ORACLE

Oracle Cloud Infrastructure Bare Metal and KVM

Configure KVM and Oracle Linux 9 with Linux and Windows Guest VMs

August 14, 2025 | Version 1.1

PURPOSE STATEMENT

This document will explain how to install and use Kernel-based Virtual Machine (KVM) on Bare Metal on OCI. The host operating system for KVM will be Oracle Linux 9. Once KVM is setup, we will create guest VMs.
The main purpose of this document is to show how easy and quickly to provision a KVM Host.

DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.
This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.
Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

REVISION

VERSION	DATE	PURPOSE
1.0	11 August 2025	Initial document creation
1.1	15 August 2025	Review

TABLE OF CONTENTS

WHY USE BARE METAL AND KVM?

Why use a Bare Metal? Bare metal compute instances are provisioned on an entire physical compute node, giving you direct access to the node's CPUs, memory, disks, network interface cards (NICs), and accelerators without any virtualization layer. Importantly, you do not share any hardware resources with other customers and have full control over the system once it has been provisioned to your tenancy. With the OCI console you can quickly deploy a single operating system such as Linux or Windows server on the Bare Metal compute instance. For a list of Bare Metal servers on OCI, refer to https://docs.oracle.com/en-us/iaas/Content/Compute/References/computeshapes.htm#baremetalshapes

Kernal-based Virtual Machine (KVM) is a virtualization solution integrated into the Linux kernel that turns a Linux system into a hypervisor, enabling it to run multiple isolated virtual machines (virtual computers) on the same physical hardware. These virtual machines can run Windows or Linux operating system, allowing for isolation of VMs on the same Bare Metal system.

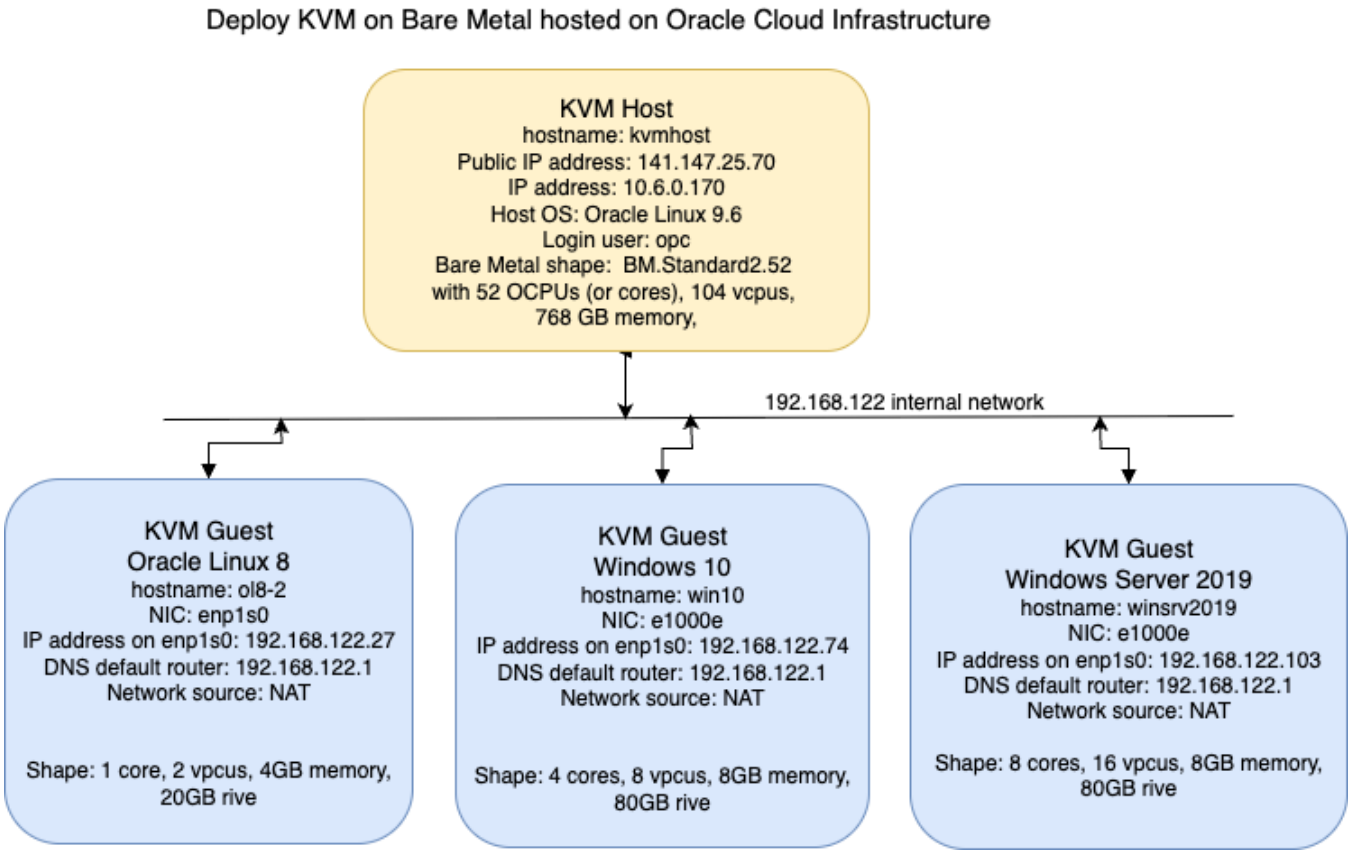
Deploying KVM on Bare Metal has been made easy on Oracle OCI. The document will explain detailed steps to deploy KVM.

