



Virtuozzo 7 Beta Homework

July 25, 2016

Parallels IP Holdings GmbH

Vordergasse 59

8200 Schaffhausen

Switzerland

Tel: + 41 52 632 0411

Fax: + 41 52 672 2010

<http://www.virtuozzo.com>

Copyright © 1999-2016 Parallels IP Holdings GmbH and its affiliates. All rights reserved.

This product is protected by United States and international copyright laws. The product's underlying technology, patents, and trademarks are listed at <http://www.virtuozzo.com/legal/>.

Microsoft, Windows, Windows Server, Windows NT, Windows Vista, and MS-DOS are registered trademarks of Microsoft Corporation.

Apple, Mac, the Mac logo, Mac OS, iPad, iPhone, iPod touch, FaceTime HD camera and iSight are trademarks of Apple Inc., registered in the US and other countries.

Linux is a registered trademark of Linus Torvalds.

All other marks and names mentioned herein may be trademarks of their respective owners.

Table of Contents

1. Overview	4
2. Installing Virtioozzo 7 Beta	5
2.1. Virtioozzo 7 Beta System Requirements	5
2.2. Installing Virtioozzo 7 Beta On Bare Metal	5
2.3. Installing Virtioozzo 7 Beta In Virtual Machines	5
3. Guest Operating Systems Supported by Virtioozzo 7 Beta	6
4. Use Cases	7
4.1. Creating New Virtual Machines	7
4.1.1. Creating a Virtual Machine from a Template	7
4.1.2. Creating a Virtual Machine from an ISO Image	7
4.2. Installing Virtioozzo Guest Tools	8
4.3. Checking Virtual Machine Hypervisor Type	9
4.4. Using Online Memory Management for Virtual Machines	9
4.5. Setting the Memory Guarantee for a Virtual Machine	10
4.6. Setting the Memory Guarantee for a Container	10
4.7. Managing Virtioozzo 7 Containers and Virtual Machines with libvirt	11
4.8. Managing Virtioozzo 7 Virtual Machines with virsh	11
4.9. Managing Virtioozzo 7 Virtual Machines with virt-manager	12
4.9.1. Installing virt-manager	12
4.9.2. Using virt-manager	13
4.10. Working with Microsoft Windows Virtual Machines	18

Chapter 1. Overview

Some years ago we introduced our own hypervisor technology in Virtuozzo in addition to our containers. Such an offering added more flexibility to Virtuozzo, as in some cases customers preferred traditional hypervisor solutions over containers. Fast and reliable, Virtuozzo hypervisor offered a lot of enterprise features answering the needs of service providers.

Nowadays KVM has become fast and reliable enough to suite our customers' needs. For this reason, we do not need to develop and support our own hypervisor anymore. Yet KVM still lacks many required features and we are eager to implement such functionality in our new Virtuozzo 7 with KVM-based virtualization.

The two technical previews of Virtuozzo 7 we released before were focused mostly on container technology, whereas this beta is focused on KVM-based virtual machines.

Key changes in Virtuozzo 7 Beta are:

- Unified management of containers and virtual machines,
- Transition from own hypervisor to KVM,
- Online memory management for containers and VMs,
- Memory guarantees for containers and VMs,
- Features enabling VM memory overcommitment: memory autoballooning and kernel same-page merging (KSM).
- Ability to manage containers and VMs with libvirt and virt-manager,
- Guest tools for Windows and Linux VMs.

We are still working to improve the memory overcommitment mechanism to allow the commercial customers to increase density with a minimal impact on performance for end-users.

The main goal of Virtuozzo 7 Beta is to assure of our customers of the complete CLI and SDK compatibility between Virtuozzo 6 and Virtuozzo 7 that allows managing containers and new KVM-based virtual machines.

In this homework you will walk through the new Virtuozzo 7 features using a container and a virtual machine with a CentOS 7 distribution.

