Acronis Cyber Infrastructure 3.5

Self-Service Guide

July 21, 2020

Contents

CHAPTER 1

About This Guide

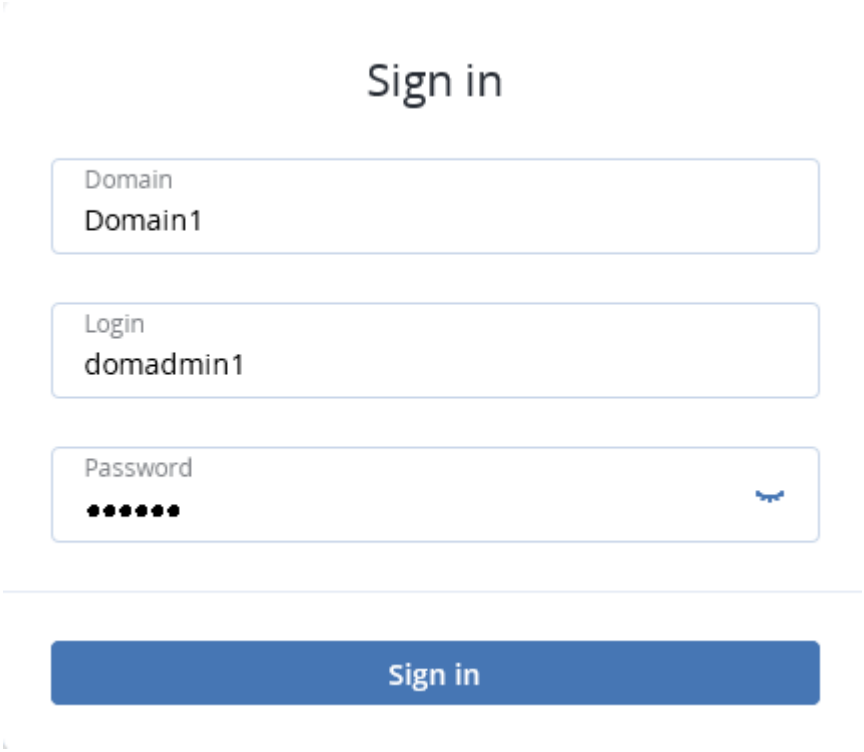
This guide is intended for domain administrators and project members and explains how to manage project users and compute resources using the self-service panel.

CHAPTER 2

Logging in to the Self-Service Panel

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

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



CHAPTER 3

Managing Users and Projects

A user can be assigned one of the following roles:

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

You can create, view, and edit users on the **All users** tab. Creating a user account differs slightly depending on the user role and is described in the following sections.

To edit the user credentials or permissions, click the ellipsis button next to the user and then click **Edit**.

Enabling and disabling a user account means allowing and prohibiting user login, respectively.

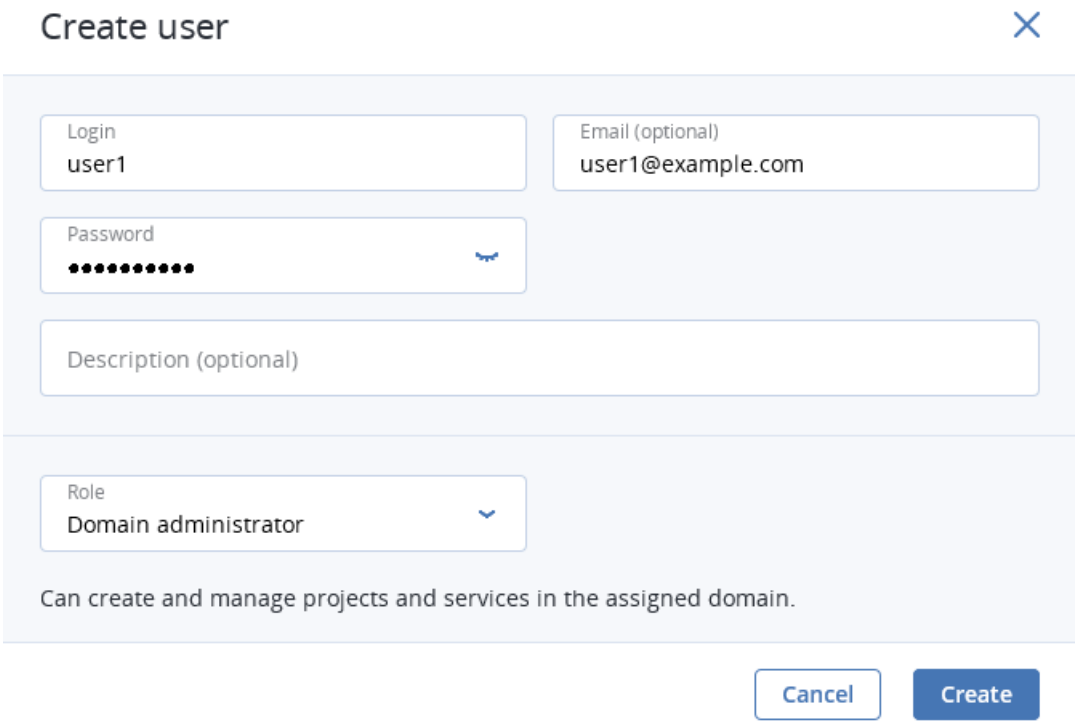
To enable/disable or remove a user, click the corresponding ellipsis button and select the desired action.

3.1 Creating Users

To create a user, do as follows:

1. Select the domain in the drop-down list in the top right corner.
2. Switch to **All users** and click **Create user**.
3. In the **Create user** window, specify the user name, password, and, if required, a user e-mail address and description. The user name must be unique within a domain.
4. Select the desired role from the **Role** drop-down menu.

5. Click **Create**.

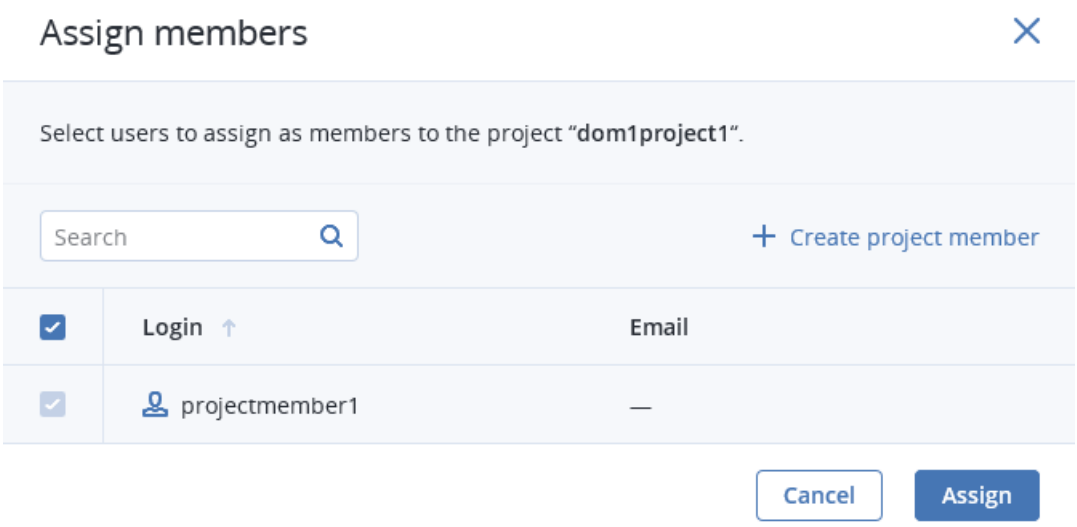


3.2 Assigning Users to Projects

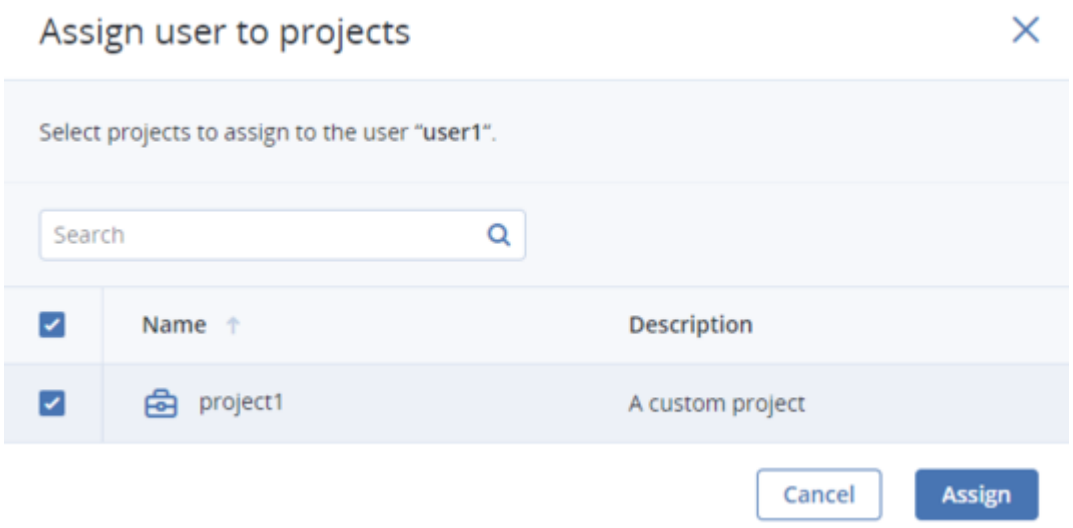
Domain administrators can manage project members' assignment on the **All projects** and **All users** screens.

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

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



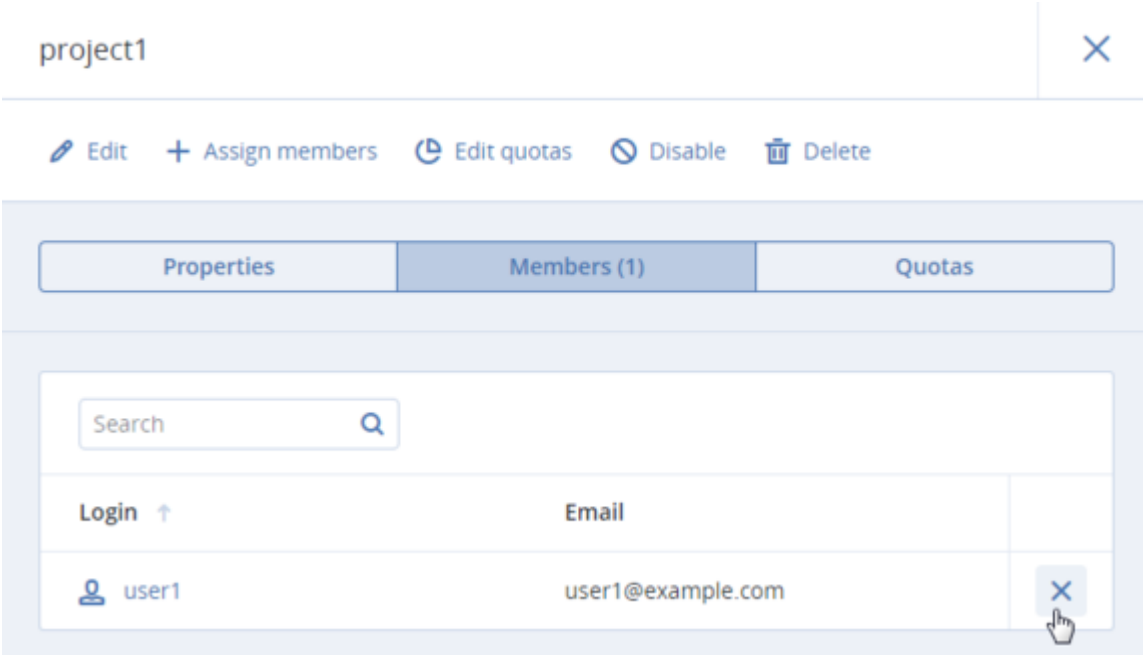
- On the **All users** screen:
  1. Click the user account with the **Project member** role whom you want to assign to the project.
  2. On the user panel, click **Assign to project**.
  3. On the **Assign user to projects** window, select one or multiple projects and click **Assign**.



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

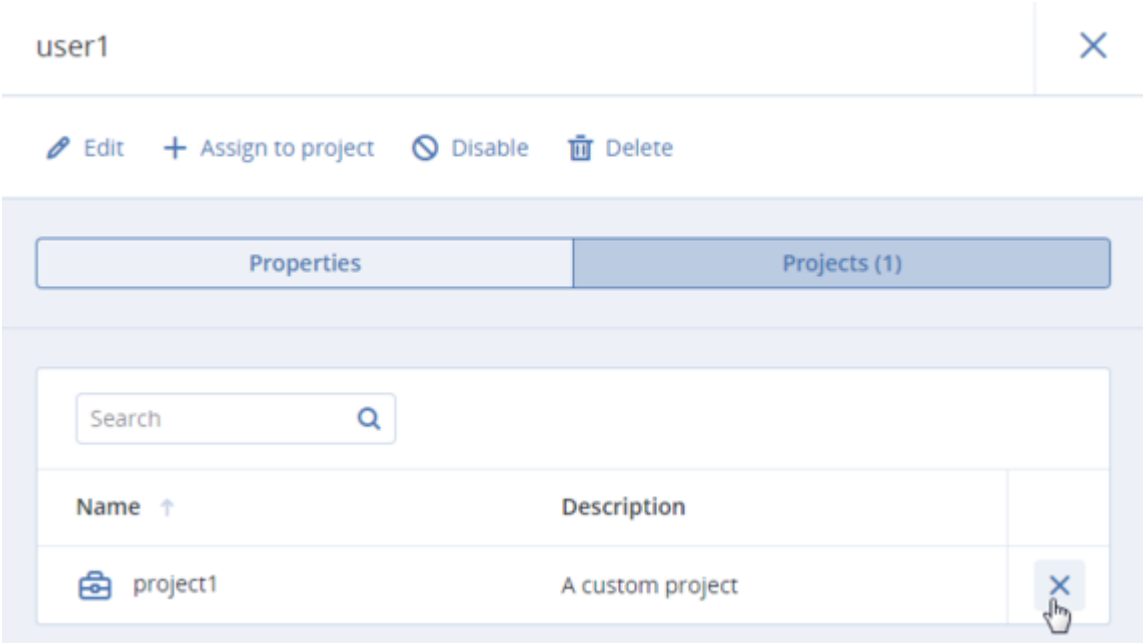
- On the **All projects** screen:
  1. Click the project to unassign users from.
  2. On the project panel, open the **Members** tab.

3. Click the cross icon next to a user you want to unassign.



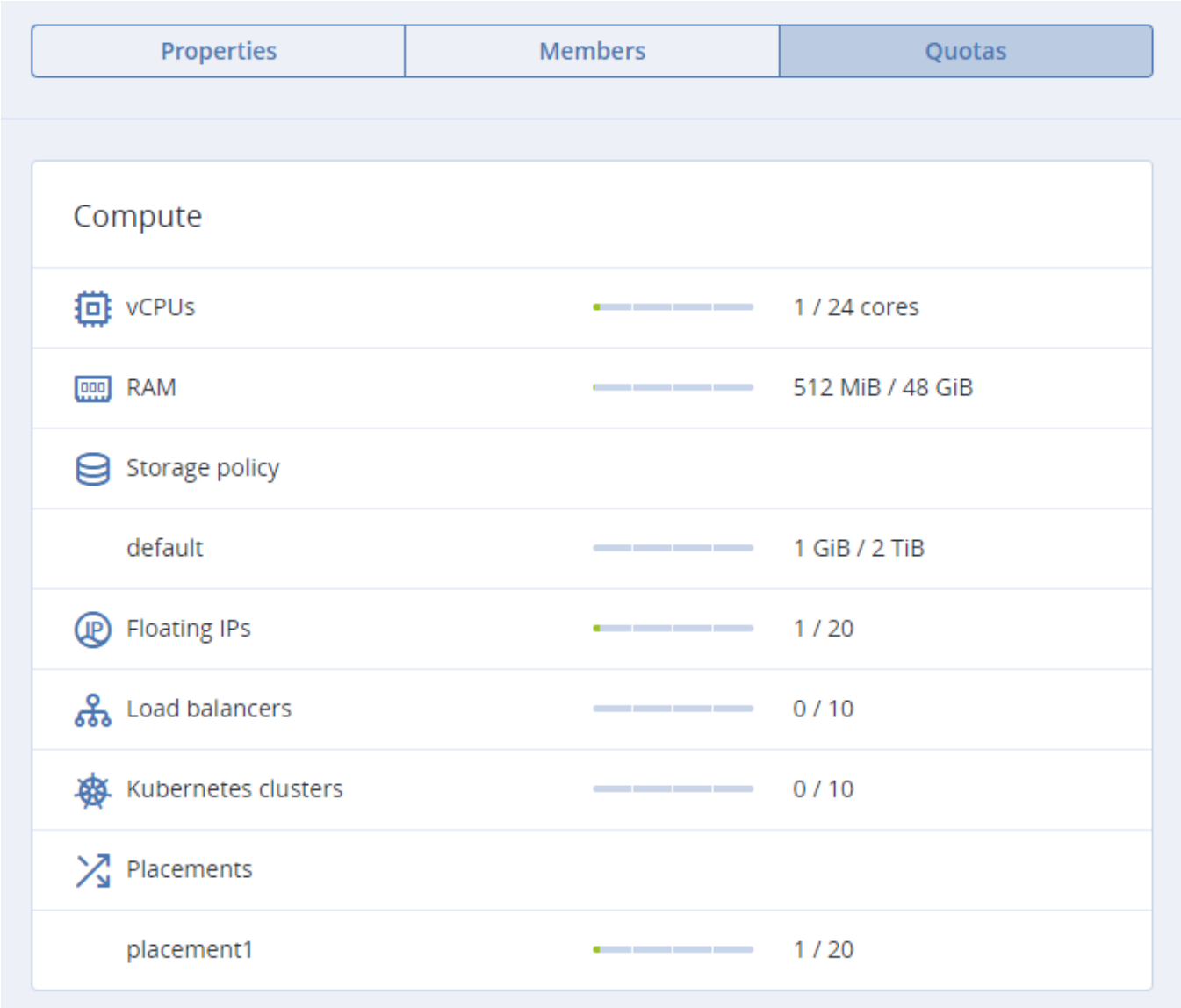
- On the **All users** tab:

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



3.3 Viewing Project Quotas

Each project is allocated a certain amount of compute resources by means of quotas. To view quotas of a project, open **PROJECTS**, click the desired project in the list, and switch to the **Quotas** tab.



