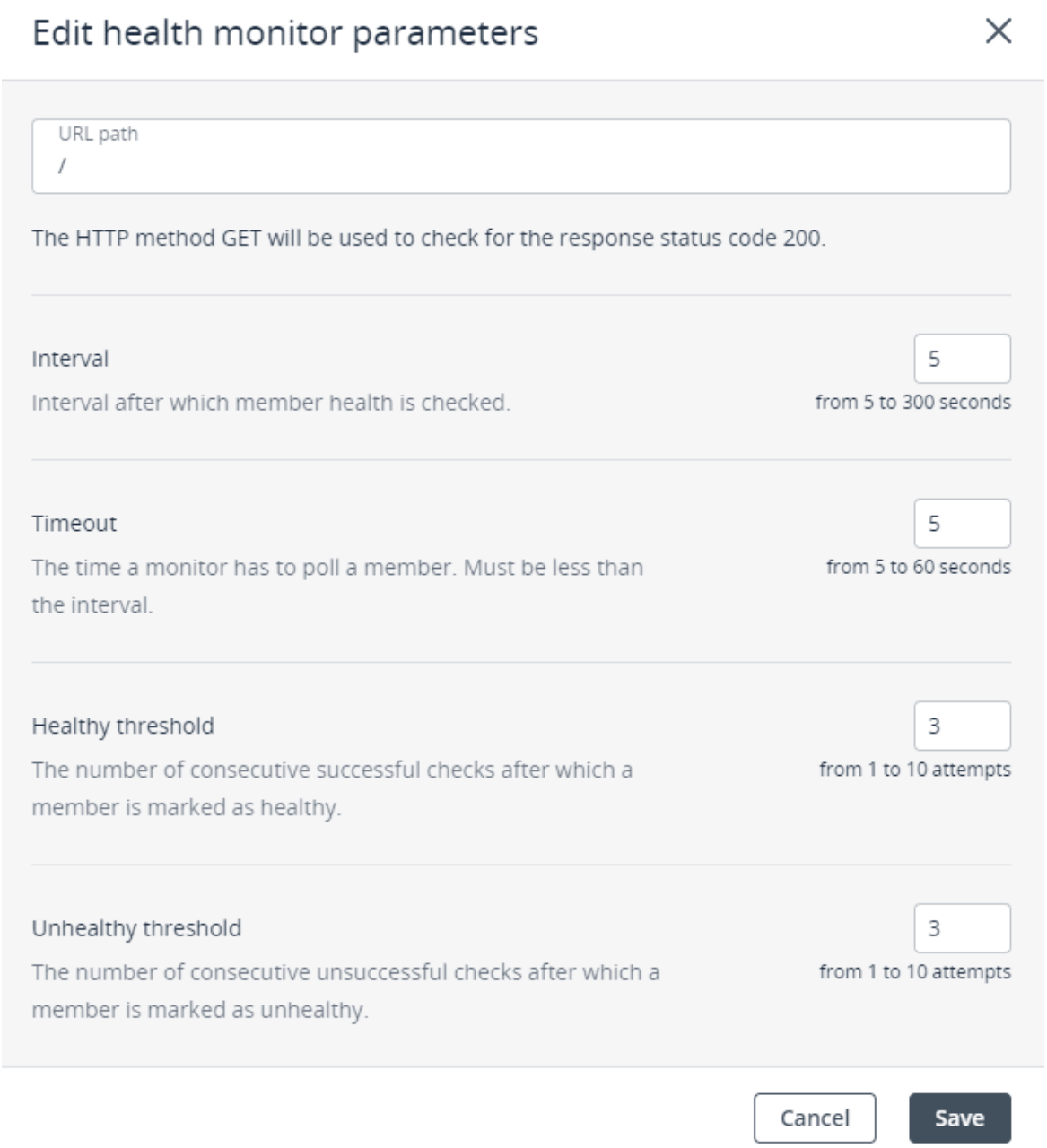
   Note: You can select only between VMs that are connected to the chosen network.



5. 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.



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**.



6. Click **Create**.

The newly added pool will appear in the list of balancing pools.

14.2.2 Editing Balancing Pool

- 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**.

14.2.3 Addinng More Mmbers to Balancing Pool

1. Click the ellipsis icon next to the required balancing pool, and then click **+ Add members**.
2. In the **Add members** window, select virtual machines to be added to the balancing pool, and then click **Add**.

14.2.4 Removing Balancing Pool

1. Click the ellipsis icon next to the required balancing pool, and then click **Delete**.
2. Click **Delete** in the confirmation window.