Chapter 2. Installing Virtuozzo 7 Beta

The Virtuozzo 7 installer is based on the RHEL7 installer, so installation process is simple and straightforward. To install Virtuozzo 7 Beta, you will need the distribution ISO image available at http://updates.virtuozzo.com/virtuozzo/releases/7.0-beta/x86_64/iso/VZ-7.0-beta-x86_64-DVD.iso.

You can install Virtuozzo directly on bare metal or in a virtual machine on a system with nested virtualization support (supported hypervisors are Parallels Desktop for Mac, VMware Fusion, VMware Workstation, and VMware ESXi). Make sure that nested virtualization support is enabled in your hypervisor.

2.1. Virtuozzo 7 Beta System Requirements

The following configuration is recommended for this homework:

- At least 4 CPU cores;
- At least 4GB of RAM;
- At least 64GB disk space.

2.2. Installing Virtuozzo 7 Beta On Bare Metal

To install Virtuozzo on bare hardware, burn the distribution ISO image onto a USB stick or CD-ROM and boot your hardware node from it. Then follow wizard's instructions to install Virtuozzo 7 Beta.

2.3. Installing Virtuozzo 7 Beta In Virtual Machines

To install Virtuozzo in a VM, copy the distribution ISO image to a local drive and create a new VM from it according to your virtualization software documentation. Start the VM, boot to the Virtuozzo installer, and follow the wizard's instructions to install Virtuozzo 7 Beta.

Chapter 3. Guest Operating Systems Supported by Virtiozzo 7 Beta

Virtiozzo 7 Beta supports the following guest operating systems:

Virtual Machines

- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2 with Service Pack 1
- CentOS 7.x (x64)
- CentOS 6.x (x64)
- Debian 8.x (x64)
- Ubuntu 14.04 LTS (x64)

Containers

- CentOS 7.x (x64)
- CentOS 6.x (x64)
- Debian 8.x (x64)
- Ubuntu 14.04 LTS (x64)

Chapter 4. Use Cases

4.1. Creating New Virtual Machines

4.1.1. Creating a Virtual Machine from a Template

For your convenience, we have prepared a custom VM template `myvm` that you can use in this homework. The template is based on CentOS 7 x64 Minimal Install and includes some additional packages.

To create a VM from the template, do the following:

1. Download the template from http://updates.cloudserver.virtuozzo.com/server/pcs/en_us/virtuozzo/7/beta/virtuozzo7-sample-vm.tar.gz.
2. Unpack the template to a VM folder:

```
# tar -xvf virtuozzo7-sample-vm.tar.gz -C /vz/vmprivate
```

3. Register the new template:

```
# prlctl register /vz/vmprivate/myvm.pvm
```

4. Check that your new VM can be listed:

```
# prlctl list -a
```

5. Start your new VM:

```
# prlctl start myvm
```

The default

- root password is `Virtuozzo1!`,
- VNC port is 5901, and
- VNC password is `12345678`.

4.1.2. Creating a Virtual Machine from an ISO Image

To create a new VM `myvm` from scratch, do the following:

1. Create a blank VM and set its RAM to 1GB:

```
# prlctl create myvm --distribution centos
# prlctl set myvm --memsize 1024
```

2. Enable VNC console access:

```
# prlctl set myvm --vnc-mode manual --vnc-port <port> --vnc-passwd <password>
```

3. Download the CentOS image from http://isoredirect.centos.org/centos/7/isos/x86_64/CentOS-7-x86_64-Minimal-1511.iso.

4. Connect the image to the VM:

```
# prlctl set myvm --device-set cdrom0 --image <path_to_image>
```

5. Start the VM:

```
# prlctl start myvm
```

6. Access the VNC console with a VNC viewer.

7. Install the OS.

The virtual machine is ready. Proceed to install the guest tools as described in the next section.

4.2. Installing Virtio Guest Tools

Virtio 7 ships with the guest tools for Linux and Windows virtual machines. The guest tools allow you to:

- Configure virtual machine and container network,
- Set passwords,
- Run commands inside containers or VMs from the host,
- Enter containers or VMs,
- Change SIDs,
- Perform graceful shutdowns,
- Manage devices.