CHAPTER 4

Managing Compute Resources

4.1 Managing Virtual Machines

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

- A virtual machine resembles and works like a regular computer. It has its own virtual hardware. Software applications can run in virtual machines without any modifications or adjustment.
- Virtual machine configuration can be changed easily, e.g., 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:

Table 4.1.1: Virtual machine hardware

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

A logical CPU is a core (thread) in a multicore (multithreading) processor.

4.1.1 Supported Guest Operating Systems

The following guest operating systems have been tested and are supported in virtual machines:

Table 4.1.1.1: Windows guest operating systems

Operating System	Edition	Architecture
Windows Server 2019	Essentials, Standard, Datacenter	x64
Windows Server 2016	Essentials, Standard, Datacenter	x64
Windows Server 2012 R2	Essentials, Standard, Datacenter	x64
Windows Server 2012	Standard, Datacenter	x64
Windows Server 2008 R2	Standard, Datacenter	x64
Windows Server 2008	Standard, Datacenter	x64
Windows 10	Home, Professional, Enterprise, Enterprise 2016 LTSB	x64
Windows 8.1	Home, Professional, Enterprise	x64
Windows 7	Home, Professional, Enterprise	x64

Table 4.1.1.2: Linux guest operating systems

Operating System	Architecture
CentOS 8.x	x64
CentOS 7.x	x64
CentOS 6.x	x64
RHEL 8.x	x64
RHEL 7.x	x64
Debian 9.x	x64
Ubuntu 20.04.x	x64
Ubuntu 18.04.x	x64
Ubuntu 16.04.x	x64

4.1.2 Creating Virtual Machines

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

- A guest OS source (see *Managing Images* (page 32)):

- a distribution ISO image of a guest OS to install in the VM, or
- a template that is a boot volume in the QCOW2 format, or
- a boot volume

> **Note:** To obtain a boot volume, create a volume as described in *Managing Volumes* (page 34), attach it to a VM, install an operating system in it, then delete the VM.

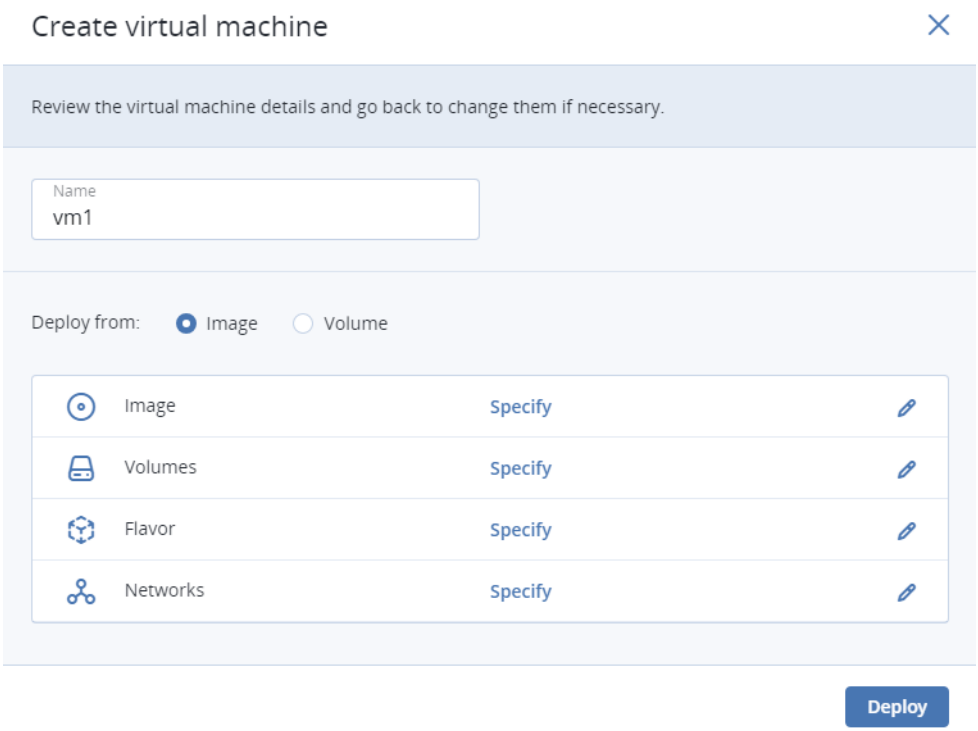
- One or more virtual networks (see *Managing Private Virtual Networks* (page 41))
- An SSH key (see *Managing SSH Keys* (page 57))

> **Note:** You can specify an SSH key only when creating VMs from a template or boot volume.

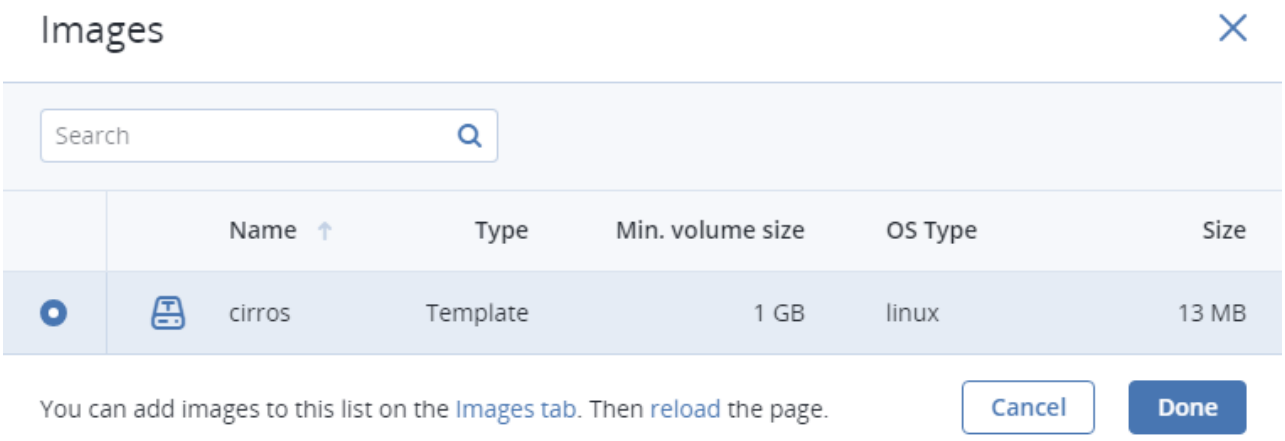
Note: Virtual machines are created with the host CPU model by default. Having compute nodes with different CPUs may lead to live migration issues. To avoid them, you can manually set CPU model for all new VMs as described in Setting Virtual Machines CPU Model.

To create a VM, do the following:

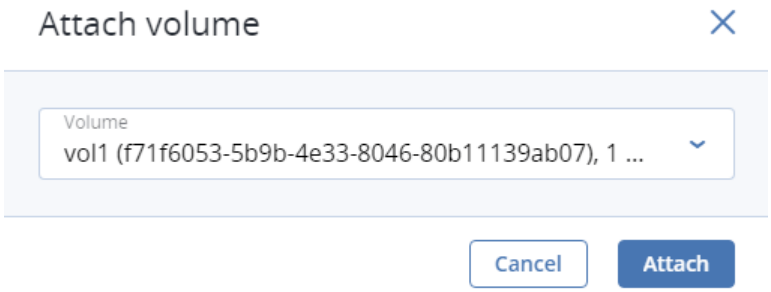
1. On the **Virtual machines** screen, click **Create virtual machine**. A window will open where you will need to specify VM parameters.

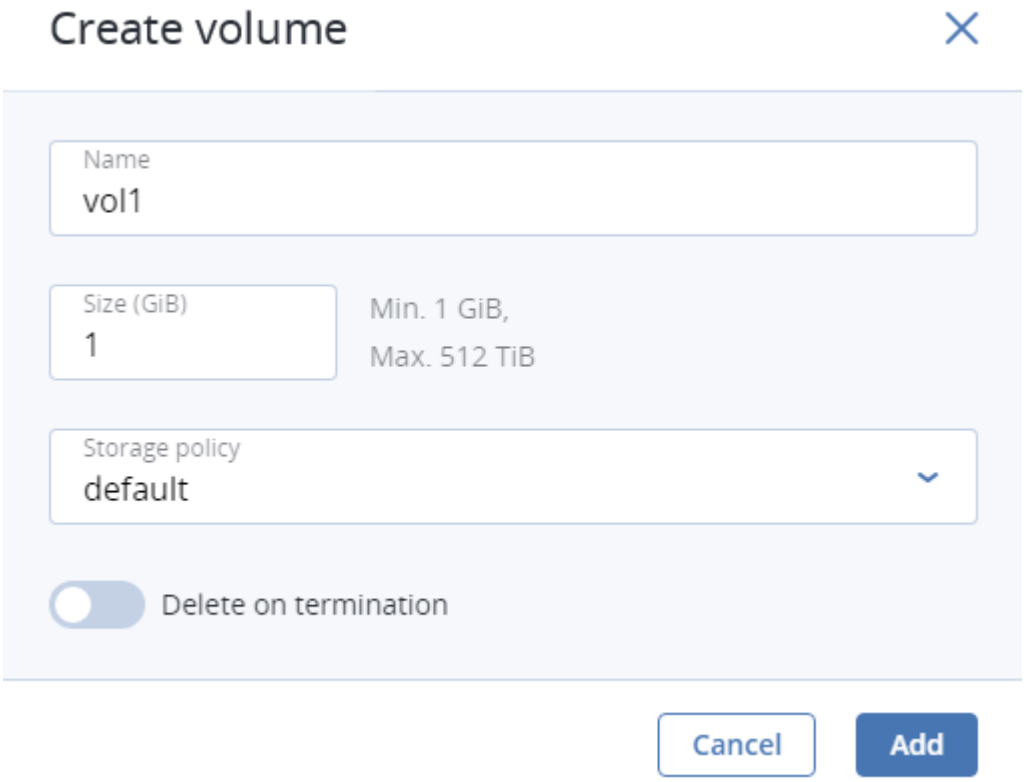


2. Specify a name for the new VM.
3. In **Deploy from**, choose **Volume** if you have a boot volume or want to create one. Otherwise, choose **Image**.
4. Depending on your choice, click the pencil icon in the **Volumes** or **Image** section and do one of the following:
   - In the **Images** window, select the ISO image or template and click **Done**.



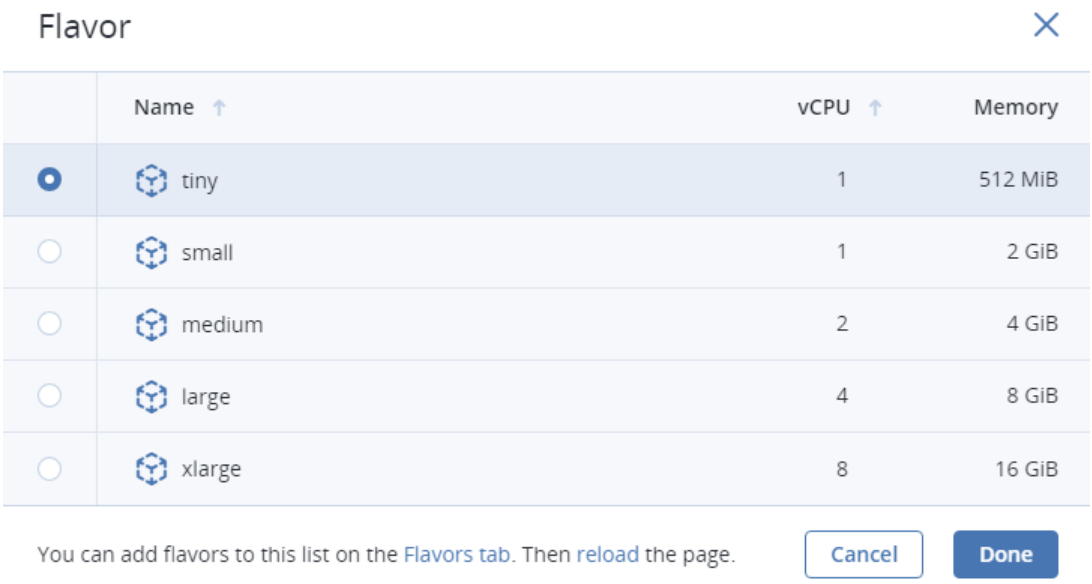
- In the **Volumes** window, do one of the following:
  - If you have prepared a volume with an installed guest OS, click **Attach**, find and select the volume, and click **Done**.



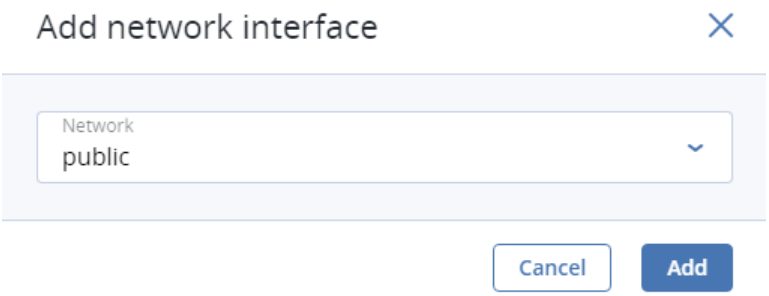


5. Optionally, in the **Volumes** window, click **Add** or **Attach** to create or attach any other volumes you need. To select a volume as bootable, place it first in the list by clicking the up arrow button next to it.
6. After you select an image or a volume, the **Placement** drop-down list is displayed. Placements are created by the administrator to group nodes or VMs sharing a distinctive feature, like a special license. Select the placement corresponding to the VM characteristics. For more information, see Managing Placements.

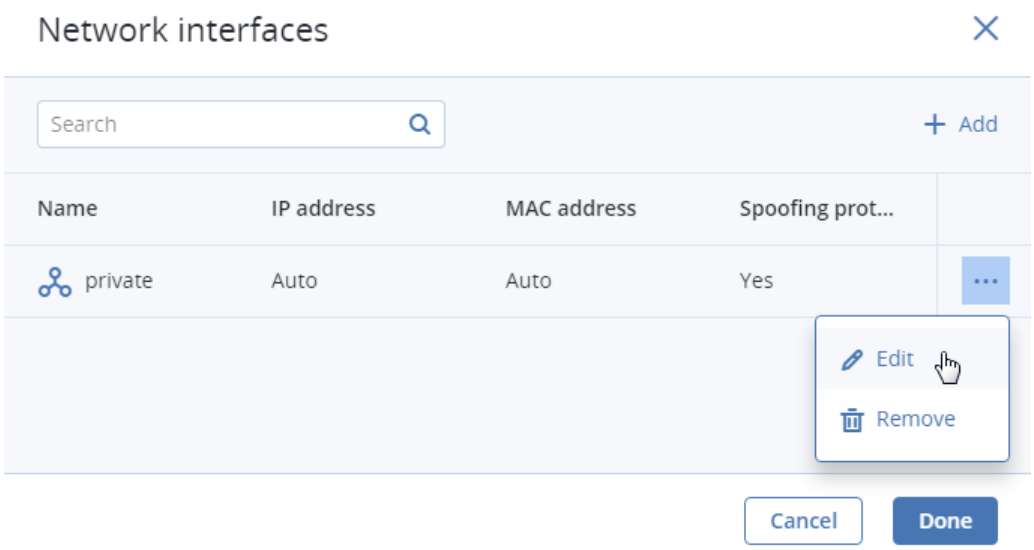
7. In the **Flavor** window, choose a flavor and click **Done**.