14.3 Monitoring Load Balancers

14.3.1 Monitoring Performance and Health of Load Balancer

Open the **Overview** tab on the load balancer 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.

14.4 Modifying and Deleting Load Balancers

14.4.1 Editing the Name or Description of Load Balancer

1. On the **Load balancers** screen, click a load balancer you want to edit.
2. On the load balancer right pane, click **Edit**.
3. In the **Edit load balancer** window, modify the name or description, and then click **Save**.

14.4.2 Disabling or Enabling Load Balancer

1. On the **Load balancers** screen, click a load balancer you want to change.
2. On the load balancer right pane, click **Disable** or **Enable**, depending on the load balancer's current state.

14.4.3 Removing Load Balancer

1. On the **Load balancers** screen, click a load balancer to delete.
2. On the load balancer right pane, click **Delete**.
3. Click **Delete** in the confirmation window.

CHAPTER 15

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 Cloud 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.

Limitations:

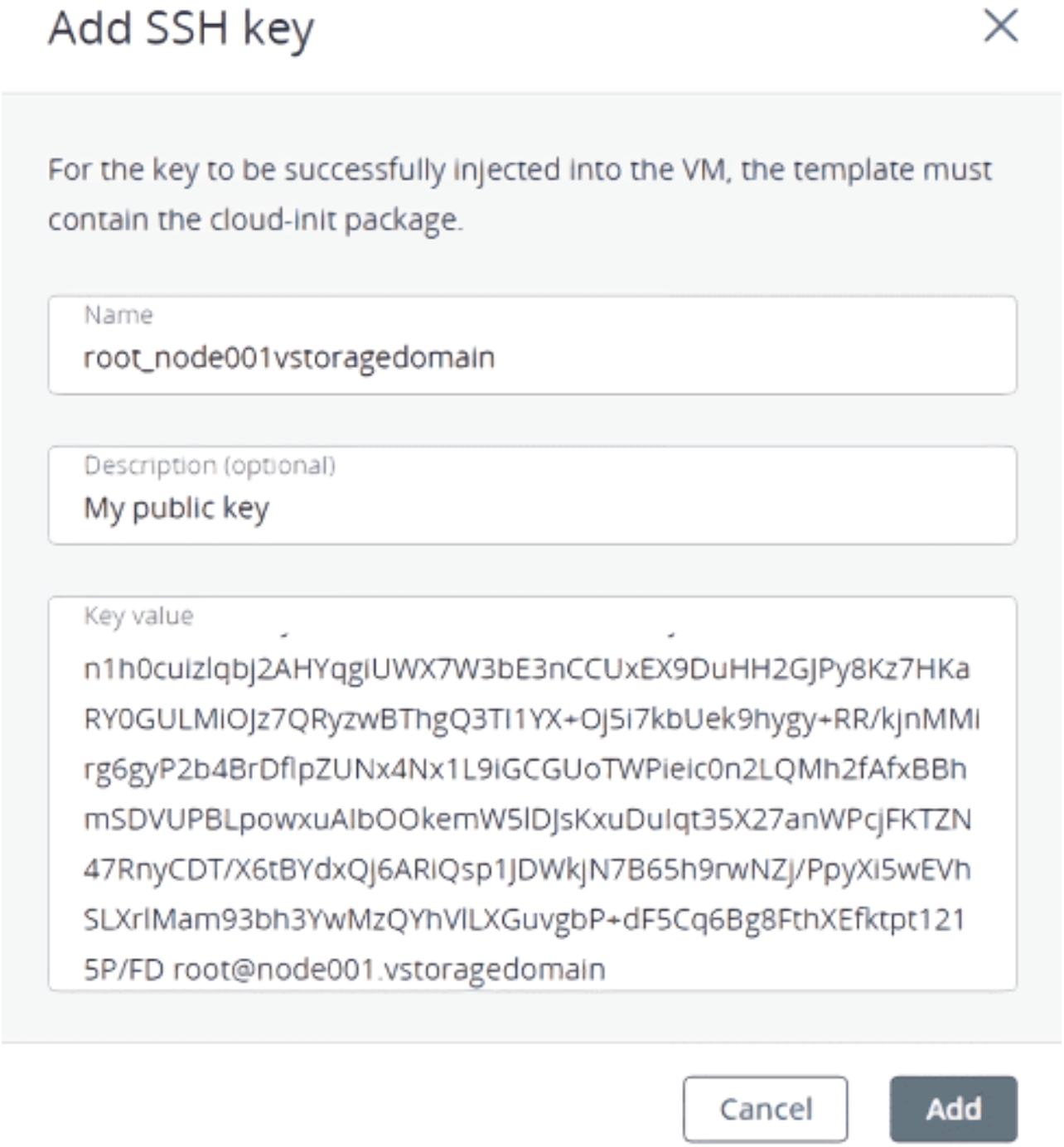
- You can specify an SSH key only if you deploy a VM from a template or boot volume (not an ISO image).
- If a key has been injected into one or more VMs, it will remain inside those VMs even if you delete it from the panel.