The next section will explain the architecture used for this lab.

ARCHITECTURE OVERVIEW

The KVM Host, which is named *kvmhost* and located in OCI Frankfurt region. The KVM Host will host several virtual machines such as Oracle Linux 8, Windows 10 desktop, and Windows Server 2019.

The Bare Metal server was created on a public subnet. The picture below shows the high-level architecture: Bare Metal on OCI with 3 Guest VMs.



PROVISION BARE METAL IN ORACLE CLOUD INFRASTRUCTURE

Before installing and configuring the Bare metal (BM) compute instance in Oracle Cloud Infrastructure (OCI), check the prerequisites. Ensure there is enough storage space to store installation media (ISO images). We recommend 1TB for the boot volume and this will allow enough space to store ISO images.
This link shows KVM Host System requirements: https://docs.oracle.com/en/operating-systems/oracle-linux/9/kvm-user/kvm-SystemRequirementsandRecommendations.html#support
For this lab, we will deploy a bare metal server (BM.Standard2.52) on OCI in the Frankfurt region. This will be a standard installation of an OCI compute and Oracle Linux. The pictures below show the Bare Metal installation on BM.Standard2.52 using the full environment, all 52 cores. For more detail on how to deploy a compute instance on OCI, follow this link: https://docs.oracle.com/en/learn/oci-basics-tutorial/index.html#introduction
Here are the high level steps for KVM installation. Each step will be explained in more detail.