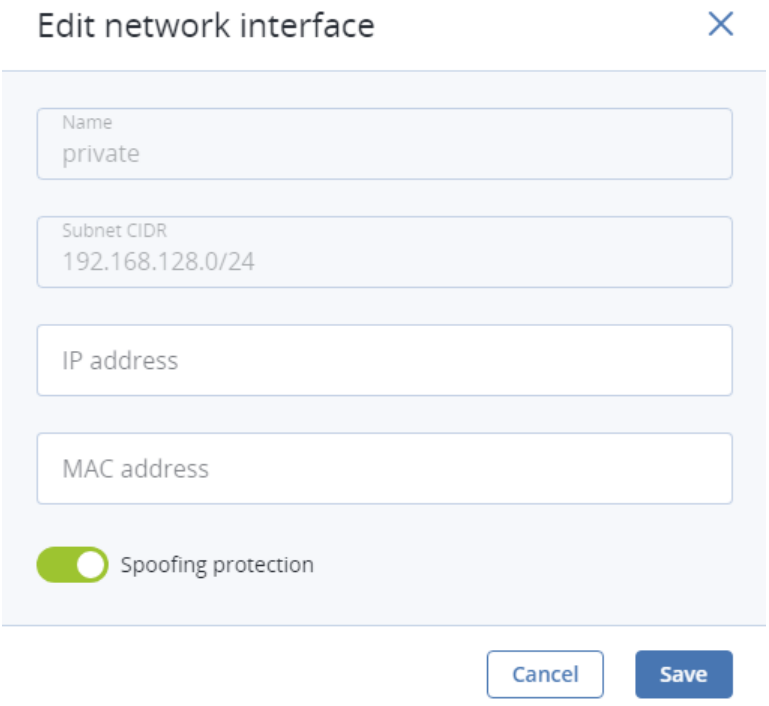
8. In the network window, click **Add**, select a virtual network interface and click **Add**. It will appear in the **Network interfaces** list.



You can edit additional parameters of newly added network interfaces, like IP and MAC addresses and spoofing protection. To do this, click interface's ellipsis icon, then **Edit**, and set parameters in the **Edit network interface** window.



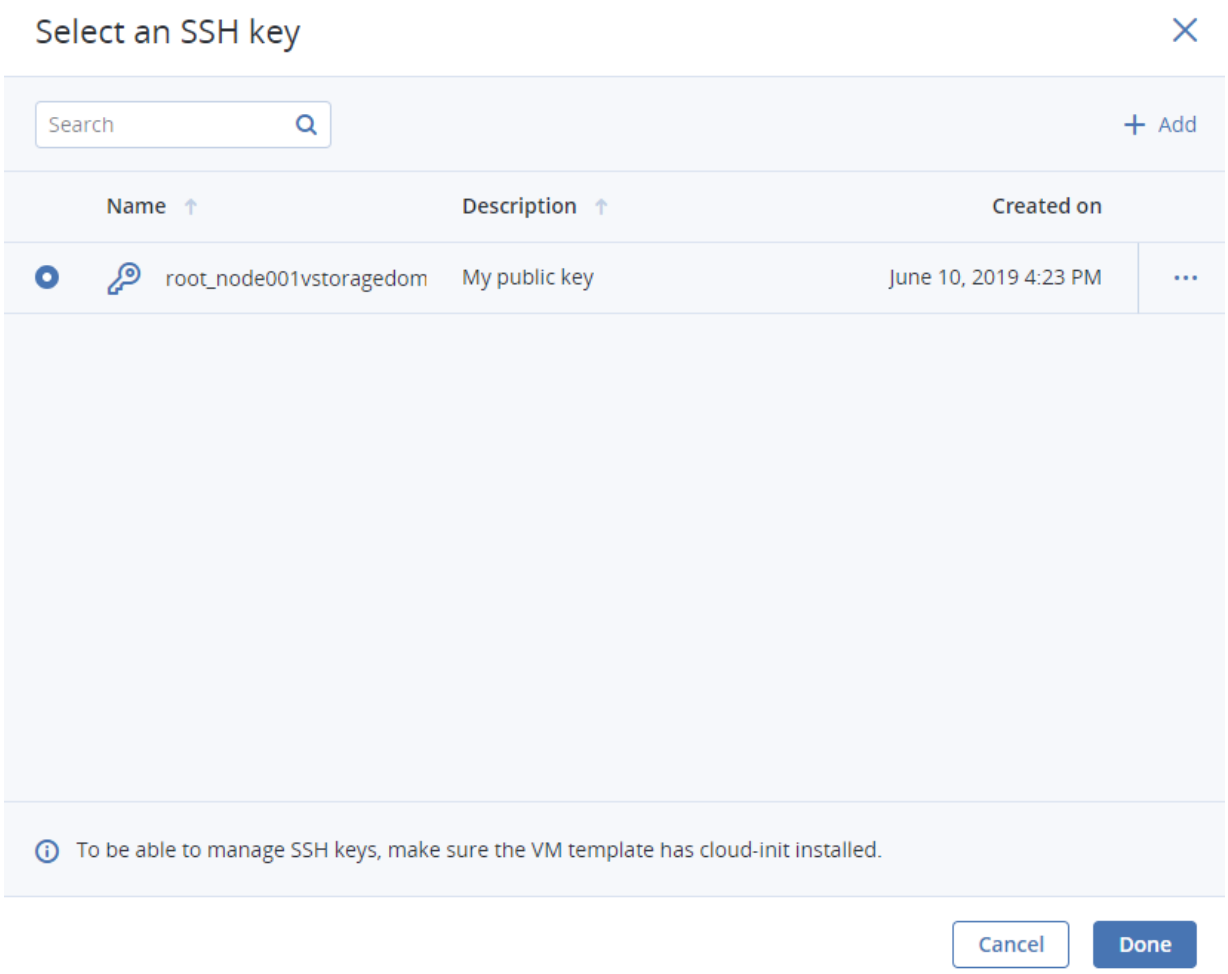
You will not be able to edit these parameters later. Instead, you will be able to delete the old network interface and replace it with a new one.



Click **Done**.

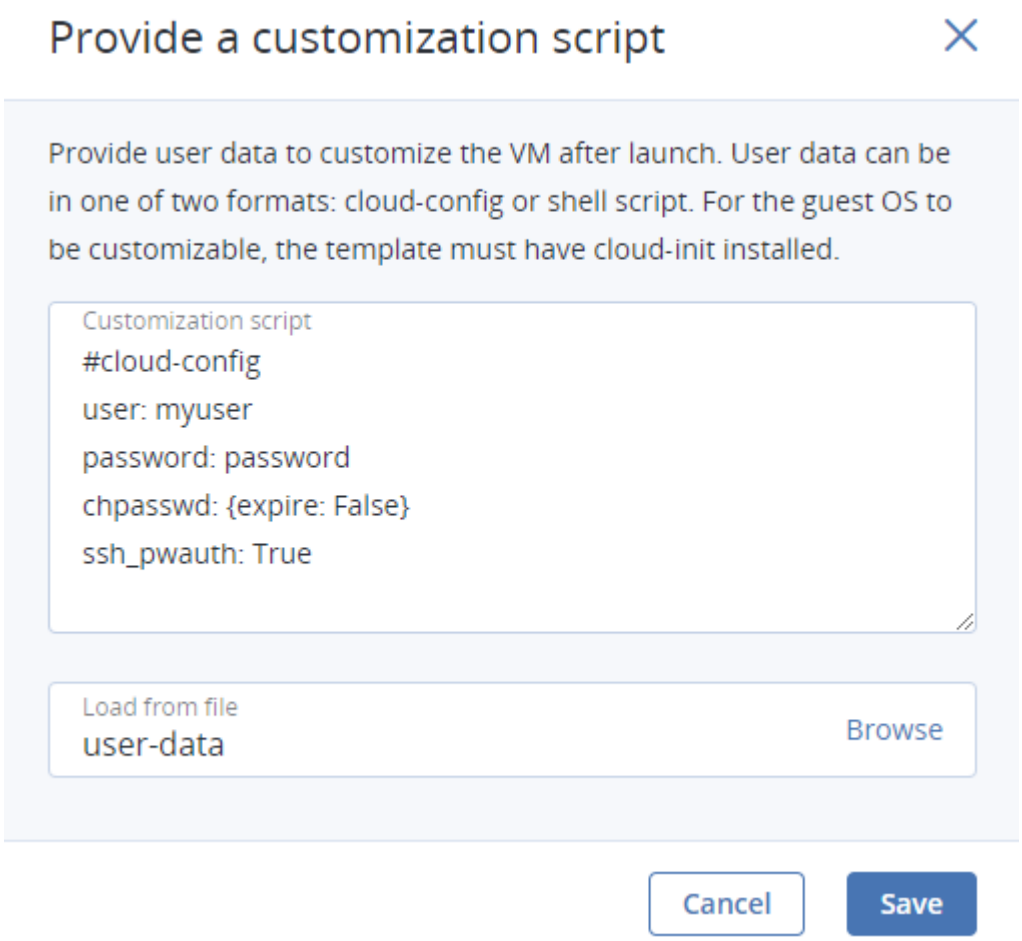
9. (Optional) If you are deploying the VM from a template or boot volume (not an ISO image), you can specify the following:

- An SSH key to be injected into the VM. To do it, select an SSH key in the **Select an SSH key** window, and click **Done**.



Note: To be able to connect to the VM via SSH, make sure the VM template or boot volume has cloud-init and OpenSSH installed (see Preparing Templates).

- User data to customize the VM after launch. You can specify user data in one of two formats: cloud-config or shell script. To do it, write a script in the **Customization script** field or browse a file on your local server to load the script from.



Note: For the guest OS to be customizable, make sure the VM template or boot volume has cloud-init installed (see Preparing Templates).

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>
```

10. Back in the **Create virtual machine** window, click **Deploy** to create and boot the VM.
11. If you are deploying the VM from an ISO image (not a boot volume template or a volume with a pre-installed guest OS), select the VM, click **Console**, and install the guest OS using the built-in VNC console.

12. (Optional) If you are deploying the VM from a prepared template with an injected SSH key, you can connect to it via SSH using the username and the VM IP address:

    - For Linux templates, enter the username that is default for the cloud image OS (for example, for a CentOS cloud image, the default login is `centos`).
    - For Windows templates, enter the username that you specified during Cloudbase-Init installation.

    For example:

    ```
    # ssh myuser@10.10.10.10
    ```

4.1.3 Virtual Machine Actions Overview

After you create a virtual machine, you can manage it using the actions available for its current state. To see the full list of available actions, click the ellipsis button next to a VM or on top of its panel. Actions include:

- **Run** powers up a VM.
- **Console** connects to running VMs via the built-in VNC console. In the console browser window, you can send a key combination to a VM, take a screenshot of the console window, and download the console log.
- **Reboot** soft-reboots a running VM.
- **Shut down** gracefully shuts down a running VM.
- **Hard reboot** cuts off and restores power, then starts a VM.
- **Power off** forcibly cuts off power from a VM.
- **Shelve** unbinds a stopped VM from the node it is hosted on and releases its reserved resources such as CPU and RAM. A shelved VM remains bootable and retains its configuration, including the IP addresses.

  Virtual machines in other states can be shelved by clicking **Shut down** or **Power off** and selecting the checkbox **Shelve virtual machine** in the confirmation window.
- **Unshelve** spawns a shelved VM on a node with enough resources to host it.
- **Suspend** saves the current VM state to a file.

  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.
- **Resume** restores a VM from suspended state.

- **Download console log** downloads the console log. Make sure logging is enabled inside the VM, otherwise the log will be empty (for more information, see *Enabling Logging inside Virtual Machines* (page 18)).

  Examining console logs may be useful in troubleshooting failed virtual machines.

- **Reset state** resets the VM stuck in a failed or transitional state to its last stable state: active, shut down or shelved.

- **Delete** removes a VM from the compute cluster.

4.1.4 Enabling Logging inside Virtual Machines

VM's console log will contain log messages only if the TTY1 and TTYS0 logging levels are enabled inside the VM. For example, you can enable them as follows in Linux VMs:

1. Add the line `GRUB_CMDLINE_LINUX_DEFAULT="console=tty1 console=ttyS0"` to the file `/etc/default/grub`.
2. Depending on the boot loader, run either

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

   or

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

3. Reboot the VM.

In Windows VMs, you can enable Emergency Management Services (EMS) console redirection for this purpose. Do the following:

1. Start **Windows PowerShell** with administrator privileges.
2. In the PowerShell console, set the COM port and baud rate for EMS console redirection. As Windows VMs have only the COM1 port with the transmission rate of 9600 bps, run:

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

3. Enable EMS for the current boot entry:

   ```
   bcdedit /ems on
   ```

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

1. Start **System Configuration** with administrator privileges.

2. In the **System Configuration** windows, open the **Boot** tab, select the checkboxes **OS boot information** and **Make all boot settings permanent**.
3. Confirm the changes and restart the system.

4.1.5 Reconfiguring and Monitoring Virtual Machines

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

The default time interval for the charts is 12 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.

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

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

4.1.6 Managing Guest Tools

This section explains how to install and uninstall the guest tools. This functionality is required for creating consistent snapshots of a running VM's disks (refer to *Managing Volume Snapshots* (page 39)).

4.1.6.1 Installing Guest Tools

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

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

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

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

4.1.6.2 Uninstalling Guest Tools

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

- Inside a Windows VM:

    1. Remove the QEMU device drivers from the device manager.

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

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

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

    4. Unregister Guest Tools Monitor from Event Log:

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

    5. Delete the autorun registry key for RebootNotifier:

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

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

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

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

    After removing the guest tools, restart the virtual machine.

- Inside a Linux VM:

    1. Remove the packages:

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

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

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

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

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

2. Remove the files:

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

3. Reload the udev rules:

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

After removing guest tools, restart the virtual machine.

4.2 Managing Kubernetes Clusters

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

The prerequisites for creating a Kubernetes cluster are:

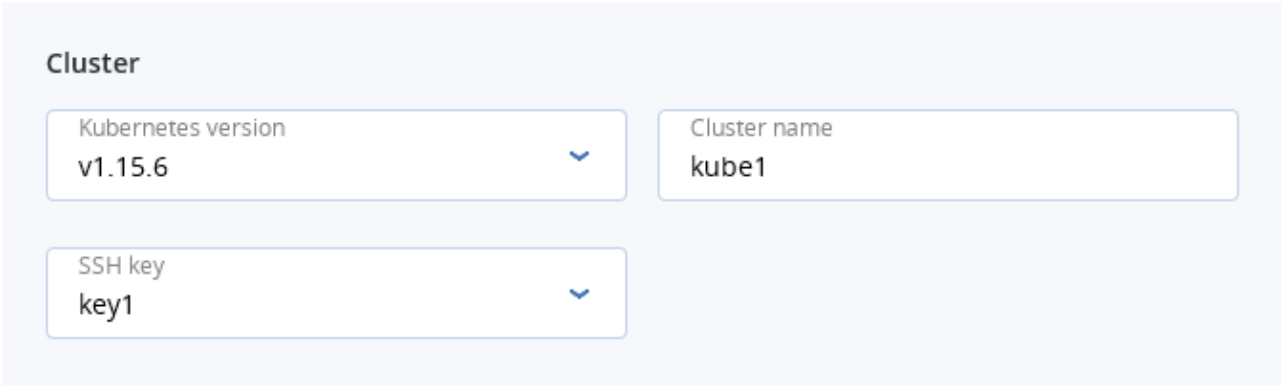
- The Kubernetes-as-a-service component. It can be deployed along with the compute cluster or later (see Creating the Compute Cluster or Managing Add-On Services).
- A network that will interconnect the Kubernetes master and worker nodes. It can be either a shared public network or a private network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

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

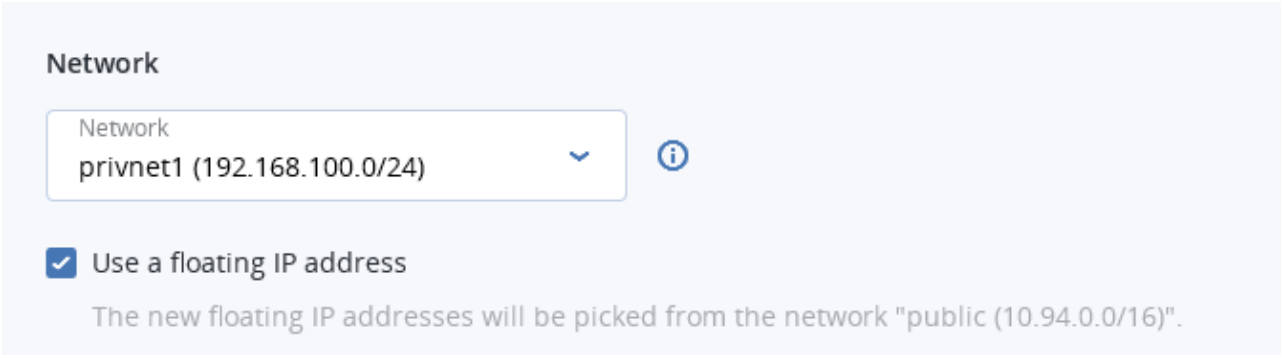
- 10.100.0.0/24—used for pod-level networking
- 10.254.0.0/16—used for allocating Kubernetes cluster IP addresses from

To create a Kubernetes cluster, on the **Kubernetes clusters** screen, click **Create** on the right. A window will open where you can set your cluster parameters:

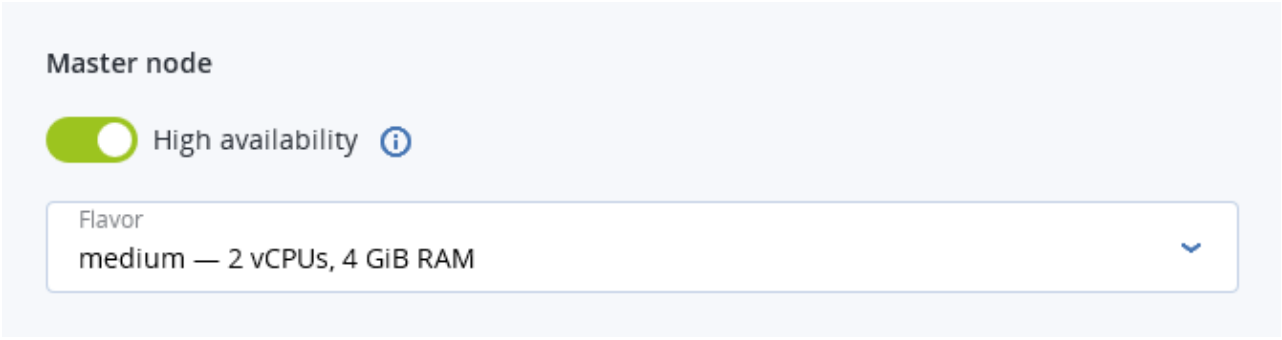
1. In the **Cluster** section, select a Kubernetes version, enter a cluster name, and select an SSH key.



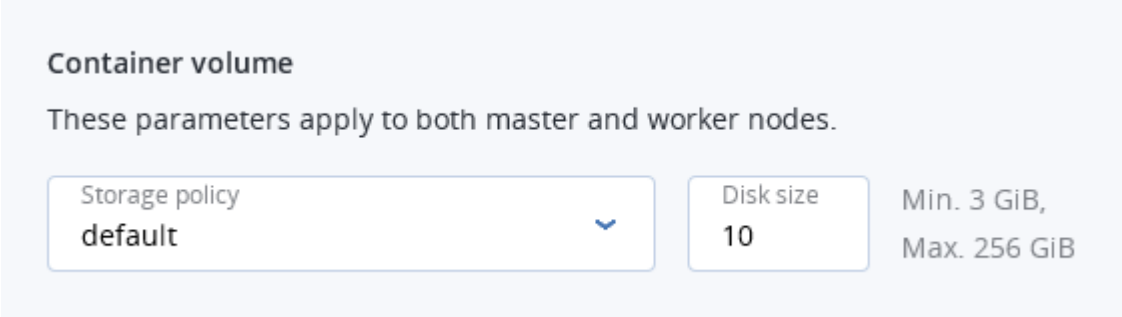
2. In the **Network** section, select a virtual router mentioned in the prerequisites above. It is also recommended to check the **Use a floating IP address** box. In this case, the Kubernetes nodes will be assigned public IP addresses, simplifying access to them.



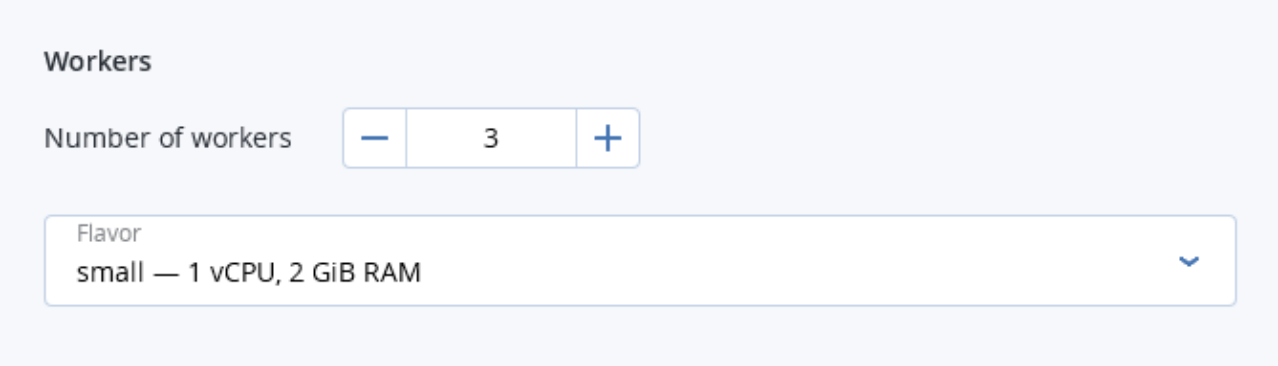
3. In the **Master node** section, select a flavor and choose whether or not to enable high availability for the master node. If you enable HA, three master node instances will be created. They will work in the Active/Active mode.



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



5. In the **Workers** section, set a number of workers to create and select a flavor for each worker.



6. Finally, click **Create**.

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

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

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

4.2.1 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.

4.2.1.1 Creating Storage Classes

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

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

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: mysc
provisioner: csi-cinderplugin
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.

4.2.1.2 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.

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

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

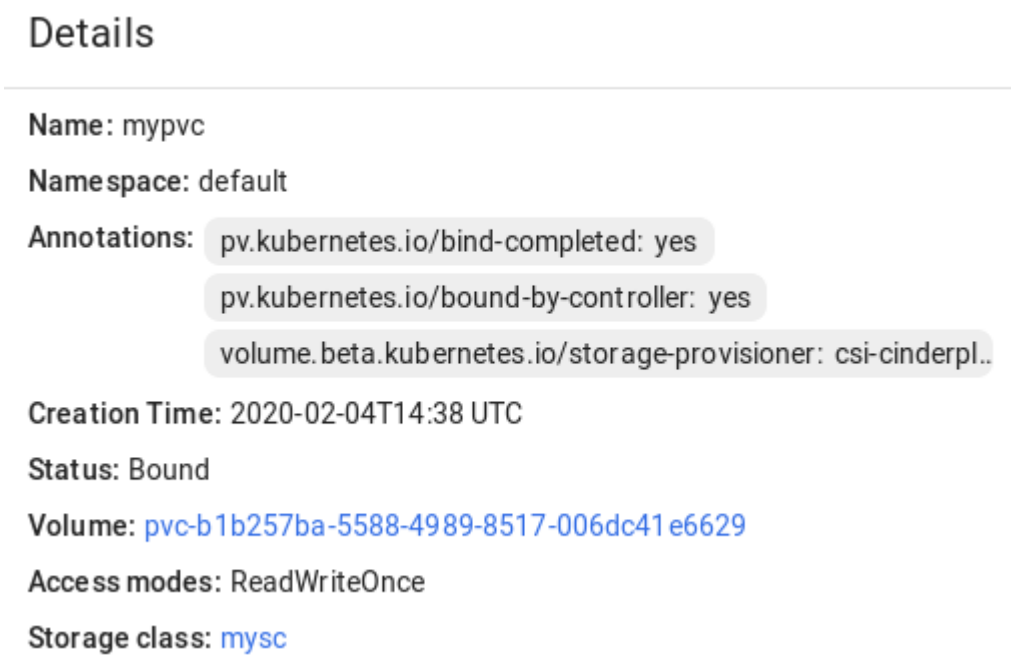
1. Access the Kubernetes cluster via the dashboard. Click **Kubernetes access** for instructions.
2. On the Kubernetes dashboard, create a storage class as described in *Creating Storage Classes* (page 25).
3. Create a persistent volume claim. To do it, click **+ CREATE** and specify the following YAML 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:
```