In addition, Virtio guest tools also contain the autoballoon driver.

To install guest tools inside a VM `myvm`, do the following:

1. Mount the guest tools image to the VM:

```
# prlctl installtools myvm
```

2. Connect to the VM either via VNC (as described previously in this homework) or SSH if your VM already has an IP address set manually or via DHCP:

```
# ssh root@<VM_IP_addr>
```

3. Run the guest tools installer inside the VM:

```
# mkdir /mnt/cdrom
# mount /dev/cdrom /mnt/cdrom
# /mnt/cdrom/install
```


The guest tools are installed.

4.3. Checking Virtual Machine Hypervisor Type

To check that the new virtual machine (`myvm`) is KVM-based, run

```
# prlctl exec myvm dmidecode -s system-product-name
```

The result must be `KVM`.

4.4. Using Online Memory Management for Virtual Machines

Online memory management is a new feature of Virtuozzo 7, based on memory hotplugging and ballooning, that allows increasing or reducing VM memory on the fly, without rebooting the VM. OOM enables the following scenarios:

- Pay-As-You-Grow (PAYG), customers can scale VM vertically according to workload;
- Changing billing plans without reboot.

The feature can only be enabled for stopped virtual machines with at least 1GB of RAM. To enable online memory management, run:

```
# prlctl stop myvm
# prlctl set myvm --mem-hotplug on
# prlctl start myvm
```

To check that the feature works:

1. Wait until VM is fully booted and the balloon driver is loaded, then run:

```
# virsh dominfo myvm
```

Used Memory should be 1GB while **Max Memory** should be 5GB. The balloon size should be about 7GB now.

2. Now increase memory size to 1.5GB:

```
# prlctl set myvm --memsize 1536
```

3. Check the VM memory:

```
# virsh dominfo myvm
```

Used Memory should be 1.5GB.

4. Reduce the amount of VM memory to 1GB:

```
# prlctl set myvm --memsize 1024
```

5. Check the VM memory again:

```
# virsh dominfo myvm
```

4.5. Setting the Memory Guarantee for a Virtual Machine

In Virtuozzo 7, the `--memquota` option of the `prlctl` command has been dropped. Now to set the minimum amount of memory available for a container or VM, the `prlctl --memguarantee` command is used. The default guarantees are 40% for virtual machines and 0% for containers.

To set a memory guarantee (in percentage of virtual machine's configured memory), run:

```
# prlctl set myvm --memguarantee 80
```

In this example, the memory guarantee is set to 80% of virtual machine's RAM.

To revert the memory guarantee value to default, run:

```
# prlctl set myvm --memguarantee auto
```

To guarantee an application inside a VM permanent access to all configured memory with no dependency on host memory pressure, set a memory guarantee for it.

4.6. Setting the Memory Guarantee for a Container

Memory guarantee is a new feature for Virtuozzo Containers. Now you can guarantee your customers that their containers will have the set minimal amount of RAM.

Memory guarantees enable you to control the overcommitment ratio in your datacenter. For example, if you need an overcommitment ratio of 2:1, set the memory guarantees of all containers and VMs to 50%.

Efficient overcommitment allows you to lower your datacenter costs and keep prices competitive. Customers usually do not use all of their purchased resources, and our predictive algorithm provides these resources to customers at once. To customers that do need all of their purchased resources all the time, you can offer containers with memory guarantees set to 100%, that is without overcommitment.

1. For this example, make sure all other containers or virtual machines on the host are stopped.
2. Create two containers as follows:

```
# prlctl create myct --vmtype ct
# prlctl create workload --vmtype ct
```

3. Set memory sizes for the containers. The workload is meant to have nearly as much RAM as your host has. For example, host RAM minus 2GB.

```
# prlctl set myct --memsize 3G
# prlctl set workload --memsize <host_RAM_minus_2GB>
```

4. Set the memory guarantees for the containers:

```
# prlctl set myct --memguarantee 100
# prlctl set workload --memguarantee auto
```

5. Start the workload container:

```
# prlctl start workload
```

6. Try to start `myct`:

```
# prlctl start myct
```

It should start successfully, because `workload` has no workload inside or any reserved memory (as the default guarantee is 0%).