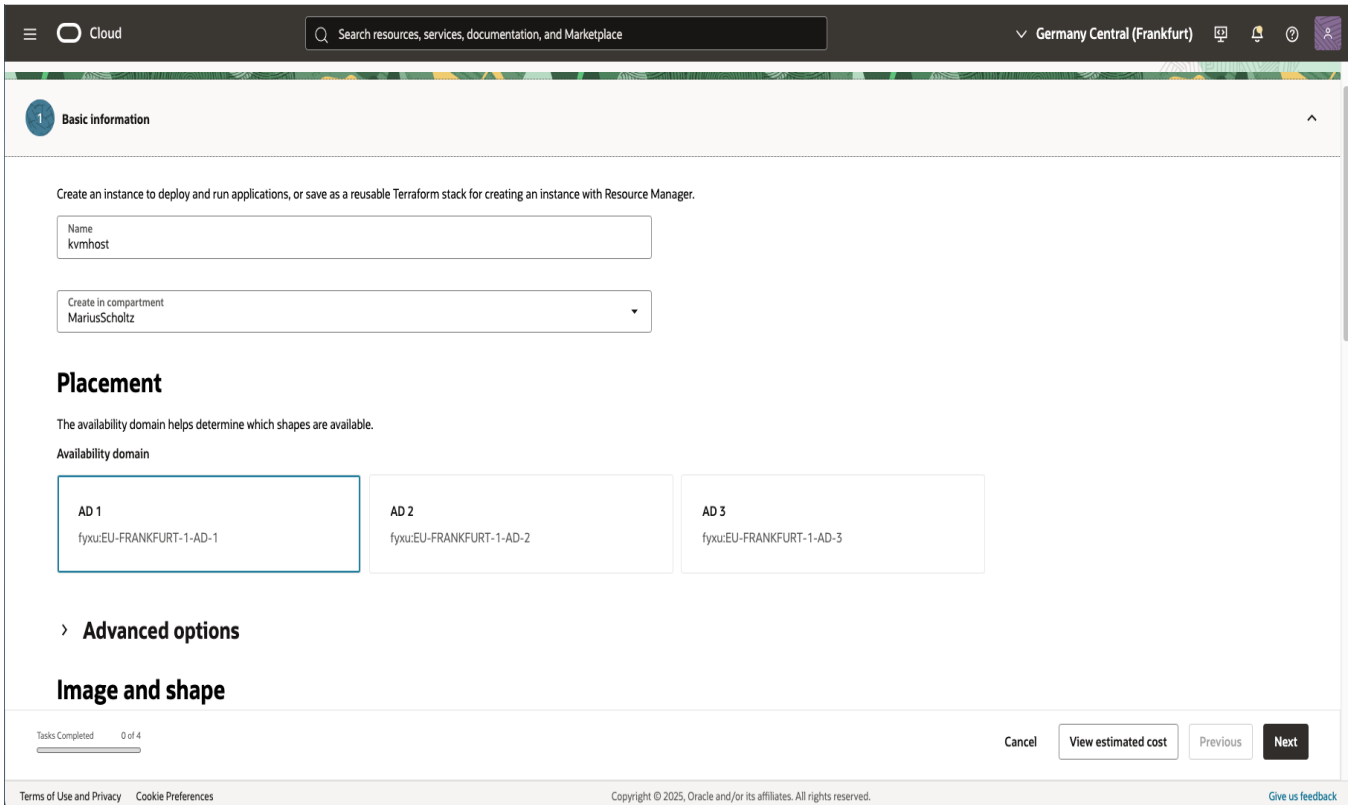
- Provision the Bare Metal BM.Standard2.52 with Oracle Linux 9 on OCI
- Connect to the KVM host using ssh
- Assign a password to the 'opc' user
- Access the OCI serial Console to monitor startup of the Bare Metal server
- Ensure Oracle Linux 9 is updated
- Install the KVM packages
- Validate the KVM installation
- Create Guest VMs
- Import KVM Guest VMs

Detailed steps to install and configure KVM are described I the next section.

