Virtuozzo Hybrid Server 7
Installation on ASRock Rack

May 18, 2022
# Contents

1. **About This Guide** .................................................. 1

2. **Installing Virtuozzo Hybrid Server** .................................. 2
   2.1 Preparing Installation Media ........................................... 2
   2.2 Booting from USB ..................................................... 4
   2.3 Performing Installation ............................................. 5
       2.3.1 CLI Installation ................................................ 6
       2.3.2 GUI Installation .............................................. 14

3. **Enabling SR-IOV on ASRock Rack** .................................. 25
   3.1 About SR-IOV ...................................................... 25
   3.2 Enabling SR-IOV and IOMMU Support ........................... 27
   3.3 Configuring SR-IOV VFs Network Adapters ................... 28
   3.4 Assigning SR-IOV VF Network Adapter to Linux KVM VM ........ 30
CHAPTER 1

About This Guide

This guide explains how to install Virtuozzo Hybrid Server 7 on the ASRock Rack X570D4U-2L2T server powered by AMD Ryzen 9 5950X 16-Core @ 3.40GHz with CLI management or GUI (Virtuozzo Automator) support.

About the Hardware

The ASRock Rack X570D4U-2L2T features:

• CPU AMD Ryzen 9 5950X 16-Core @3.40GHz.

• Four 288-pin DDR4 DIMM Slots. Support for up to 128GB DDR4 ECC/UDIMM (conditionally supports ECC error reporting function).

• Four hot swap 3.5-inch HDD/SSD bays.

• Three fixed 2.5-inch HDD/SSD bays.

• Two M.2 Slots on the Motherboard, our test server came with 2 NVMe Drives on 1 x each slot.

• Dual Intel Gigabit Ethernet ports (Intel i210 controller).

• IPMI connectivity (ASpeed AST2500) BMC.

• Two 10 Gigabit Ethernet ports.
CHAPTER 2

Installing Virtuozzo Hybrid Server

This chapter explains how to install Virtuozzo Hybrid Server 7 on the AMD Ryzen 9 5950X 16-Core @ 3.40GHz.

Note: You can check the general Virtuozzo Hybrid Server 7 Installation Guide.

2.1 Preparing Installation Media

In this guide, we will perform a USB-based installation. The Fedora media writer software will be used to create the installation USB.

1. Download the Fedora media writer for your OS from here:
   - MacOS: https://getfedora.org/fmw/FedoraMediaWriter-osx-latest.dmg

2. Download the Virtuozzo Hybrid Server 7 ISO by following the steps in the linked Obtaining the Distribution Image guide. Also, in the received email, you will find a license key number to activate a Virtuozzo Hybrid Server.

3. Open the Fedora media writer and click the Custom Image option.
4. Select the VHS ISO and click the **Write to Disk** button.

5. You can close once the media is ready.
2.2 Booting from USB

Once the installation USB is ready, the next step will be starting the ASRock Rack X570D4U-2L2T server and going to the boot menu.

1. Boot the server.

2. Press F11 to access the boot menu.

3. Choose the installer USB to boot.
2.3 Performing Installation

Based on your preferences, you can install Virtuozzo Hybrid Server 7 with:

- **CLI support** - no GUI/Automator will be installed; VHS administration will be performed from the command line

- **GUI support** - Automator is installed; VHS administration will be done from the Control Plane (Web UI)
2.3.1 CLI Installation

1. Click the *Install Virtuozzo Hybrid Server 7 with CLI management* option.

2. Within the installation window, you need to configure the following points:
   - accept the EULA
   - select storage type
   - choose the installation destination
   - configure the network interface
3. Click **EULA** and read the agreement. Accept it and click **Done** to proceed.
4. Return to the main menu and go to Select Storage Type. Ensure that “Basic Storage” is selected and click the Done button.
5. Next, configure the network interface by clicking **Network & Host Name**. We use the *vhs.example.com* hostname as an example. Choose the required interface (*enp36s0f0* in our case). Navigate to **Configure > IPv4 Settings > Method > Manual**, click **Add** and enter the following data:

- **LAN IP address**
- **Netmask**
- **Gateway**
- **DNS servers**

Click **Save** and **Done** when ready.