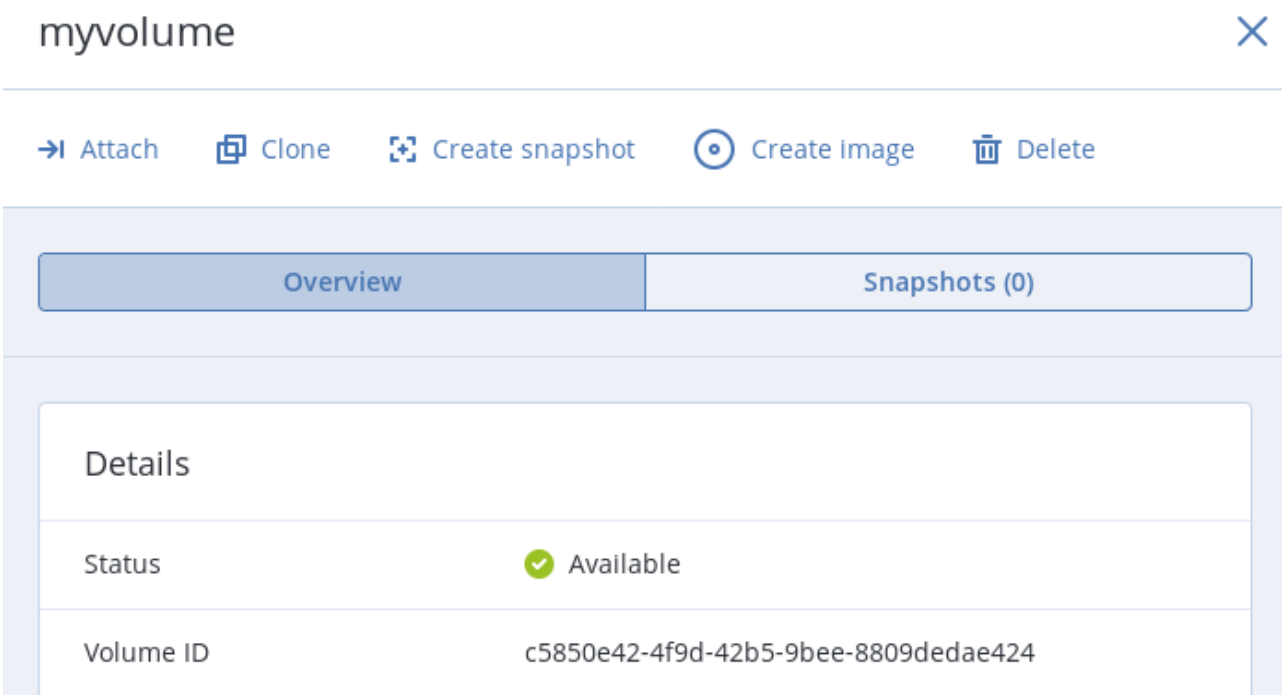
```
      - mountPath: /var/lib/www/html
        name: mydisk
  volumes:
  - name: mydisk
    persistentVolumeClaim:
      claimName: mypvc
      readOnly: false
```

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.

4.2.1.3 Statically Provisioning Persistent Volumes

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

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



2. Access the Kubernetes cluster via the dashboard. Click **Kubernetes access** for instructions.
3. On the Kubernetes dashboard, create a storage class as described in *Creating Storage Classes* (page 25).
4. Create a persistent volume. To do it, click **+ CREATE** and specify the following YAML file:

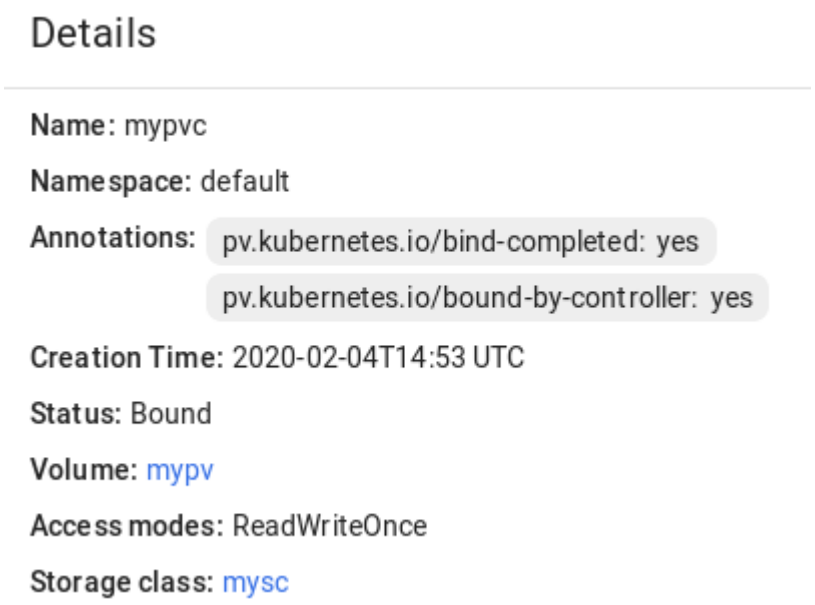
```
apiVersion: v1
kind: PersistentVolume
metadata:
  annotations:
    pv.kubernetes.io/provisioned-by: csi-cinderplugin
  name: mypv
spec:
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 10Gi
  csi:
    driver: cinder.csi.openstack.org
    fsType: ext4
    volumeHandle: c5850e42-4f9d-42b5-9bee-8809dedae424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

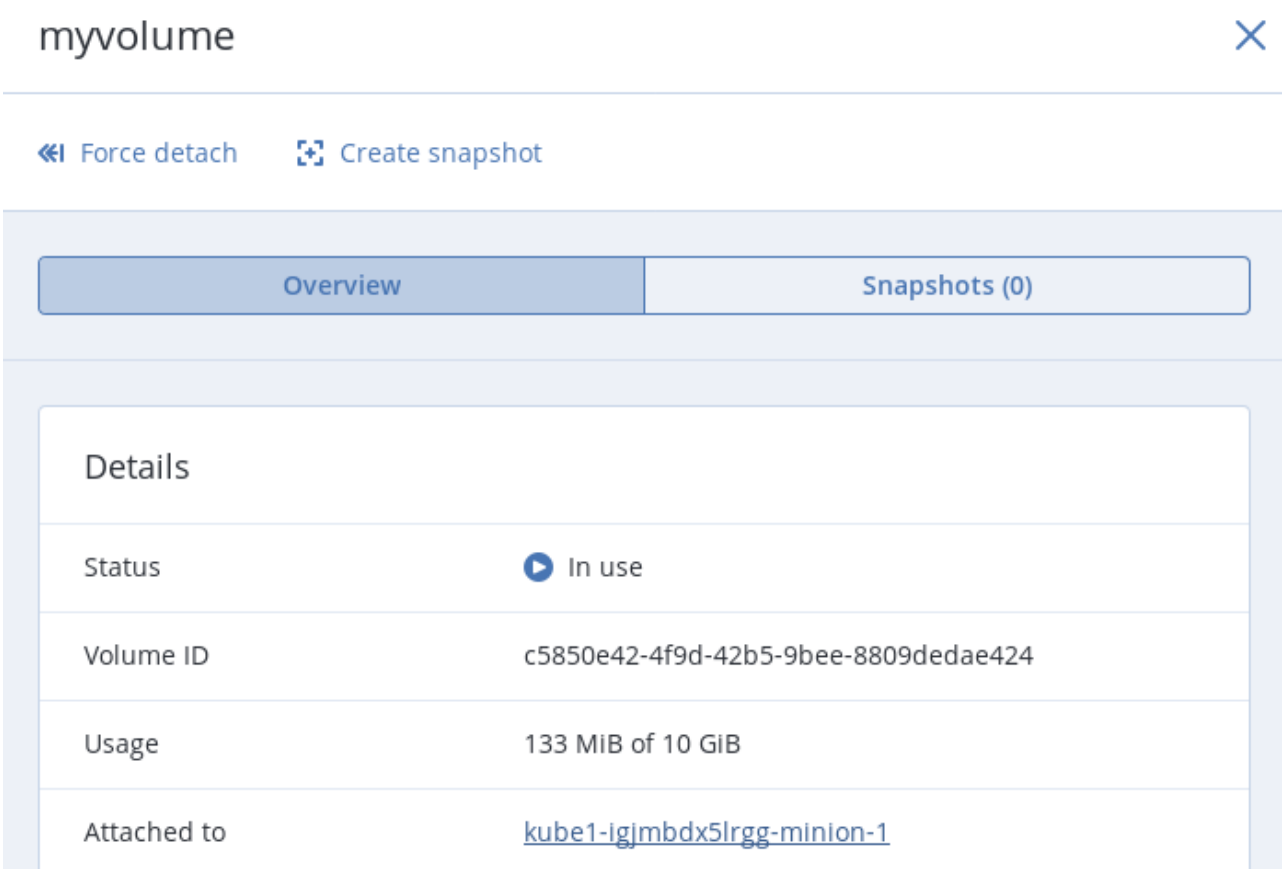
```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: mysc
```

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 3 in *Dynamically Provisioning Persistent Volumes* (page 25).

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



4.2.2 Creating External Load Balancers in Kubernetes

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

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

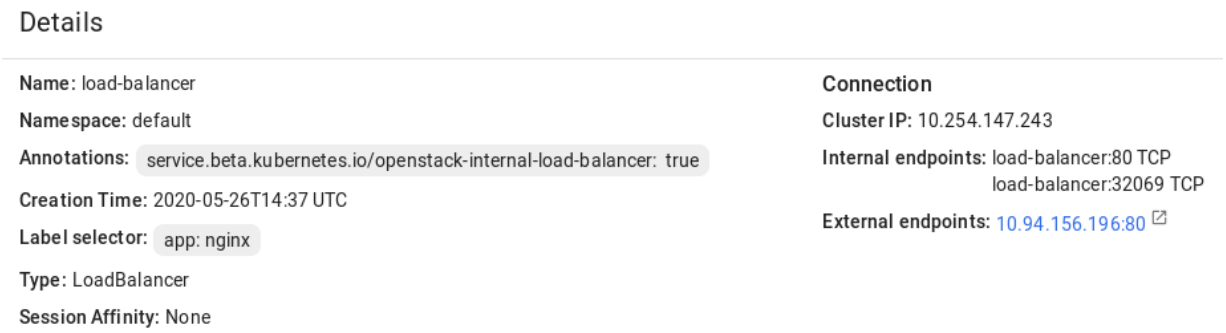
1. Access the Kubernetes cluster via the dashboard. Click **Kubernetes access** for instructions.
2. On the Kubernetes dashboard, create a deployment and service of the `LoadBalancer` type. To do it, click **+ CREATE** and specify a YAML file that defines these objects. For example:
   - If you have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

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
```

```
  name: nginx
spec:
  replicas: 2
  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.

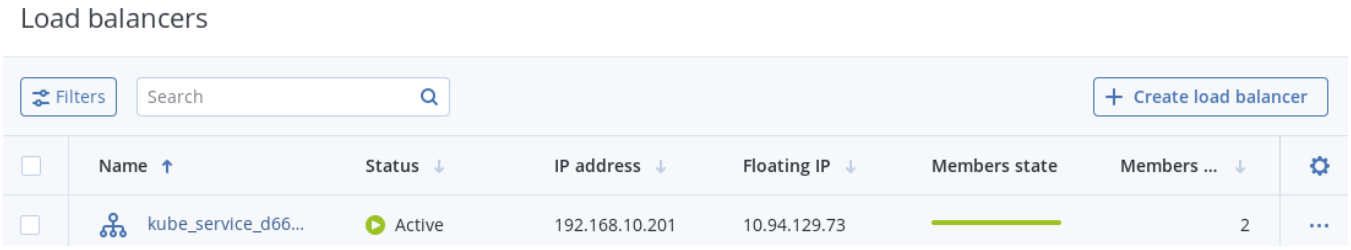
Once the load balancer is created, it will be allocated an IP address from the shared public network and can be accessed at this external endpoint.



- If you have deployed the Kubernetes cluster in a private network linked to a public 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 public network and can be accessed at this external endpoint.

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



4.3 Managing Images

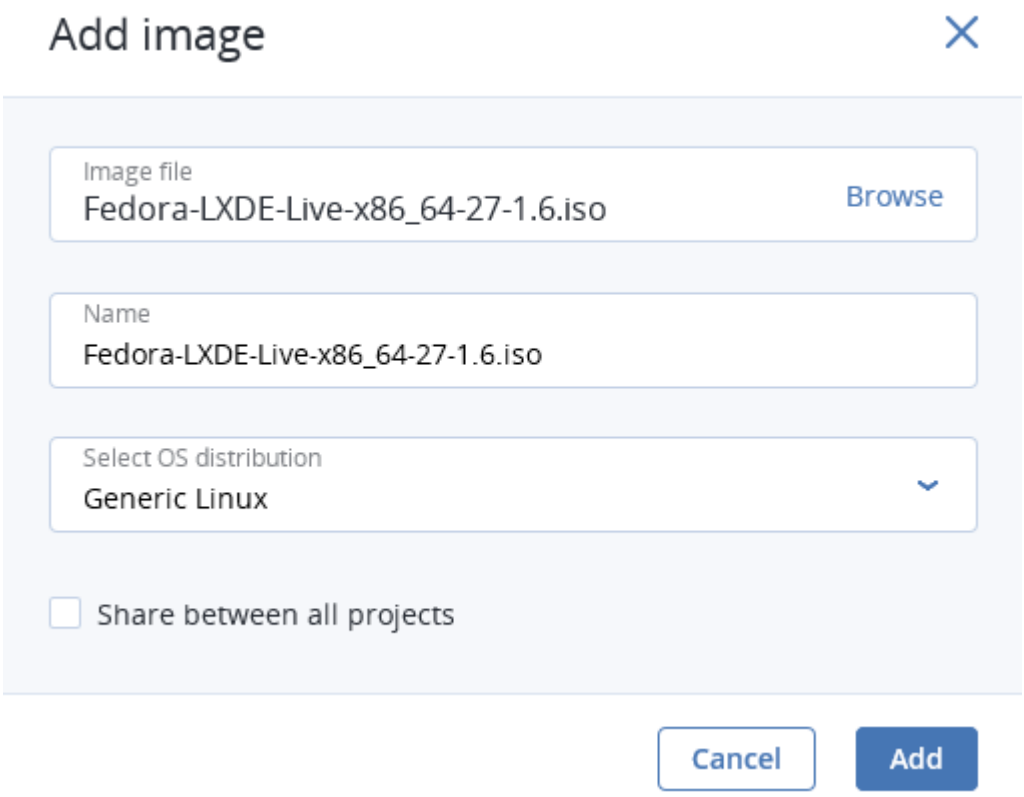
Acronis Cyber Infrastructure allows you to upload ISO images and templates that can be used to create VM volumes. An ISO image is a typical OS distribution that needs to be installed on disk. In turn, a template is a ready volume in the QCOW2 format with an installed operating system and applications and a set minimum size. Many OS vendors offer templates of their operating systems under the name “cloud images”. For a list of guest OSes supported in virtual machines, see *Supported Guest Operating Systems* (page 9).

4.3.1 Uploading and Removing Images

To add an image, do the following:

1. On the **Images** screen, click **Add image**.
2. In the **Add image** window, do the following:
   2.1. Click **Browse** and select a template or ISO file.
   2.2. Specify an image name to be shown in the admin panel.
   2.3. Select a correct OS type from the drop-down list.

> **Important:** OS type affects VM parameters like hypervisor settings. VMs created from an image with a wrong OS type may not work correctly, e.g., crash.



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

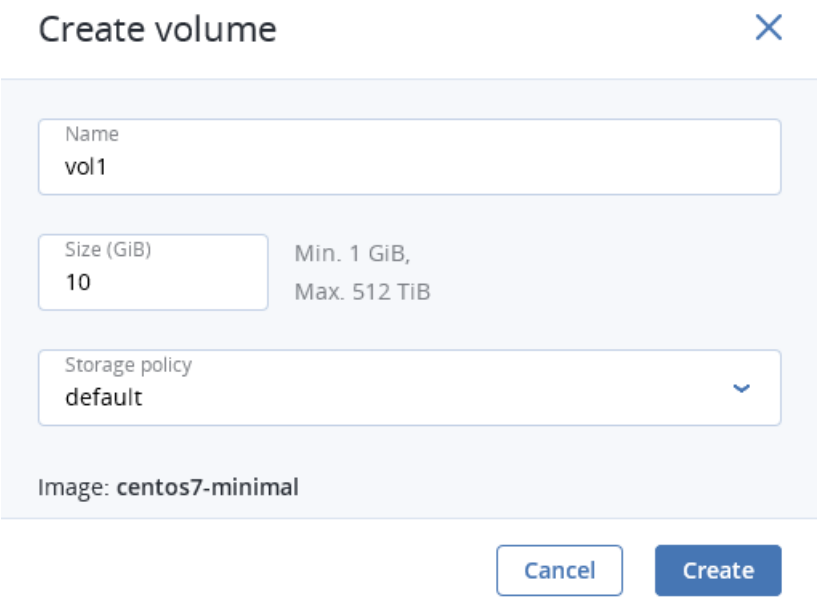
Note: If you select an image assigned to a placement, the created virtual machine will also be included in this placement. For more information, see the Administrator's Guide.

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

4.3.2 Creating Volumes from Images

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

1. On the image panel, click **Create volume**.
2. In the **Create volume** window, specify the volume name, size, and choose a storage policy.



3. Click **Create**.

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

4.3.3 Mounting ISO Images to Virtual Machines

Note: This feature is supported only for Linux virtual machines.

To mount an ISO image to a Linux VM, do as follows:

1. Create a volume from the ISO image as described in *Creating Volumes from Images* (page 33).
2. Attach the resulting volume to the desired VM as described in *Attaching and Detaching Volumes* (page 37).

The mounted disk will appear inside the Linux VM.

4.4 Managing Volumes

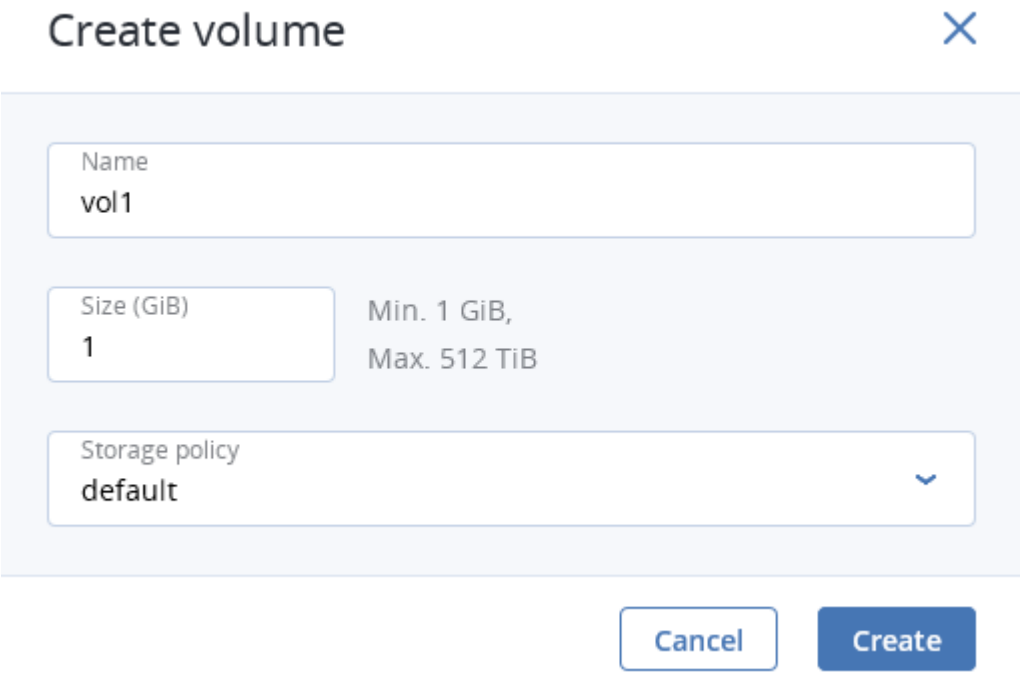
A volume in Acronis Cyber Infrastructure is a virtual disk drive that can be attached to a VM. The integrity of data in volumes is protected by the redundancy mode specified in the storage policy.

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

4.4.1 Creating, Editing, and Removing Volumes

To create a volume, do the following:

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 click **Add**.

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

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

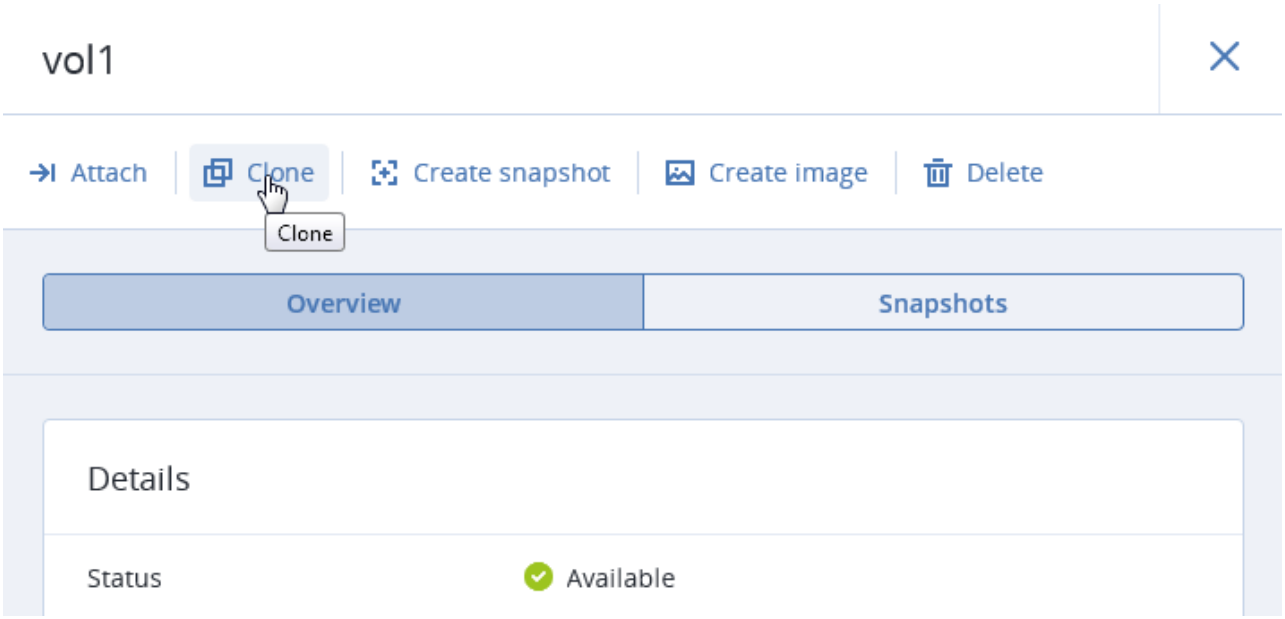
To remove a volume, click its ellipsis button then click **Delete**. To remove multiple volumes at once, select them and click **Delete**. To remove a volume that is in use, detach it first.

Note: A volume is removed along with all its snapshots.

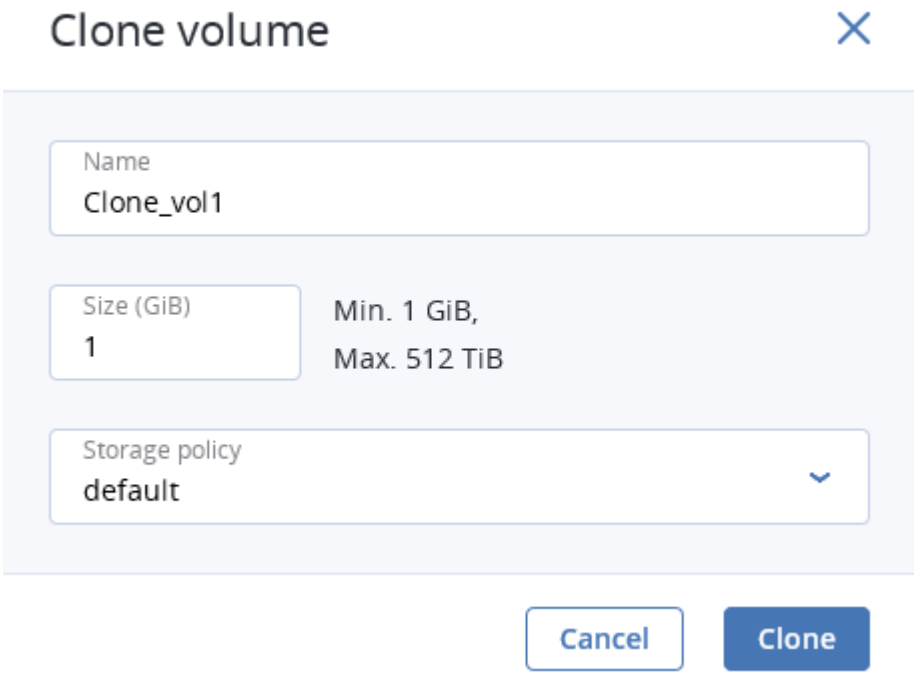
4.4.2 Cloning Volumes

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

1. On the **Volumes** screen, click a volume.
2. In volume details that opens, click **Clone**.



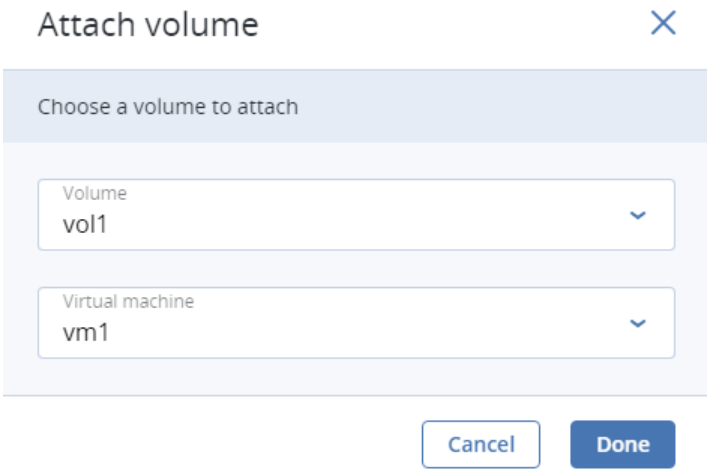
3. In the **Clone volume** window that opens, specify a volume name, size, and storage policy. Click **Clone**.



4.4.3 Attaching and Detaching Volumes