7. Stop `myct` and change the memory guarantee for `workload`:

```
# prlctl stop myct
# prlctl set workload --memguarantee 100
```

8. Try to start `myct` again:

```
# prlctl start myct
```

It should not start. The reason is `workload` is guaranteed all of its configured memory now (host RAM minus 1GB), and `myct` is guaranteed 3GB, so the sum of guarantees exceeds host physical RAM.

To start `myct`, you will need to stop `workload` first. To start both containers, you will need to adjust their guarantees so that their total does not exceed host physical RAM.

4.7. Managing Virtuozzo 7 Containers and Virtual Machines with libvirt

Virtuozzo 7 allows you to manage containers and VMs via the libvirt API. The supported API calls for Virtuozzo are described in the column **vz** at <http://libvirt.org/hvsupport.html>.

To manage Virtuozzo containers and VMs via libvirt, you must use our libvirt driver with the `vz:///system` connection option.

4.8. Managing Virtuozzo 7 Virtual Machines with virsh

Virsh is a default utility for managing KVM-based virtual machines. As both containers and virtual machines in Virtuozzo 7 are managed via the libvirt API, virsh works too.

Here is how you use virsh to perform basic tasks with Virtuozzo 7 virtual machines:

1. Set the default connection URI:

```
# export LIBVIRT_DEFAULT_URI="vz:///system"
```

- List all the instances:

```
# virsh list --all
```

Virsh should show all instances, including virtual machines and containers.

- Start the VM:

```
# virsh start myvm
```

- Start the container:

```
# virsh start myct
```

- Output container configuration:

```
# virsh dominfo myct
```

- Stop the VM:

```
# virsh destroy myvm
```

- Stop the container:

```
# virsh destroy myct
```

- To check that both `myvm` and `myct` are deleted, list the instances again:

```
# virsh list --all
```

For the complete list of supported options, visit <https://libvirt.org/hvsupport.html>.

4.9. Managing Virtuozzo 7 Virtual Machines with virt-manager

Once again, as Virtuozzo 7 containers and VMs are managed via the libvirt API, you can use Virtual Machine Manager (virt-manager) for the same purpose. You will need to install our customized version of virt-manager that supports any Red-Hat-based OS. We tested our version of virt-manager with CentOS 7.

4.9.1. Installing virt-manager

To prepare for managing Virtuozzo 7 containers and VMs with virt-manager, do the following:

- Add the Virtuozzo repository to your OS by creating `/etc/yum.repos.d/virtuozzo.repo` with the following contents:

```
[vz]
name=vz
mirrorlist=http://download.openvz.org/virtuozzo/mirrorlists/7.0/factory-os.mirrorlist
```

```
enabled=1
gpgcheck=0
```

2. Install virt-manager from the Virtuoizzo repository:

```
# yum install virt-manager
```

3. Connect virt-manager to Virtuoizzo:

- a. Launch virt-manager.
- b. In the **File > Add connection** window, select **Virtuoizzo** in the hypervisor drop-down list.
- c. Select the preferred connection type (SSH by default) and specify connection options.
- d. Enter the password.