6. **Installation Destination** is the last option you need to configure. We have two disks, select **System one** (to install the OS) and tick both for **Datastore**. At the bottom of the window, select the “*I will configure partitioning*” radio button and click **Done**.
7. By default, a small partition is created for the OS (64GB), and the remaining space is used as a **Datastore**. Click the **Done** button when ready.

**Note:** Everything on those drives will be erased.
8. Now, you can click the **Begin Installation** button.

9. While the installation is in progress, you can provide the license key, set the root password, and create a user.
10. In the **License Key** section, enter the license key you've got via email alongside the VHS image and click **Done**.

11. Go to the **Root Password** section and create a root account password. Click **Done** to proceed.
12. The **Create User** section helps create a new user. Click **Done** when ready.

13. Once the installation is done and all the configurations are provided, click **Finish configuration** and **Reboot** the server.
14. Verify installation by accessing the server over SSH via the LAN IP assigned during the installation.

```
jbustos@gibson ~ % ssh root@192.168.0.109
The authenticity of host '192.168.0.109 (192.168.0.109)' can't be established.
ED25519 key fingerprint is SHA256:o2zzZHAo8uJwb3NilOyBm4vX3mp+eb4ddzMEeTmg/XE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.109' (ED25519) to the list of known hosts.
root@192.168.0.109's password:
MOTD generated at: 03:09:06
Uptime: 1 min
OS: Virtuozzo release 7.5.4 (53)
IP: 192.168.0.109 10.37.130.2 fdb2:2c26:f4e4::1
Hostname: vhs.example.com
Kernel: 3.10.0-1160.53.1.v7.185.3 GNU/Linux
System Load: 0.11
/vz Usage: 0% of 1.6T
Swap Usage: 0%
RAM Free: 98% of 125.7GB
```

2.3.2 GUI Installation

1. Click the *Install Virtuozzo Hybrid Server 7 with GUI management* option.
2. Within the installation window, you need to configure the following points:

- accept the EULA
- disable Virtuozzo Storage (use local storage)
- configure the network interface
- define the IP of the Virtuozzo Automator (GUI)
- choose the installation destination
3. Click **EULA** and read the agreement. Accept it and click **Done** to proceed.
4. Return to the main menu and go to Virtuozzo Storage. Select the “No Virtuozzo Storage” option and click the **Done** button.

5. Next, configure the network interface by clicking Network & Host Name. We use the *vhs.example.com* hostname as an example. Choose the required interface (*enp36s0f0* in our case). Navigate to Configure > IPv4 Settings > Method > Manual, click **Add** and enter the following data:

- **IP address**
- **Netmask**
- **Gateway**
- **DNS servers**
Click **Save** and **Done** when ready.

6. In the **Virtuozo Automator** section, choose the “*Management Panel and Compute*” option, enter the IP and hostname for the Automator GUI and click the **Done** button.

   This action will deploy a Virtuozo system container with the Automator GUI. The network interface specified in the previous step will be used to create a Linux bridge so that Automator GUI IP will be available through the bridge on the same LAN.
7. **Installation Destination** is the last option you need to configure. We have two disks, select System one (to install the OS) and tick both for Datastore. At the bottom of the window, select the “I will configure partitioning” radio button and click Done.

8. By default, a small partition is created for the OS (64GB), and the remaining space is used as a Datastore. Click the Done button when ready.

**Note:** Everything on those drives will be erased.
9. Now, you can click the **Begin Installation** button.

10. While the installation is in progress, you can provide the license key, set the root password, and create a user.
11. In the **License Key** section, enter the license key you've got via email alongside the VHS image and click **Done**.

12. Go to the **Root Password** section and create a root account password. Click **Done** to proceed.
13. The **Create User** section helps create a new user. Click **Done** when ready.

14. Once the installation is done and all the configurations are provided, click **Finish configuration** and **Reboot** the server.
15. Once the system is booted, go to the IP specified during the Virtuozzo Automator configuration (https://ip-of-the-automator-gui). Use the root user with the password you've specified in the 12th step to log in.