To add a writable virtual disk drive to a VM, attach a volume to it. To do this:

1. On the **Volumes** screen, click the ellipsis button next to an unused volume and click **Attach** in the context menu.
2. In the **Attach volume** window, select the VM from the drop-down list and click **Done**.



To detach a volume, do the following:

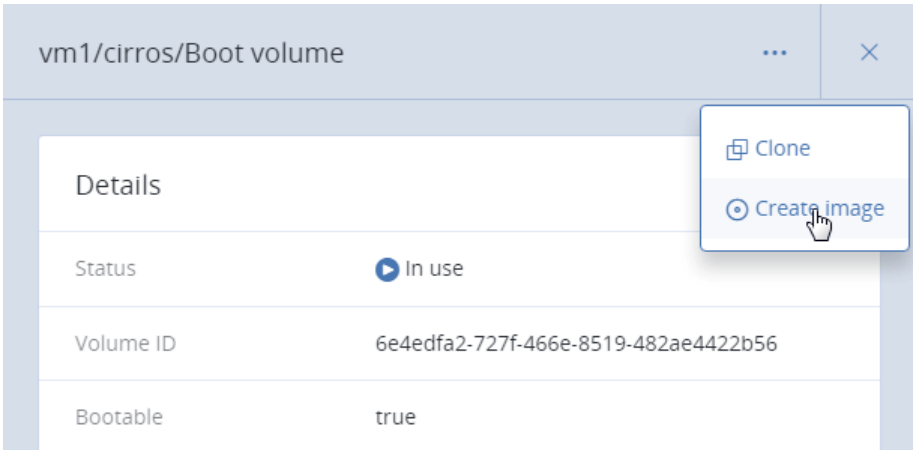
1. Click the ellipsis button next to the volume that is in use.
2. If the VM is not running, click **Detach**. If the VM is running, you can only click **Force detach** to immediately detach the volume with a risk of data loss.

4.4.4 Creating Images from Volumes

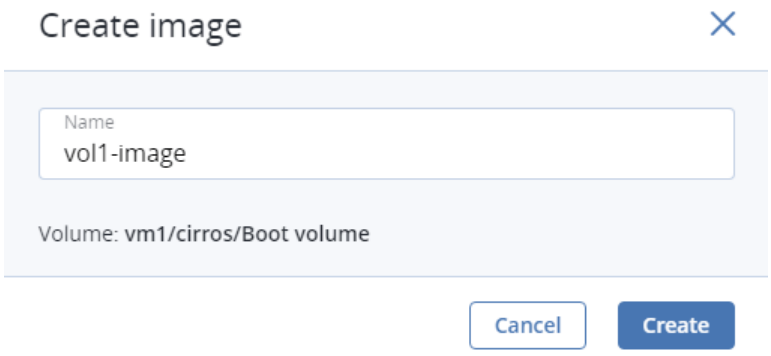
To create multiple VMs with the same boot volume, you can create an image from an existing boot volume and deploy VMs from it. Make sure to install cloud-init in the volume before creating the image.

Do the following:

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



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



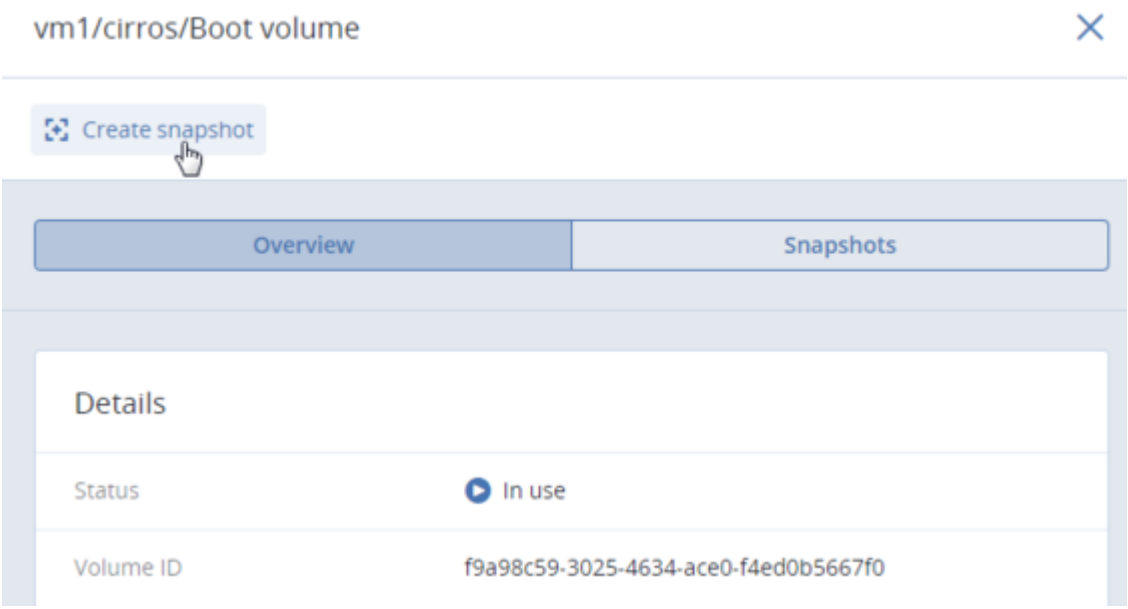
The new image will appear on the **IMAGES** tab.

4.4.5 Managing Volume Snapshots

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

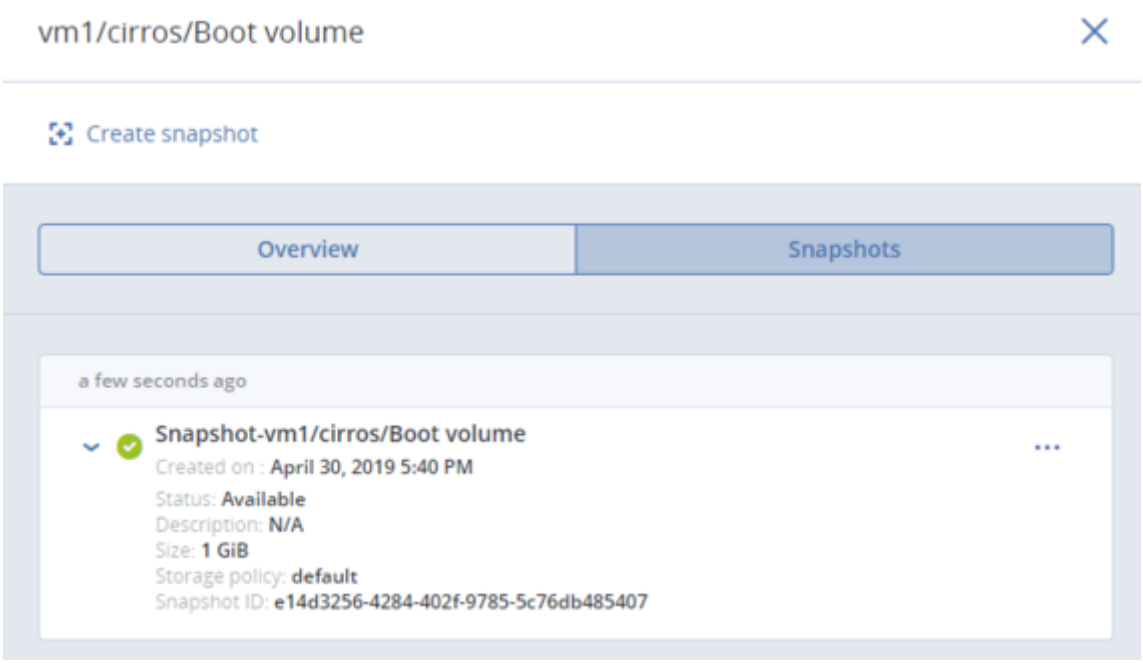
To create a snapshot of a volume, do the following:

1. On the **Volumes** screen, click a volume.
2. In the volume panel that opens, switch to **Snapshots** and click **Create snapshot**.



Note: To create a consistent snapshot of a running VM's volume, make sure the guest tools are installed in the VM. QEMU guest agent included in the guest tools image automatically quiesces the filesystem during snapshotting. For the instructions on installing the guest tools, see *Installing Guest Tools* (page 20).

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



To see the full list of available actions, click the ellipsis button next to a snapshot. Actions include:

- **Create volume** creates a new volume from the snapshot.
- **Create image** creates a template image from the snapshot.
- **Revert to snapshot** discards all changes that have been made to the volume since the snapshot was taken. This action is available only for VMs with the “Shut down” and “Shelved offloaded” statuses.

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

- **Edit** changes the snapshot name and description.
- **Reset** resets the snapshot stuck in the “Error” state or one of transitional states to the “Available” state.
- **Delete** removes the snapshot.

4.5 Managing Private Virtual Networks

To add a new virtual private network, do the following:

1. On the **Networks** screen, click **Create virtual network**.
2. In the **Network configuration** section, configure the network parameters:

   2.1. Enable or disable IP address management.

   With IP address management enabled, Acronis Cyber Infrastructure will handle virtual machine IP addresses and provide the following features:

   - Allocation pools. You can specify ranges of IP addresses that will be automatically assigned to VMs.
   - Built-in DHCP server. Assigns IP addresses to virtual machines. With the DHCP server enabled, VM network interfaces will automatically be assigned IP addresses: either from allocation pools or, if there are no pools, from network's entire IP range. With the DHCP server disabled, VM network interfaces will still get IP addresses, but you will have to manually assign them inside VMs.
   - Custom DNS servers. You can specify DNS servers that will be used by VMs. These servers will be delivered to virtual machines via the built-in DHCP server.

   With IP address management disabled:

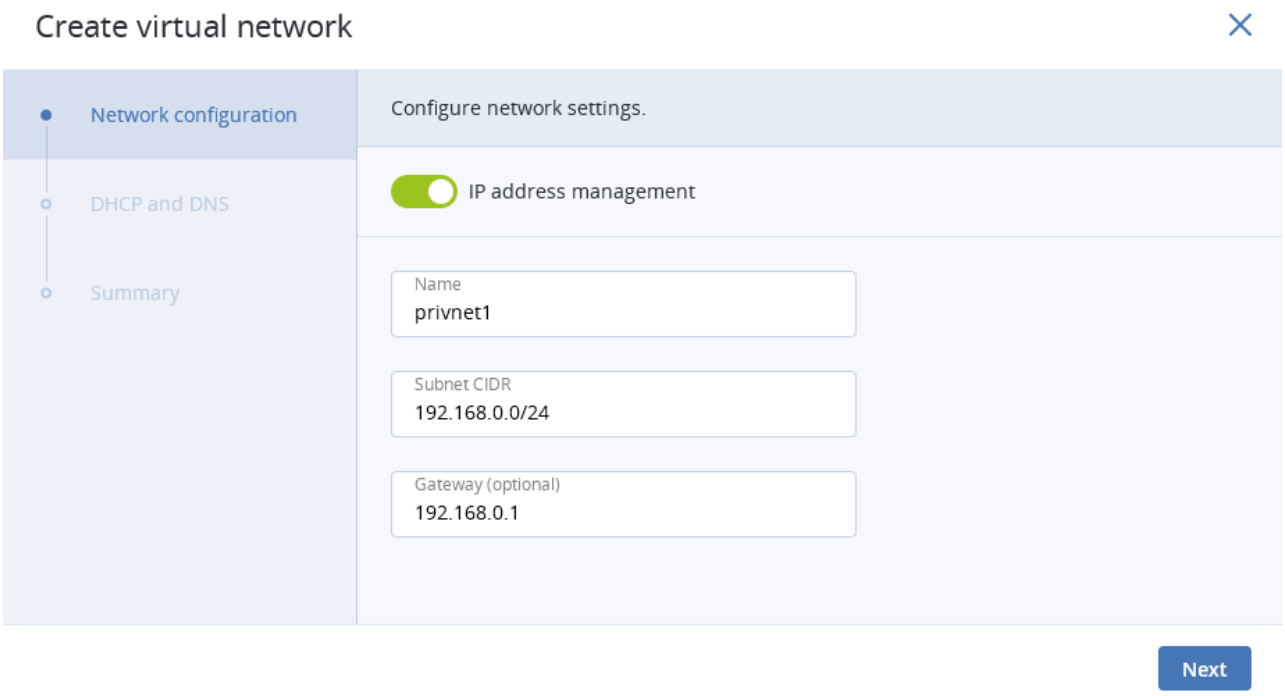
   - VMs connected to a network will be able to obtain IP addresses from DHCP servers in that network.
   - Spoofing protection will be disabled for all VM network ports. Each VM network interface will accept all traffic, even frames addressed to other network interfaces.

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

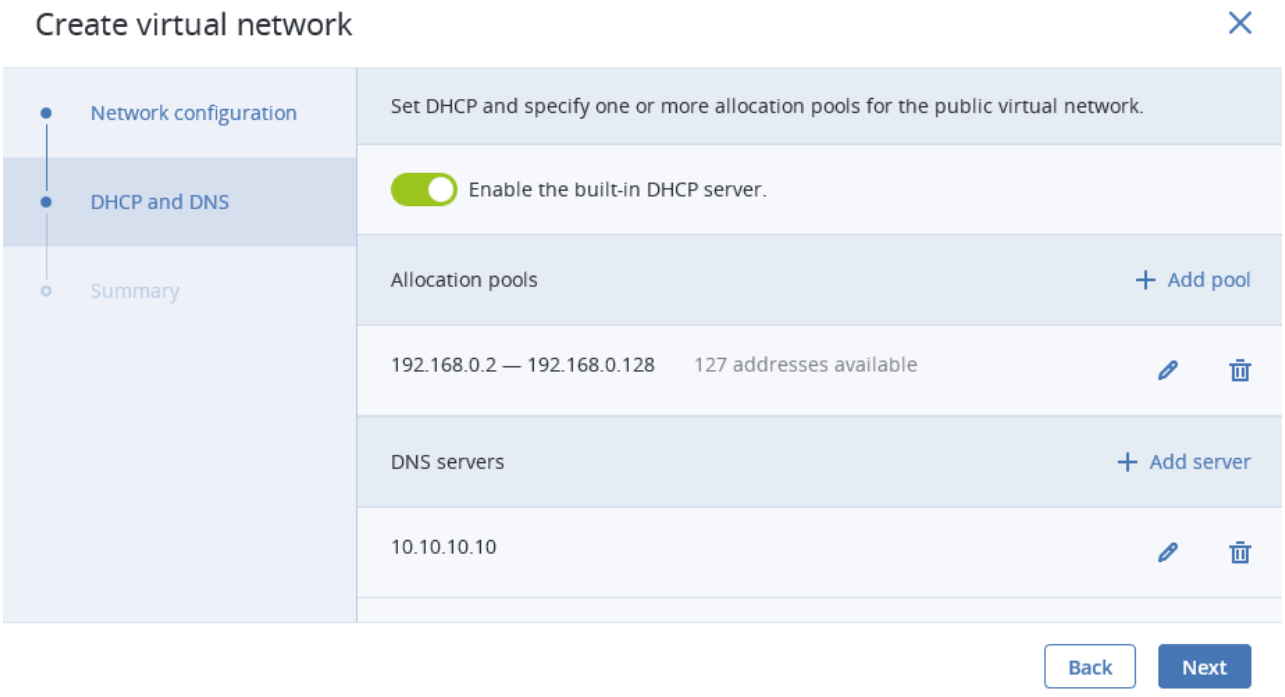
   2.2. Choose network type.

   2.3. Specify a name. If IP address management is enabled, specify network's IPv4 address range in **Subnet CIDR**. Optionally specify a gateway. If you leave the **Gateway** field blank, the gateway will be omitted from network settings.

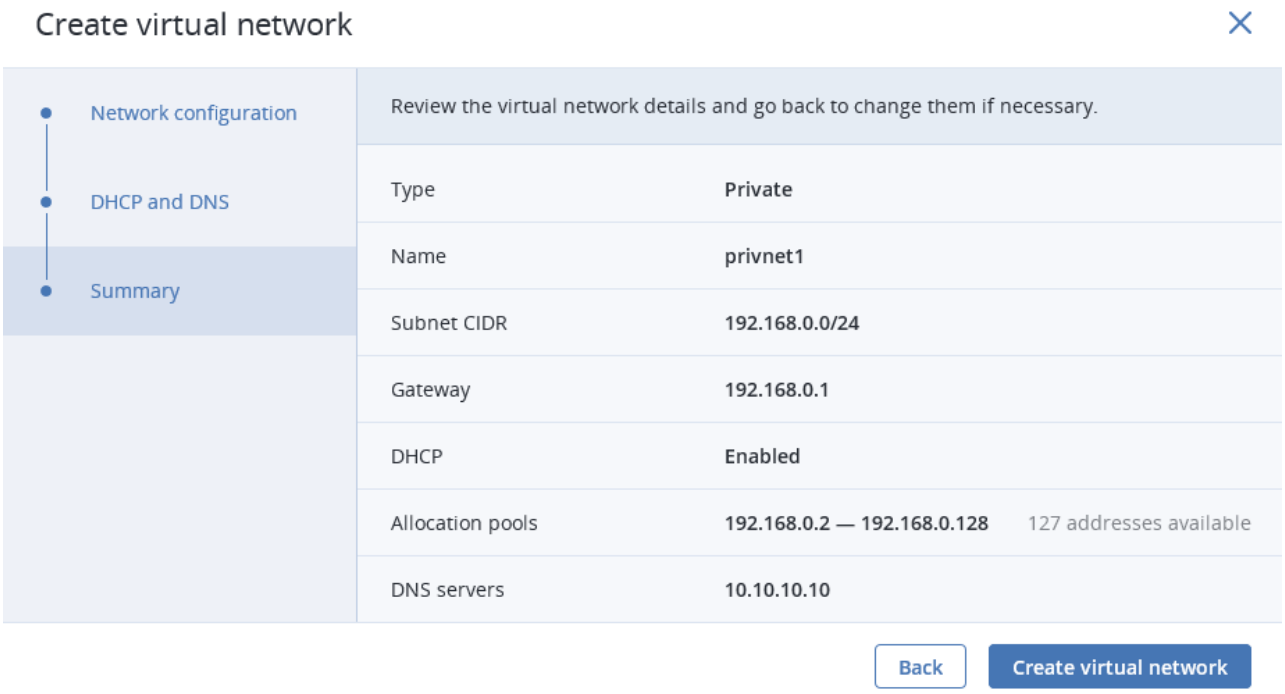
   Click **Next**.



3. If you enabled IP address management on the previous step, you will move on to the **DHCP and DNS** section. In it, enable or disable the built-in DHCP server and specify one or more allocation pools and DNS servers. Click **Next**.



4. In the **Summary** section, review the configuration and click **Create virtual network**.



To view and edit parameters of a virtual network, click it on the **Networks** screen. On the virtual network panel, you can change the virtual network name, gateway, DHCP settings, allocation pools, and DNS servers. To do this, click the pencil icon, enter a new value, and click the check mark icon to confirm.

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

4.6 Managing Virtual Routers

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

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

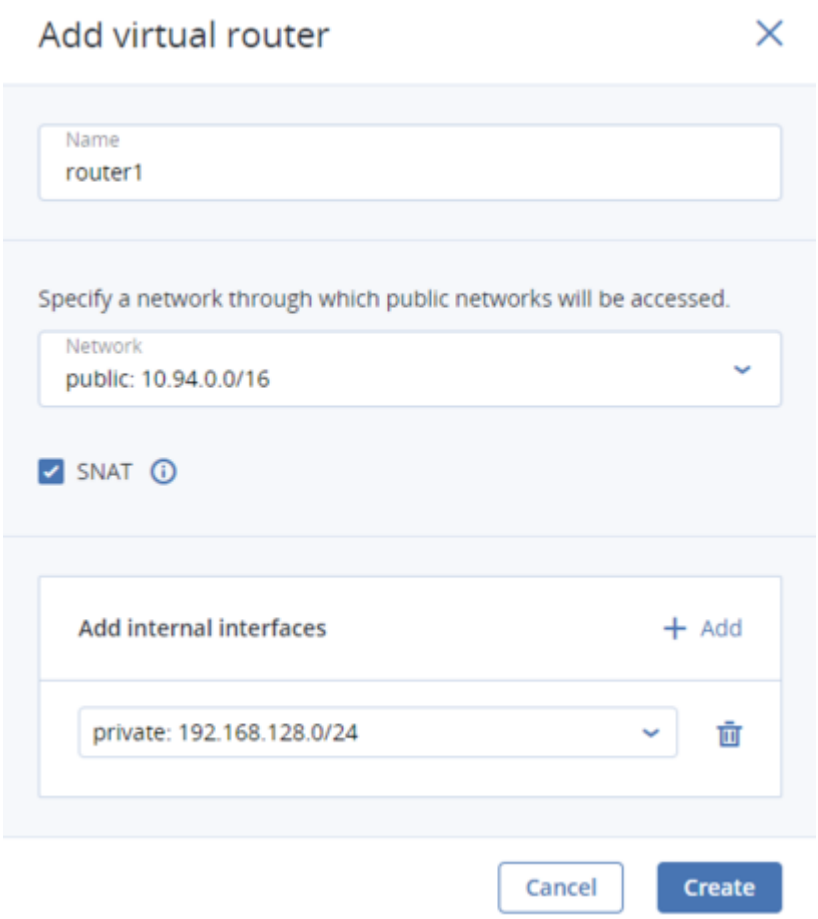
A virtual router has two types of ports:

- an external gateway that is connected to a public network,