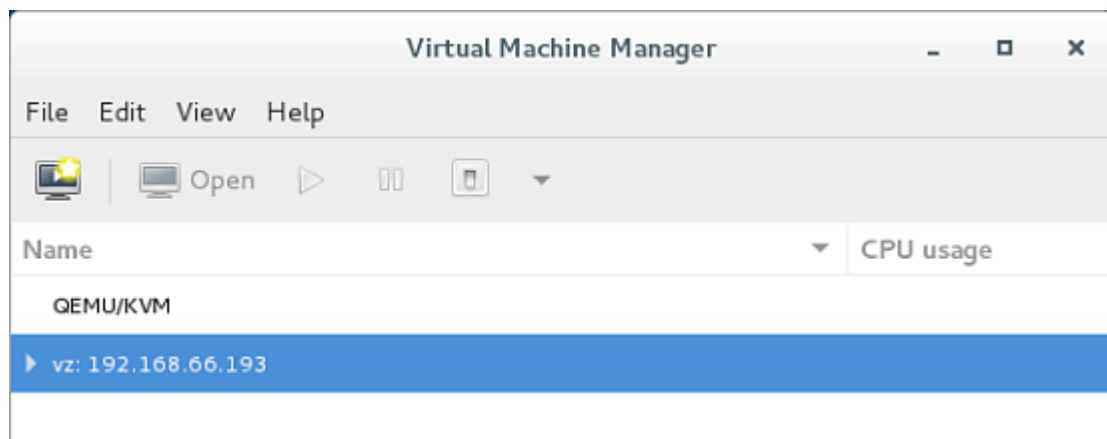
4. On the **Edit > Preferences > NewVM** tab, change **Storage format** from `System default (qcow2)` to `QCOW2`.

Note: Virt-manager can create its own default network with NAT while creating a container or VM. Virtuoizzo 7 does not support NAT by default. For instructions on how to enable NAT, visit https://openvz.org/Using_NAT_for_container_with_private_IPs. We recommend that you use a Virtuoizzo bridged or host-routed network.

4.9.2. Using virt-manager

To create a virtual machine with virt-manager, do the following:

1. Launch virt-manager.
2. Select the Virtuoizzo hypervisor.



3. Click the new VM icon.
4. Choose the options as shown on the screenshot and click **Forward**:

New VM

Create a new virtual machine
Step 1 of 5

Connection: vz: 192.168.66.193 ▼

Choose virtualization type

☒ Virtual machine
☐ Container

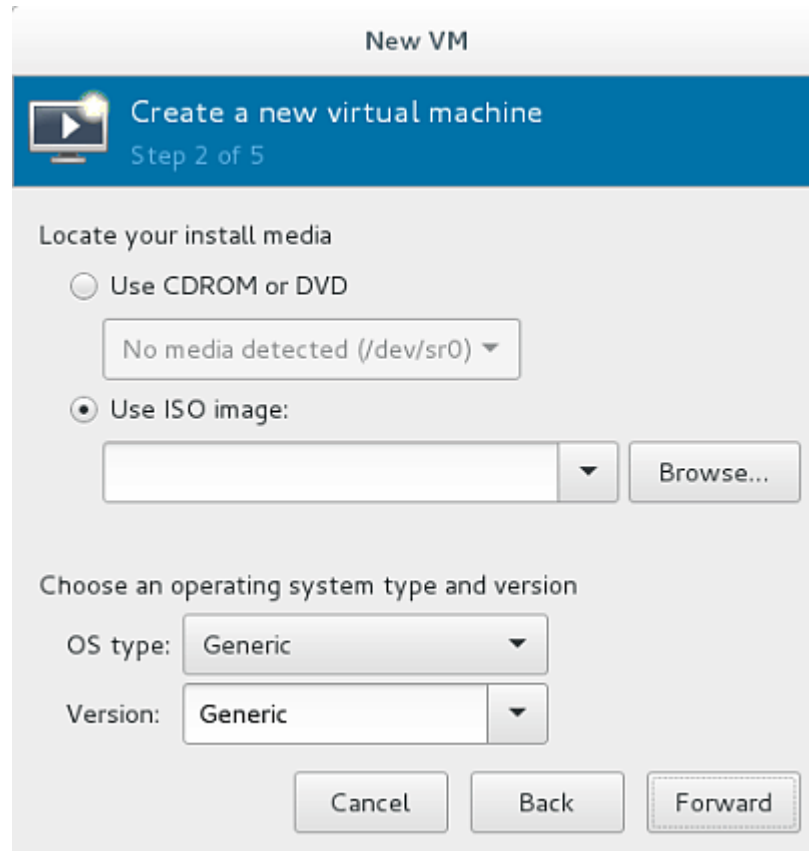
Choose how you would like to install the operating system