Or you can verify the installation by accessing the server over SSH via the LAN IP assigned during the installation.
The authenticity of host '192.168.0.109 (192.168.0.109)' can't be established.
ED25519 key fingerprint is SHA256:o2zzZHAo8uJwb3NIOyBm4YX3mp+eb4ddzMEeTmgyXE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.109' (ED25519) to the list of known hosts.
root@192.168.0.109's password:
MOTD generated at: 03:09:06
Uptime: 1 min
OS: Virtuozzo release 7.5.4 (53)
IP: 192.168.0.109 10.37.130.2 fdb2:2c26:f4e4::1
Hostname: vhs.example.com
Kernel: 3.10.0-1160.53.1.vz7.185.3 GNU/Linux
System Load: 0.11
/vz Usage: 0% of 1.6T
Swap Usage: 0%
RAM Free: 98% of 125.7GB
This chapter explains how to enable and configure SR-IOV on Virtuozzo Hybrid Server 7 and assign the PCI SR-IOV devices to virtual machines.

3.1 About SR-IOV

**Single Root I/O Virtualization (SR-IOV)** is a hardware specification that allows a single PCI Express (PCIe) endpoint to function as numerous devices (e.g., a single Ethernet port will appear as multiple, separate, physical devices). **Physical Functions (PFs)** and **Virtual Functions (VFs)** are two PCIe functions that enable this functionality. **PCI passthrough** is another functionality provided by the IOMMU driver, which allows assigning PCIe devices directly to Virtual Hosts. It allows virtual machines to benefit from direct PCI device assignment while only requiring one slot on the host physical computer.

When SR-IOV is used in conjunction with the PCI passthrough functionality in its most basic form (**SR-IOV VF PCI**), we can inject an SR-IOV network VF into a Linux KVM VM.

**Physical Functions** are Full PCIe devices that include SR-IOV capabilities (PCIe Devices).

**Virtual Functions** are PCIe functions that only compute I/O. These are the resulting VFs from the PCIe PFs device (Virtual NICs).

The direct assignment (**PCI passthrough**) approach also bypasses the Linux virtual bridges that are usually in place for virtual machine networking.
Why use SR-IOV on VHS?

Virtual Functions offer **better performance** than paravirtual drivers or emulated devices. The performance is nearly **close to native**. An overlooked feature when using VFs is **data protection**; all the data is managed and controlled by the hardware, which improves data protection between virtual machines on the same host. These features allow the host to have an increased virtual machine density. It is important to mention that Virtual Machines with assigned PCI devices cannot be migrated to other hosts, which in our case works well as we will be using a dedicated host with local storage, and machines will always be running on this host.
3.2 Enabling SR-IOV and IOMMU Support

**SR-IOV VF PCI** relies heavily on hardware support. Here is a list of **hardware requirements for SR-IOV**:

- Your firmware (BIOS or UEFI) must support SR-IOV.
- Your PCIe devices (e.g., Ethernet Port) must support SR-IOV.
- Root Ports or the PCIe switch must support ARI (alternative routing ID interpretation).
- Device assignment requires IOMMU (I/O Memory Management Unit) support in the CPU and firmware.

The ASRock Rack X570D4U-2L2T server powered by AMD Ryzen 9 5950X 16-Core @ 3.40GHz was certified for compatibility with Virtuozzo Hybrid Server version 7.5.4 by the Virtuozzo Technology Alliances team. All validation tests available for the standalone server configuration were completed with a 100% pass rate. A wide range of installation scenarios and functional tests were covered, including SR-IOV and PCIe passthrough support.

1. Check if **IOMMU** is enabled on the BIOS or UEFI. Go to Advanced > AMD CBS > NBIO Common Options > IOMMU and enable the option, if needed.
2. Check that **SR-IOV** support is enabled on the BIOS or UEFI. Go to *Advanced > Chipset Configuration > SR-IOV Support* and enable the option, if needed.

### 3.3 Configuring SR-IOV VFs Network Adapters

There are a few configuration steps required to create SR-IOV VFs network adapters:

- Identify network cards on your system and verify SR-IOV support.
- Verify that IOMMU is supported and loaded by your kernel when booting.
- Define how many SR-IOV VFs network adapters are created.
- Enable persistent device creation after reboot.