- an internal port that is connected to a private network.

Note: A router can only connect networks with enabled IP management.

To create a virtual router, do the following:

1. On the **COMPUTE** > **Networks** > **NETWORKS** tab, make sure the virtual networks that are to be connected to a router have a gateway specified.
2. Navigate to the **COMPUTE** > **Routers** tab and click **Add router**.
3. In the **Add router** window:
   3.1. Specify a router name.
   3.2. From the **Network** drop-down menu, select a public network through which external access will be provided via an external gateway. The new external gateway will pick an unused IP address from the selected public network.
   3.3. In the **Add internal interfaces** section, select one or more private networks to connect to a router via internal interfaces. The new internal interfaces will attempt to use the gateway IP address of the selected private networks by default.
   3.4. Optionally, select or deselect the **SNAT** checkbox to enable or disable SNAT, respectively, 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.



4. Click **Create**.

To edit a router name, click the ellipsis icon next to it and **Rename**.

To remove a virtual router, click the ellipsis icon next to it and **Delete**. To remove multiple virtual networks at once, select them and click **Delete**. Before deleting a virtual router, make sure no floating IP addresses are associated with any network it is connected to.

4.6.1 Managing Router Interfaces

You can add an external router interface as follows:

Note: To change an external gateway, remove the existing one first.

1. On **Routers** screen, click the router name to open the list of its interfaces.

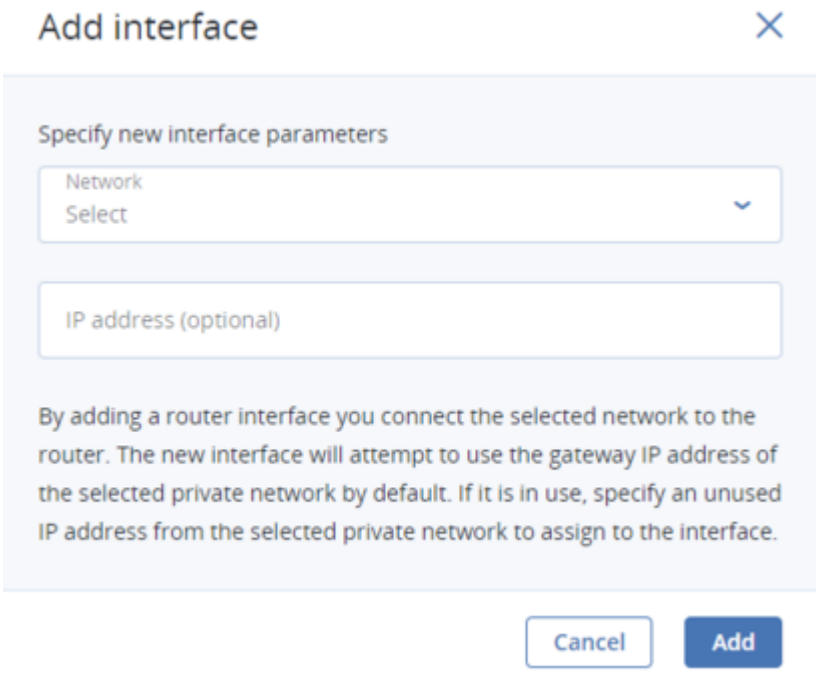
2. Click **Add** on the toolbar, or click **Add interface** if there are no interfaces to show.
3. In the **Add interface** window, do the following:
   3.1. Choose **External gateway**.
   3.2. From the **Network** drop-down menu, select a public network to connect to the router. The new interface will pick an unused IP address from the selected public network. You can also provide a specific IP address from the selected public network to assign to the interface in the **IP address** field.
   3.3. Optionally, select or deselect the **SNAT** checkbox to enable or disable SNAT, respectively, 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.

4. Click **Add**.

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

You can add an internal router interface as follows:

1. On **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 private network by default. If it is in use, specify an unused IP address from the selected private network to assign to the interface in the **IP address** field.



4. Click **Add**.

To remove a router interface, click the ellipsis icon next to it and **Delete**. To remove multiple interfaces at once, select them and click **Delete**.

4.6.2 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 private networks and want only one private network to be accessible from the other.

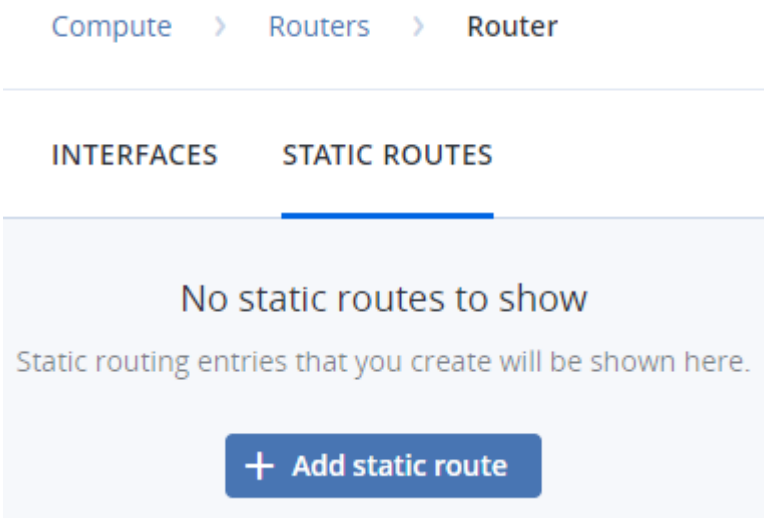
Consider the following example:

- the virtual machine `vm1` is connected to the private 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 private 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 public network via the external gateway with the IP address `10.94.129.73`,
- the router `router2` connects the network `private2` to the public network via the external gateway with the IP address `10.94.129.74`.

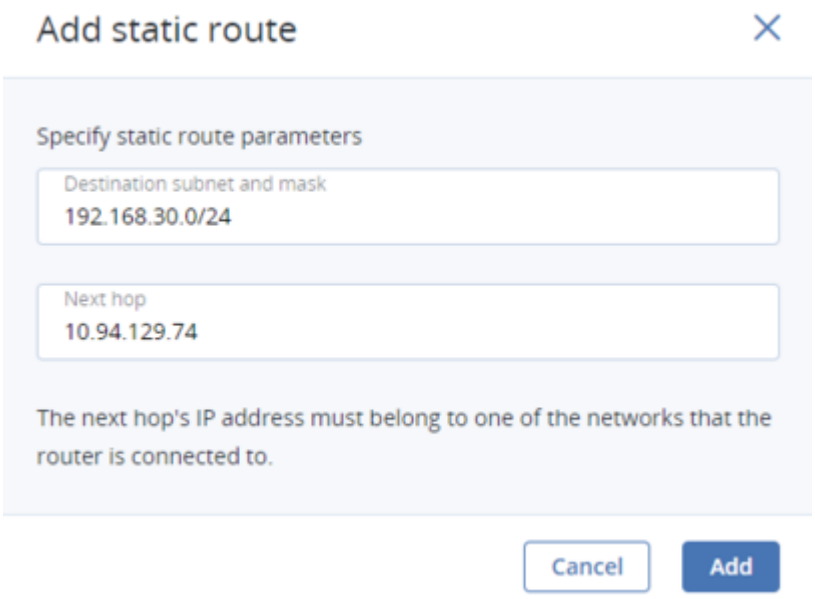
To be able to access `vm2` from `vm1`, you need to add a static route for `router1`, specifying the CIDR of `private2`, that is `192.168.30.0/24`, as the destination subnet and the external gateway IP address of `router2`, that is `10.94.129.74`, as the next hop IP address. In this case, when an IP packet for `192.168.30.10` reaches `router1`, it will be forwarded to `router2` and then to `vm2`.

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

1. On the **Routers** screen, click the router name. Open the **STATIC ROUTES** tab and click **Add** on the toolbar. Or click **Add static route** if there are no routes to show.



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**.

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

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

4.7 Managing Floating IP Addresses

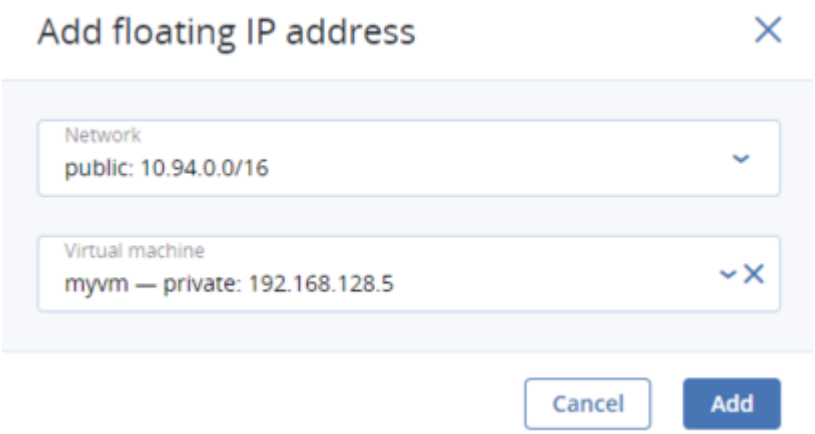
A virtual machine connected to a virtual private network can be accessed from public networks, such as the Internet, by means of a floating IP address. Such an address is picked from a public network and mapped to VM's private IP address. The floating and private IP addresses are used at the same time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

Note the following prerequisites:

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

You can create a floating IP address and assign it to a VM as follows:

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



3. Click **Add**.

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

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

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

4.8 Managing Load Balancers

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