☒ Local install media (ISO image or CDROM)
☐ Network Install (HTTP, FTP, or NFS)
☐ Network Boot (PXE)
☐ Import existing disk image

► Architecture options

Cancel Back Forward

5. Click **Browse** next to the **Use ISO image** field.



New VM

Create a new virtual machine
Step 2 of 5

Locate your install media

☐ Use CDROM or DVD

No media detected (/dev/sr0) ▼

☒ Use ISO image:

▼ Browse...

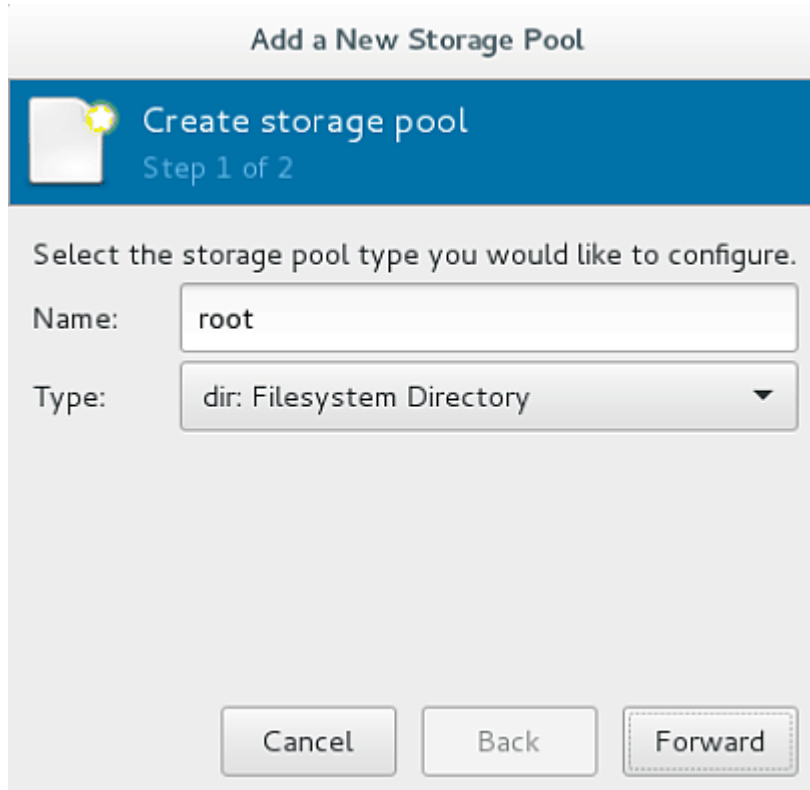
Choose an operating system type and version

OS type: Generic ▼

Version: Generic ▼

Cancel Back Forward

6. Click the plus button to add a new image repository.
7. Specify the repository name and type.



Add a New Storage Pool

Create storage pool
Step 1 of 2

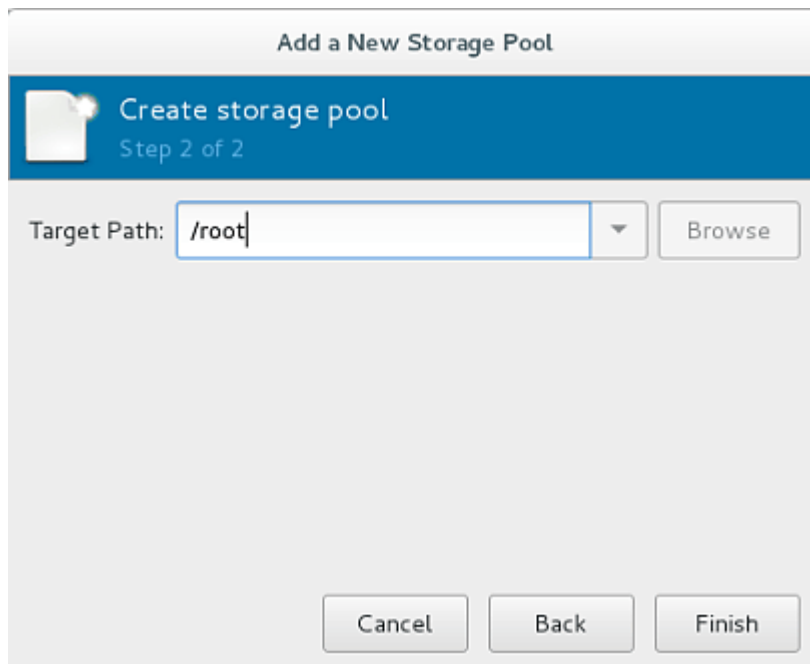
Select the storage pool type you would like to configure.