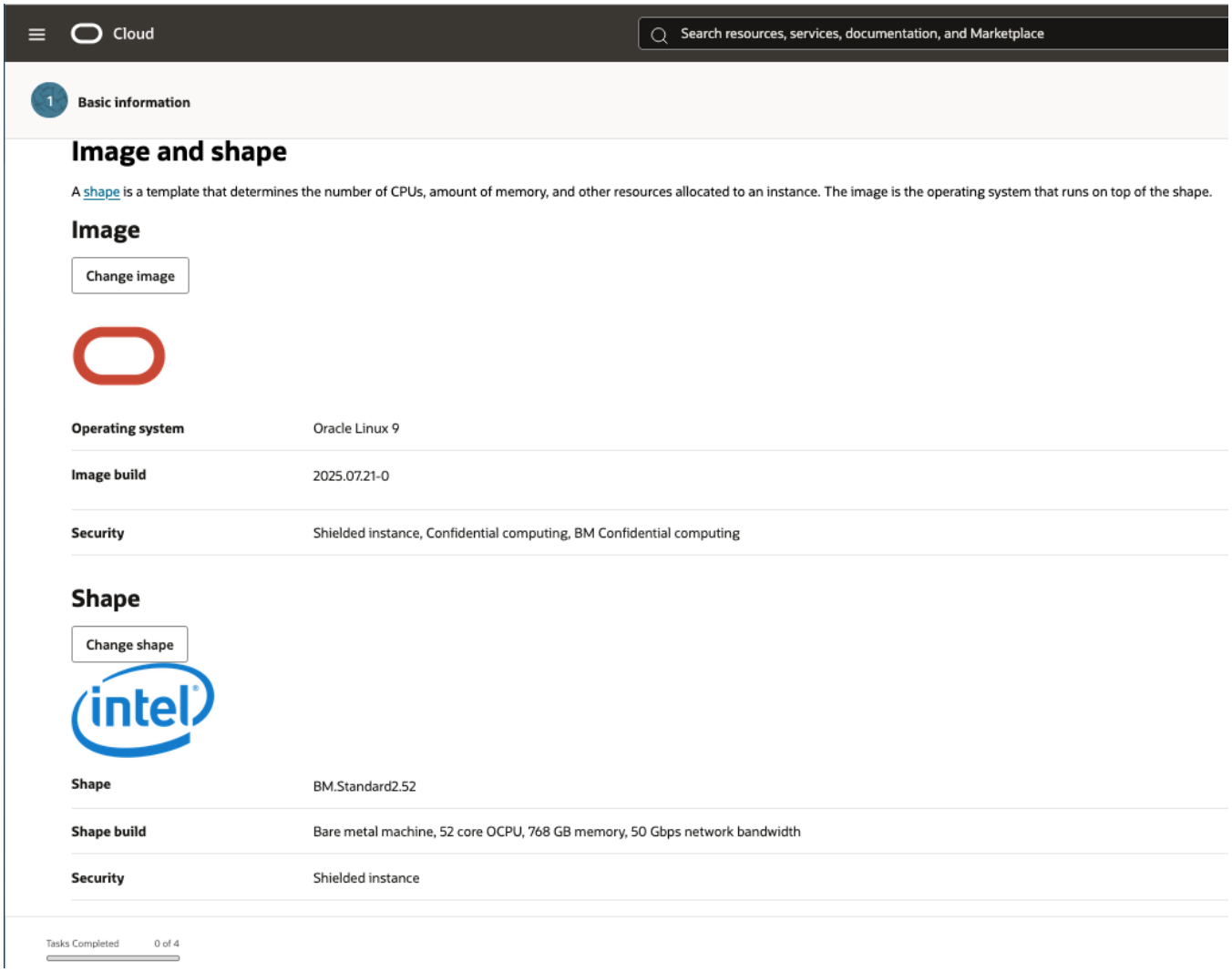
PROVISION THE BARE METAL BM.STANDARD2.52 WITH ORACLE LINUX 9 ON OCI

In this lab, the Bare Metal BM.Standard2.52 is deployed in Frankfurt.

The pictures below show Bare Metal compute instance creation on OCI Console. Use 1 TB boot volume to host the Oracle Linux 9 operating system and ISO images.



The picture below shows the shape and operating system selection.



CONNECT TO THE KVM HOST USING SSH

The default method for connecting to an Oracle Linux compute host on OCI is by using ssh. The IP Address of *kvmhost* is 141.147.25.70. To connect to this host, from your local system, use:

[myhost] $ ssh -i private-key opc@141.147.25.70

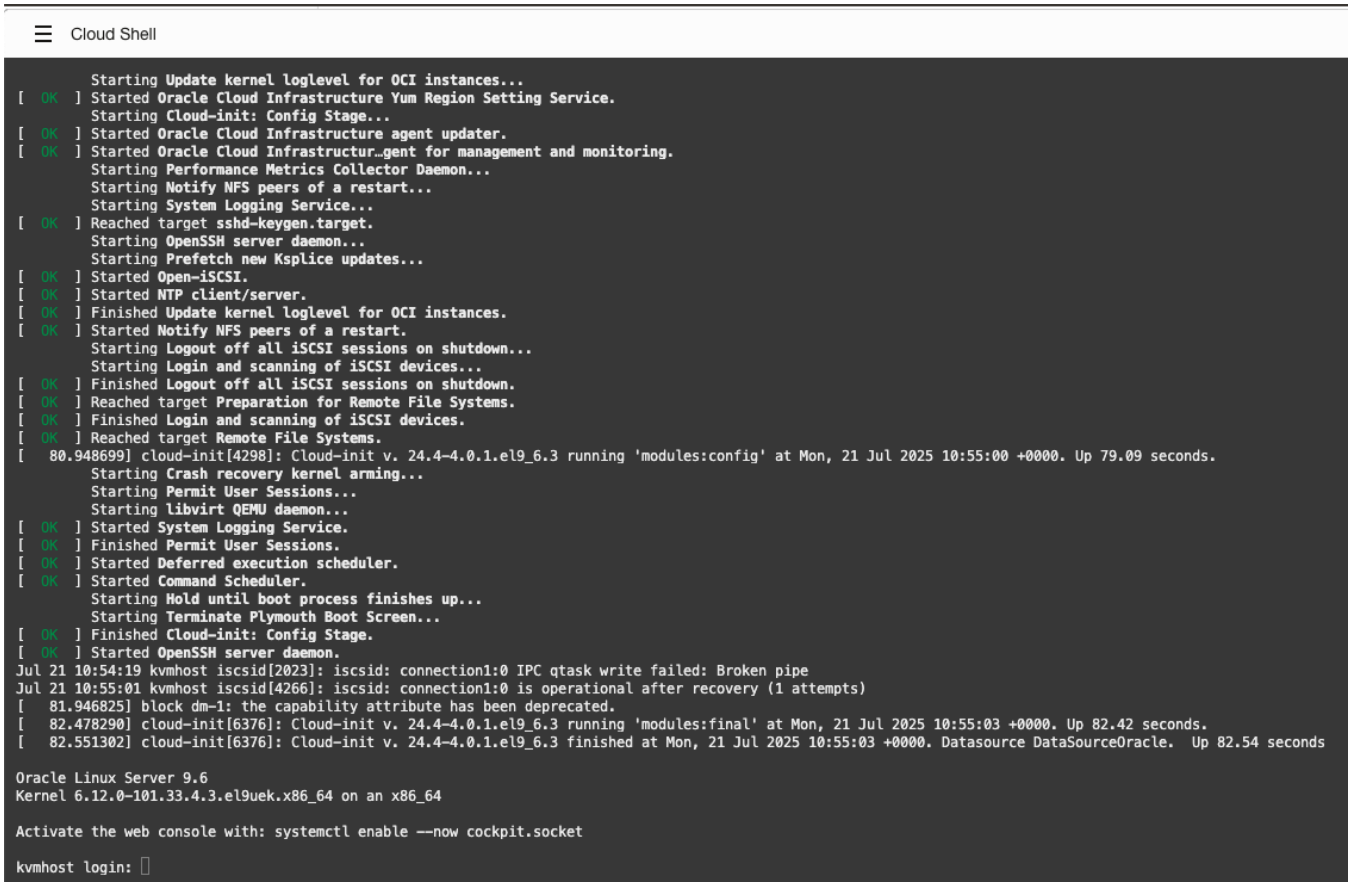
ASSIGN A PASSWORD TO THE 'OPC' USER

By default the opc password is not. Set the opc password to make it easy to log into the OCI serial console.

[opc@kvmhost ~]$ sudo passwd opc

ACCESS THE OCI SERIAL CONSOLE TO MONITOR STARTUP OF THE BARE METAL SERVER

From OCI console → click kvmhost → OS Management → Launch Cloud shell connection.
The picture below shows the Console output from *kvmhost* as it is booting up.