Note the following requirements:

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

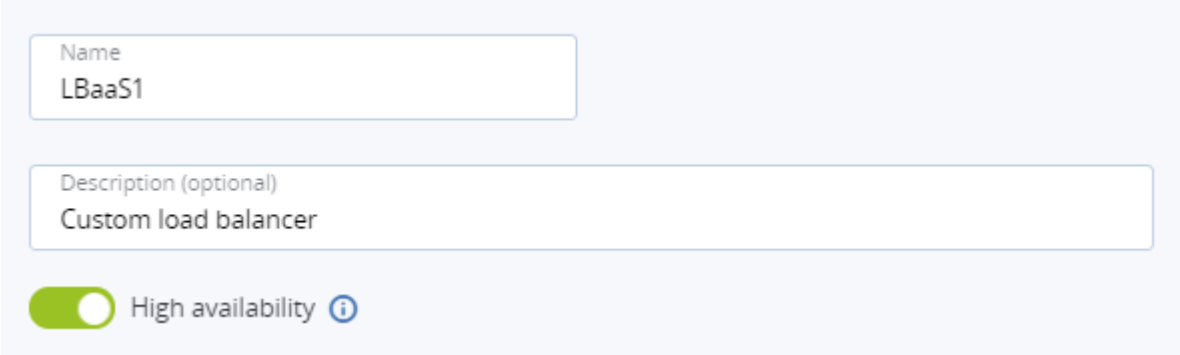
You can create a load balancer with balancing pools as follows:

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

   2.1. Specify a name and optionally description.

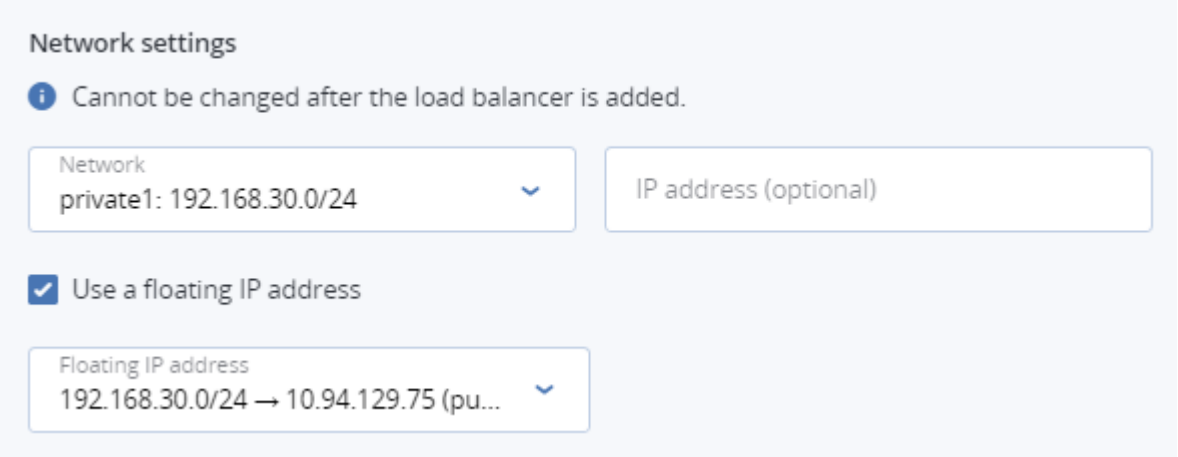
   2.2. Enable or disable high availability:

   - With high availability enabled, two load balancer instances will be created. They will work in the Active/Standby mode according to the Virtual Router Redundancy Protocol (VRRP).
   - With high availability disabled, a single load balancer instance will be created.



   2.1. In the **Network settings** section, select the network that the load balancer will operate in and optionally specify an IP address that will be allocated to the load balancer.

   If a chosen private network is connected to a public network via a router, you can assign a floating IP address to the load balancer. To do it, select **Use a floating IP address**, and from the drop-down menu that appears, choose either to use an available floating IP address or to create a new one.



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

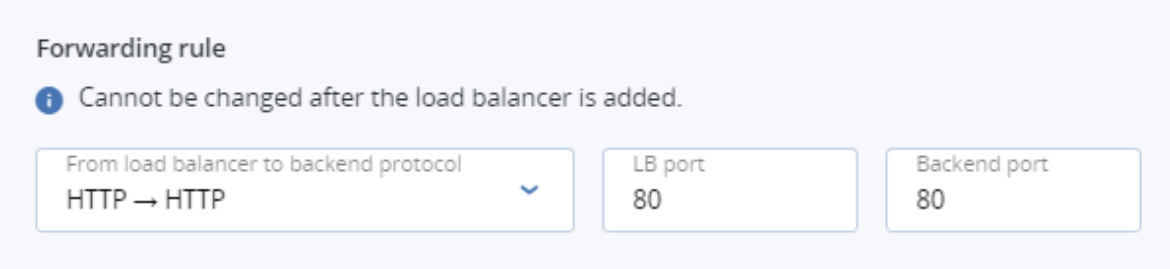
In the **Create balancing pool** window that opens, do the following:

2.1.1. In the **Forwarding rule** section, select a forwarding rule from the load balancer to the backend protocol: **HTTPS -> HTTPS**, **HTTPS -> HTTP**, **HTTP -> HTTP**, or **TCP -> TCP**. Additionally, specify the ports for incoming and destination connections in the **LB port** and **Backend port** fields.

Note the following:

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

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

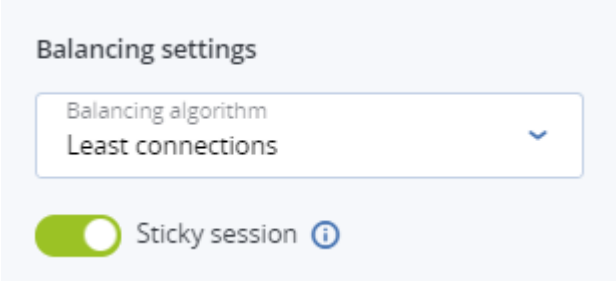


2.1.2. In the **Balancing settings** section, choose a balancing algorithm:

- **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.

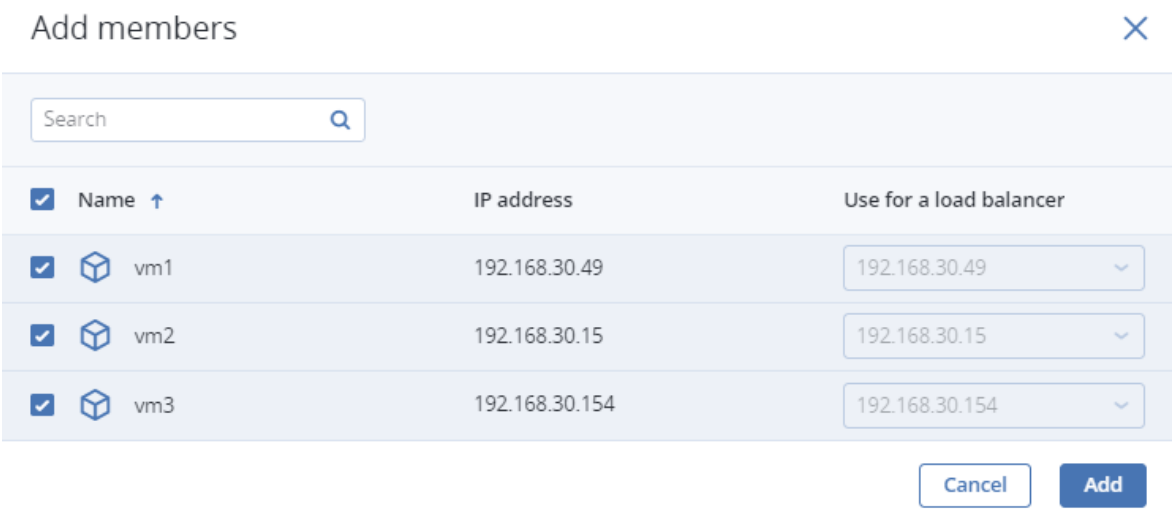
Enable/disable the **Sticky session** option to enable/disable session persistence. The load balancer will generate a cookie that will be inserted into each response. The cookie will be used to send future requests to the same VM.

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



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

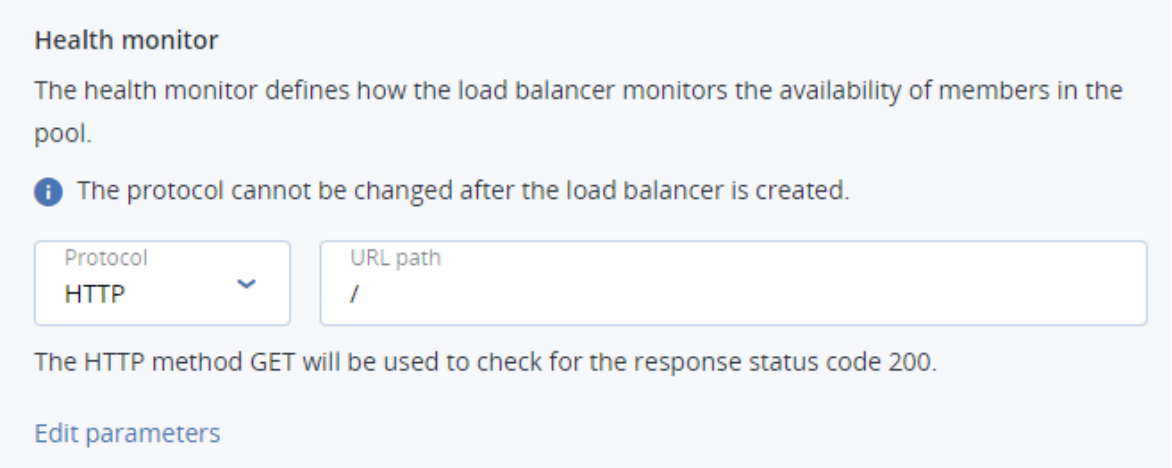
In the **Add members** window that opens, select desired VMs and click **Add**.



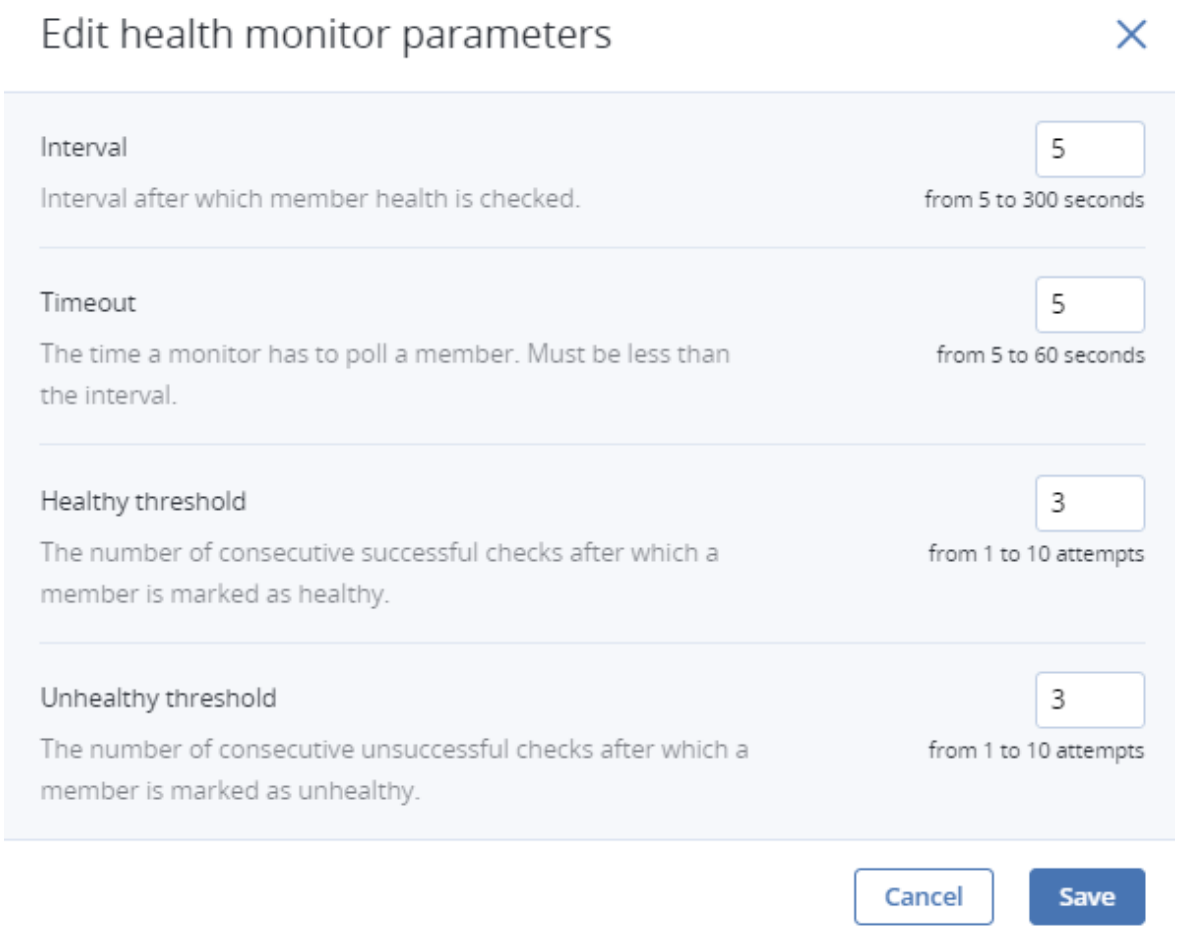
2.1.4. In the **Health monitor** section, choose the protocol that will be used for monitoring members availability:

- **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**. The health monitor will check the TCP connection on the backend port.
- **PING**. The health monitor will check members' IP addresses.

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



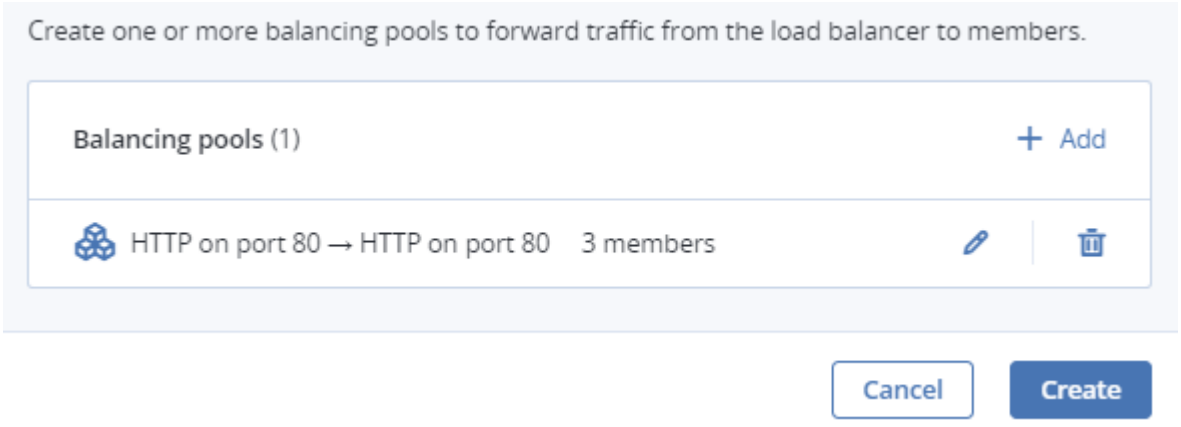
By default, the health monitor removes a member from a balancing pool if it fails three consecutive health checks of five-second intervals. When a member returns to operation and responds successfully to three consecutive health checks, it is added to the pool again. You can manually set the health monitor 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 click **Save**.



2.1.5. Click **Create**.

2.2. Add more balancing pools as described above, if required.

2.3. Click **Create**.



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

The following charts are available:

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

CPU/RAM
: CPU and RAM usage by the load balancer.

Network
: Incoming and outgoing network traffic.

Active connections
: The number of active connections.

Error requests
: The number of error requests.

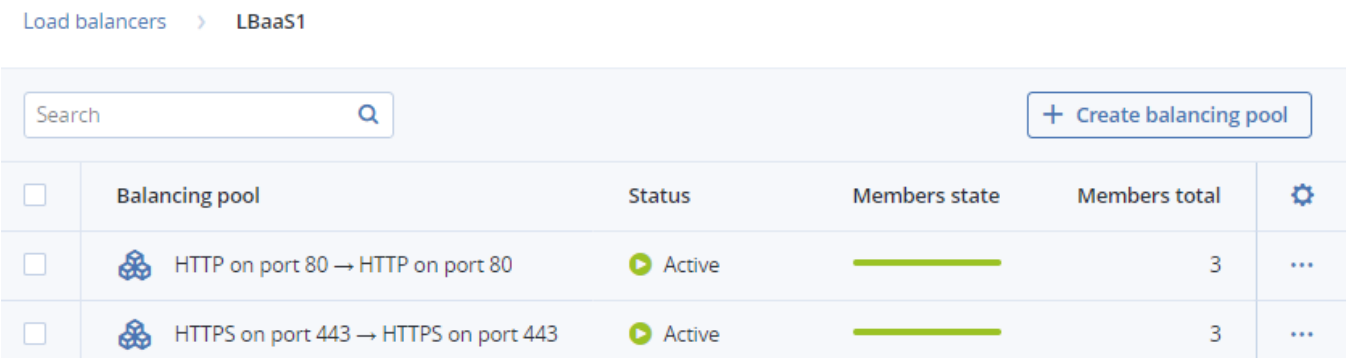
You can see the load balancer parameters on its **Properties** tab.

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

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

4.8.1 Managing Balancing Pools

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



To add another balancing pool to a load balancer, click **Create balancing pool** and fill in the fields as described in *Managing Load Balancers* (page 50). The newly added pool will appear in the list of balancing pools.

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

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

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

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

4.9 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 Acronis Cyber Infrastructure and specified during VM creation. It will be injected into the VM by `cloud-init` and used for OpenSSH authentication. Keys injection is supported for both Linux and Windows virtual machines.

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

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

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

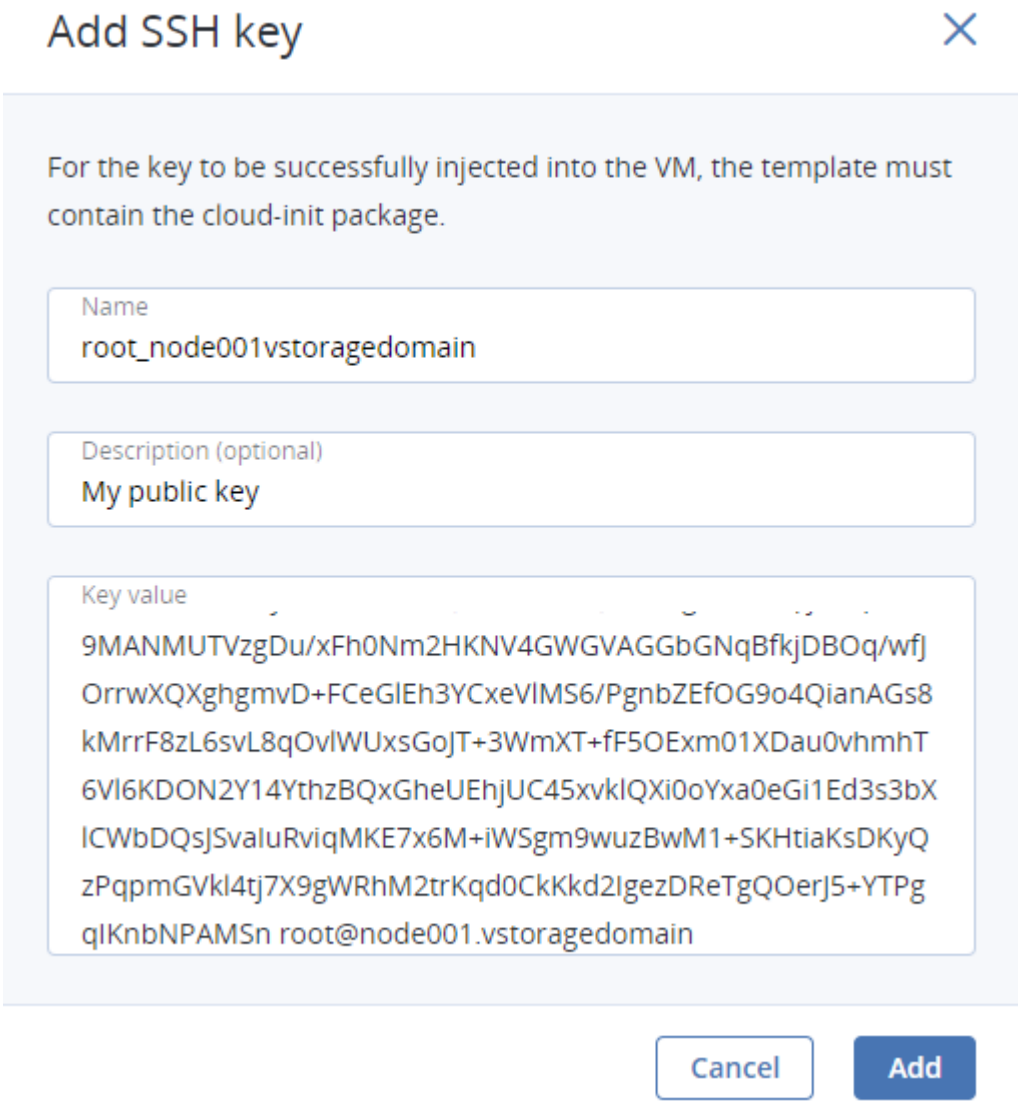
For the instructions on preparing templates or boot volumes, see Preparing Templates.

To add a public key, do the following:

1. Generate an SSH key pair on a client 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.



To delete one or more keys, select them and click **Delete**.

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