Name: root

Type: dir: Filesystem Directory

Cancel Back Forward

8. Enter the path to your image repository and click **Finish**.



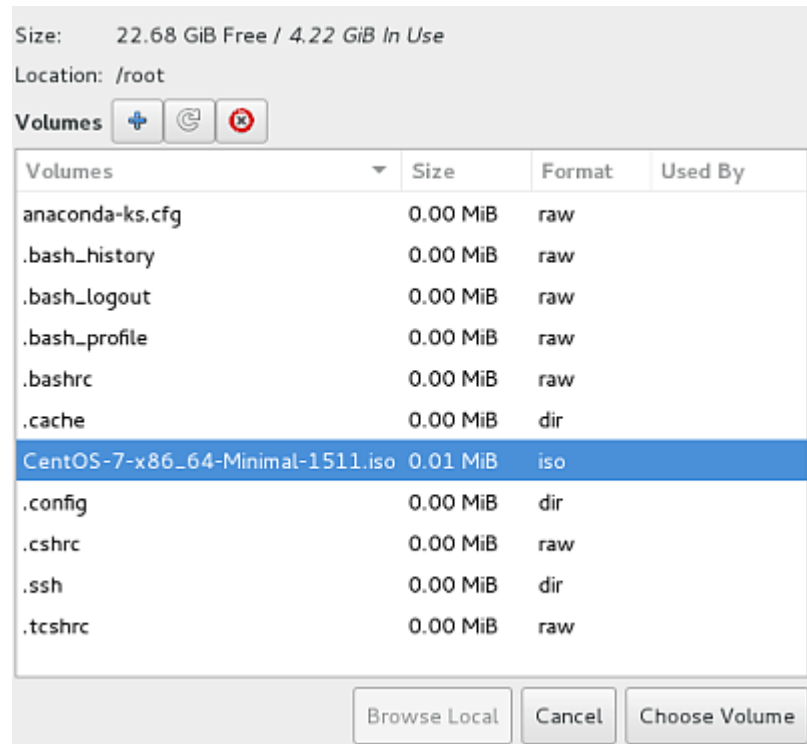
Add a New Storage Pool

Create storage pool
Step 2 of 2

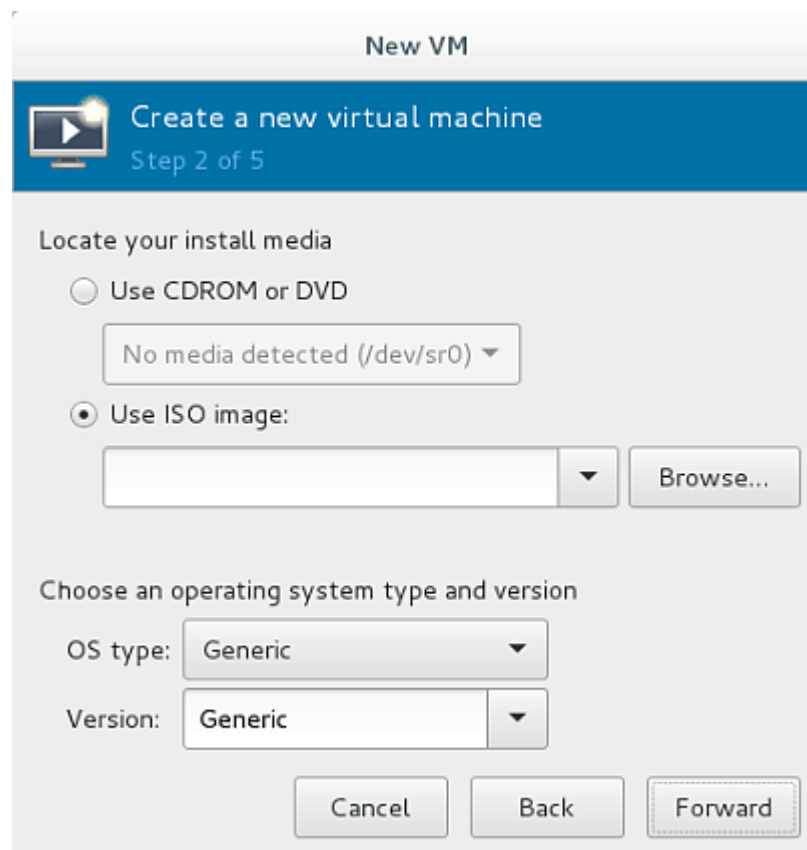
Target Path: /root Browse

Cancel Back Finish

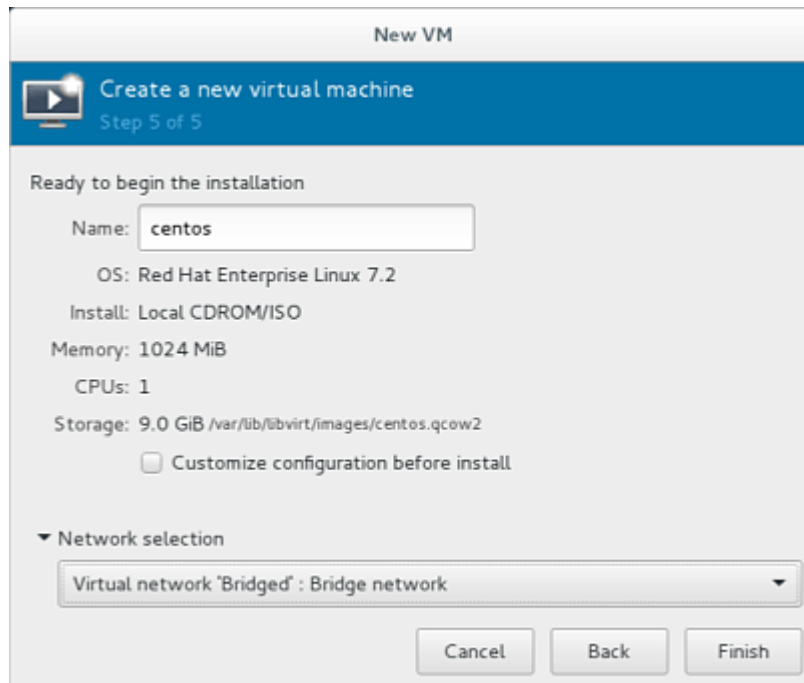
9. Choose your ISO image and click **Choose Volume**.



10. Choose your OS type and click **Forward**.



11. Choose VM options and click **Forward**.
12. Choose storage options and click **Forward**.
13. Enter the VM name and choose the network as shown below.



14. When the VM starts, connect to its console by clicking the **Open** button.

The virtual machine is ready. A Virtuozzo container can be created in a similar way.

4.10. Working with Microsoft Windows Virtual Machines

Virtuozzo 7 Beta supports several Windows Server guest operating systems listed earlier in this homework. Windows-based VMs do not differ from Linux-based ones, so you can also go through the homework with a virtual machine running Windows Server 2012 R2, for example. To do this, you will first need to:

1. Create a blank VM for a Windows distribution.
2. Install Windows Server 2012 R2 in the VM.
3. Install Virtuozzo guest tools in the VM.