15.1 Adding Public Key

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
n1h0cuizlqbJ2AHYqgIUWX7W3bE3nCCUxEX9DuHH2GJPy8Kz7HKa
RY0GULMIOJz7QRyzwBThgQ3TI1YX+OJ5i7kbUek9hygy+RR/kjnMMI
rg6gyP2b4BrDflpZUNx4Nx1L9iGCGUoTWPieic0n2LQMh2fAfxBBh
mSDVUPBLpowxuAIbOOkemW5lDJsKxuDuIqt35X27anWPcjFKTZN
47RnyCDT/X6tBYdxQJ6ARiQsp1JDWkjN7B65h9rwNZj/PpyXi5wEVh
SLXrlMam93bh3YwMzQYhVlLXGuvgbP+dF5Cq6Bg8FthXEfktpt121
5P/FD root@node001.vstoragedomain
Cancel
Add

15.2 Deleting Public Key

1. On the **SSH keys** screen, select the SSH key you want to delete, and then click **Delete**.
2. Click **Delete** in the confirmation window.

If this key has been injected into one or more virtual machines, it will remain inside those virtual machines.

CHAPTER 16

Managing Virtuozzo Hybrid Cloud Using API

16.1 Access to Virtuozzo Hybrid Cloud API

Virtuozzo Hybrid Cloud uses OpenStack-based API with password credentials.

Note: For more information about OpenStack API, please refer to OpenStack Authentication and API Request Workflow.

Here is a list of access parameters:

Name	Python CLI Variable	Value
User Domain	`OS_USER_DOMAIN_NAME`	Your domain name.
Project Domain	`OS_PROJECT_DOMAIN_NAME`	Your project domain name.
Project Name	`OS_PROJECT_NAME`	Your project name. You can view it in the upper-right corner of the self-service panel.
Username	`OS_USERNAME`	Your user name.
Password	`OS_PASSWORD`	Your password.

Continued on next page

Table 16.1.1 -- continued from previous page

Name	Python CLI Variable	Value
URL	OS_AUTH_URL	Your cloud URL should be one of the following: • https://eu1-cloud.virtuozzo.com:/5000/v3 • https://eu3-cloud.virtuozzo.com:/5000/v3 • https://us1-cloud.virtuozzo.com:/5000/v3

16.2 Access Example with Python CLI

The example below is for the Ubuntu system you can use to connect to Virtuozzo Hybrid Cloud API. It can be a virtual machine on the Virtuozzo Hybrid Cloud or your laptop with a similar system:

1. Install the Python 3 OpenStack client:

```
# apt install python3-openstackclient python3-heatclient python3-magnumclient python3-octaviac
```

2. Prepare a credentials file:

Note: In the request below, replace the values with your ones.

```
echo -e "export OS_PROJECT_DOMAIN_NAME=vhc-pc \nexport OS_USER_DOMAIN_NAME=vhc-pc \nexport OS_
```

3. Load the credentials into an Env Var:

```
. /cloud-cred.rc
```

Now, you can use the OpenStack CLI to manage your cloud objects. For more details, go to OpenStack CLI Reference. For example:

```
# openstack server list
```

The abovementioned command will output a list of virtual machines.

16.3 API Automation Solutions

You can use the same credential information set for Heat Orchestration and Terraform on Heat.

CHAPTER 17

Reporting Support Issue to Virtuozzo

17.1 How Technical Support Works

All technical support is provided and documented using a support ticket system.

17.2 How to Get Technical Support

You can submit, check, and update your support requests on the self-service portal. Besides, you can see the history of all tickets you or your team raised. Anyone from your organization can have access to the self-service portal.

Self-service portal: https://support.virtuozzo.com/hc/en-us

To access the portal, you should have an account automatically created when you sign a contract with Virtuozzo. You must reset your password for the first time accessing the Virtuozzo support self-service portal. You can do it by clicking the **Forgot password?** link upper right, filling in the corresponding field with your email address, and clicking the **Submit** button. Password reset instructions will be sent out to your email. If you have any issues with a password reset or any other portal access issues, reach us via email: support-portal-issues@virtuozzo.com

Severity definitions: https://www.virtuozzo.com/all-supported-products/severity-level-definitions/

Support scope: https://www.virtuozzo.com/all-supported-products/scope-of-support/