1. Identify the available network cards.
2. Use the **Bus info** (without the `pci@` part) from the previous step output to check for SR-IOV support. For example, on the `enp360f0` device:

```
# lspci -vs 0000:24:00.0
```

```
[root@vhs ~]# lspci -vs 0000:24:00.0
24:00.0 Ethernet controller: Intel Corporation Ethernet Controller 10G X550T (rev 01)
   Subsystem: ASRock Incorporation Device 1563
   Flags: bus master, fast devsel, latency 0, IRQ 38
   Memory at fc800000 (64-bit, prefetchable) [size=4M]
   Memory at fcc04000 (64-bit, prefetchable) [size=16K]
   Expansion ROM at fb280000 [disabled] [size=512K]
   Capabilities: [40] Power Management version 3
   Capabilities: [50] MSI: Enable- Count=1/1 Maskable+ 64bit+
   Capabilities: [70] MSI-X: Enable+ Count=64 Masked-
   Capabilities: [a0] Express Endpoint, MSI 00
   Capabilities: [100] Advanced Error Reporting
   Capabilities: [140] Device Serial Number 00-00-00-ff-ff-00-00-00
   Capabilities: [150] Alternative Routing-ID Interpretation (ARI)
   **Capabilities: [160] Single Root I/O Virtualization (SR-IOV)**
   Capabilities: [1a0] Transaction Processing Hints
   Capabilities: [1b0] Access Control Services
   Capabilities: [1c0] Latency Tolerance Reporting
   Capabilities: [1d0] #19
   Kernel driver in use: ixgbe
   Kernel modules: ixgbe
```

3. Create **SR-IOV VFs network adapter**. You need to set the required adapters' number (one, in our example):

```
# echo 1 > /sys/class/net/enp36s0f0/device/sriov_numvfs
```

4. Verify that adapters were correctly created.

```
# lshw -c network -businfo
```
3.4 Assigning SR-IOV VF Network Adapter to Linux KVM VM

We have a virtual machine running CentOS 7. This virtual machine already has a primary network interface with IP address 192.168.0.223. Our goal is to assign a function network interface created previously to this running CentOS 7 virtual machine.

We have identified the new virtual function network adapter as *enp37s16* and can see that it appears as an available network interface on our hypervisor.

Now, let's see how to pass through this network virtual function to a virtual machine in a persistent way.

1. We will be using the *Bus info* in the next steps. However, you need to replace “:” and “@” with “_” when querying the device. For example, *pci@0000:25:10:0* » *pci_0000_25_10_0*.
2. Gather information about the **VF network adapter**. Use `virsh` and the `pci` address in the format from the previous step.

```
# virsh nodetdev_dumpxml pci_0000_25_10_0
```

3. Using the information highlighted in the image from the previous step, create an `xml` file to define the
network interface.

```
[root@vhs ~]# vim /tmp/vf.xml
```

```
<interface type='hostdev' managed='yes'>
  <mac address='52:54:00:6d:90:02'/>
  <driver name='vfio'/>
  <source>
    <address type='pci' domain='0' bus='37' slot='16' function='0'/>
  </source>
  <model type='virtio'/>
  <alias name='hostdev0'/>
</interface>
```

Attach the device using `virsh`. You can use the following flags:

- **–live** - attach the interface to the running virtual machine
- **–persistent** - use this network interface definition for future guest reboots (it doesn't persist on host reboots)
- **–config** - apply the changes after VM reboot

```
[root@vhs ~]# virsh attach-device MyVM vf.xml --persistent
Device attached successfully
```

Verify that the device has been attached to the virtual machine correctly.

```
[root@vhs ~]# virsh dumpxml MyVM | grep -w hostdev -A9
<interface type='hostdev' managed='yes'>
  <mac address='52:54:00:6d:90:02'/>
  <driver name='vfio'/>
  <source>
    <address type='pci' domain='0x0000' bus='0x25' slot='0x10' function='0x0'/>
  </source>
  <model type='virtio'/>
  <alias name='hostdev0'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
</interface>
```

4. We have attached a virtual function network card to the virtual machine as a secondary network NIC. Let's see the status of the network interfaces on the virtual machine **before** assigning the device.
5. After assigning the device, we can see the new virtual function network adapter inside the virtual machine at the guest layer.

That's all! Now, you can proceed with network interface configuration as in any regular virtual machine.