ENSURE ORACLE LINUX 9 IS UPDATED

Ensure the latest patches are applied to the KVM Host and then reboot the system.

```
[opc@kvmhost ~]$ sudo dnf update -y
[opc@kvmhost ~]$ sudo init 6 (or use sudo shutdown -r now)
```

INSTALL THE KVM PACKAGES

Provision Install and configure KVM on OCI Bare Metal

To install Kernel-based Virtual Machine (KVM) on Bare Metal on OCI involves installing packages from Oracle Linux 9 repository. Two distinct stacks exist to install KVM packages. The table below list the difference between the two stacks. In our example we will deploy the "Default KVM Stack".

Table 1: KVM Stack vs. Oracle KVM Stack.

CRITERIA	KVM STACK (DEFAULT)	ORACLE KVM STACK
Host running RHCK Red Hat Compatible Kernel	Yes	No, UEK kernel only
Host running with Unbreakable Enterprise Kernel	Yes	Yes
Provides newer functionality	Yes	Yes, even more than KVM Stack
Repository name: Different repositories are used	enabled by default: ol9_appstream sudo dnf reposlist	ol9_kvm_utils
Available from where?	Standard Oracle Linux package repositories or ULN channels. Disable the ol9_virt_utils yum repository or ol9_<arch>_virt_utils ULN channel.	Enable the ol9_kvm_utils yum repository or ol9_<arch>_kvm_utils ULN channel

You can easily change between the default KVM stack and Oracle Stack by using this guide:
https://docs.oracle.com/en/operating-systems/oracle-linux/9/kvm-user/kvm-OracleLinux9_SwitchKVMStacks.html#concept_gtw_lf5_lcc

Here are the different packages for each stack:
https://docs.oracle.com/en/operating-systems/oracle-linux/9/kvm-user/kvm-OracleLinux9_YumReposULNChannels.html

Install KVM packages on Bare Metal on OCI running Oracle Linux 9

In this lab we will deploy the default KVM Stack. Installation of KVM involved installing several rpm packaged from the *'yum repository'*.

```
[opc@kvmhost ~]$ sudo group install "Virtualization Host" "Virtualization Client"
[opc@kvmhost ~]$ sudo init 6 (or use sudo shutdown -r now)
```

This link provides more detail on KVM package installation:
https://docs.oracle.com/en/operating-systems/oracle-linux/9/kvm-user/kvm-InstallingKVMUserSpacePackages.html#kvm-linux-guest

VALIDATE THE KVM INSTALLATION

After the packages have been installed, it is time to check if for a successful installation. If all checks return a PASS value, the system can host guest VMs. If any of the tests fail, a reason is provided and information is displayed on how to resolve the issue, if such an option is available.

This link provides more detail on KVM package validation:
https://docs.oracle.com/en/operating-systems/oracle-linux/9/kvm-user/kvm-ValidatingtheHostSystem.html

```
[opc@kvmhost ~]$ sudo virt-host-validate qemu

QEMU: Checking for hardware virtualization :                 PASS
QEMU: Checking if device '/dev/kvm' exists :                 PASS
QEMU: Checking if device '/dev/kvm' is accessible :          PASS
QEMU: Checking if device '/dev/vhost-net' exists :           PASS
QEMU: Checking if device '/dev/net/tun' exists :             PASS
QEMU: Checking for cgroup 'cpu' controller support :         PASS
QEMU: Checking for cgroup 'cpuacct' controller support :     PASS
QEMU: Checking for cgroup 'cpuset' controller support :      PASS
QEMU: Checking for cgroup 'memory' controller support :      PASS
QEMU: Checking for cgroup 'devices' controller support :     PASS
QEMU: Checking for cgroup 'blkio' controller support :       PASS
QEMU: Checking for device assignment IOMMU support :         PASS
QEMU: Checking if IOMMU is enabled by kernel :               PASS
QEMU: Checking for secure guest support : WARN (Unknown if this platform has Secure Guest support)
[opc@kvmhost ~]$
```

WARN. This warning may be ignored. The secure guest support check is for AMD CPU, so when running on Intel it reports a warning. In this lab we are using Intel CPUs. For more detail, refer to
https://access.redhat.com/solutions/7060371

This command shows the libvirt version:

```
[opc@kvmhost ~]$ rpm -q libvirt
libvirt-10.10.0-7.3.0.1.el9_6.x86_64
```

Manage and enable libvirt daemons:
https://docs.oracle.com/en/operating-systems/oracle-linux/9/kvm-user/kvm-OracleLinux9_ManageLibvritdService.html

```
[opc@kvmhost ~]$ for drv in qemu network nodedev nwfilter secret storage interface;
    do
    sudo systemctl enable virt${drv}d.service;
    sudo systemctl enable virt${drv}d{,-ro,-admin}.socket;
    sudo systemctl start virt${drv}d{,-ro,-admin}.socket;
    done
    sudo systemctl enable virtproxyd.service
    sudo systemctl enable virtproxyd-tls.socket
    sudo systemctl start virtproxyd-tls.socket
```

CREATE KVM GUEST VMS USING COCKPIT

Create virtual machines using the Cockpit web console. The Cockpit web console is an Oracle Linux server administration tool designed for managing and monitoring Linux systems both locally and remotely. The graphical web console for management and administration is user-friendly and accessible to administrators at all levels of experience with Linux. The Cockpit installation package arrives out of the box and ready for use in most Oracle Linux distributions.
https://docs.oracle.com/en/operating-systems/oracle-linux/cockpit/cockpit-install.html
Cockpit videos and how to install:

https://github.com/mariusscholtz/Oracle-Cloud-Infrastructure-resources/tree/main/cockpit
https://www.youtube.com/watch?v=L9fMWCRcqIE
https://cockpit-project.org/running.html

To install Cockpit on the KVM Host, do the following:

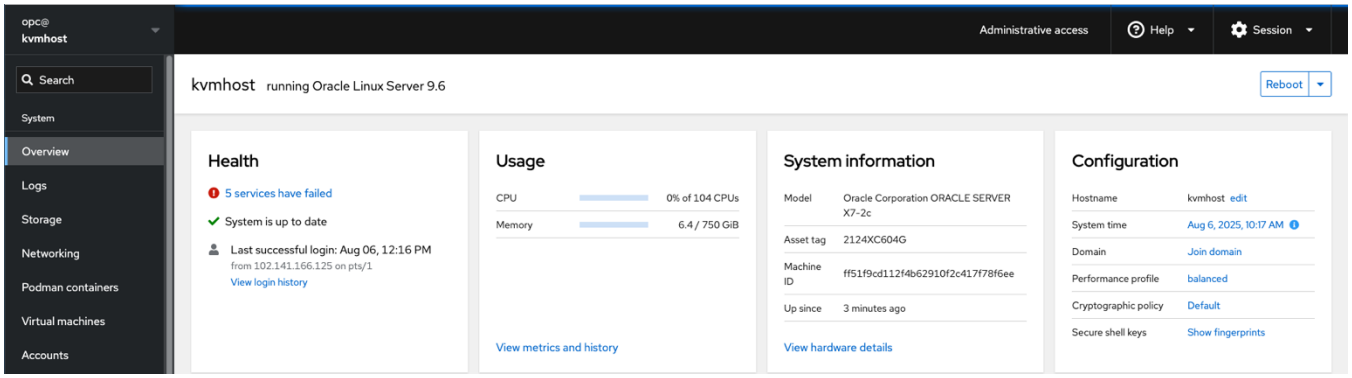
```
[opc@kvmhost ~]$ sudo dnf install -y cockpit cockpit-machines
[opc@kvmhost ~]$ sudo systemctl enable --now cockpit.socket
[opc@kvmhost ~]$ sudo systemctl status cockpit.socket
```

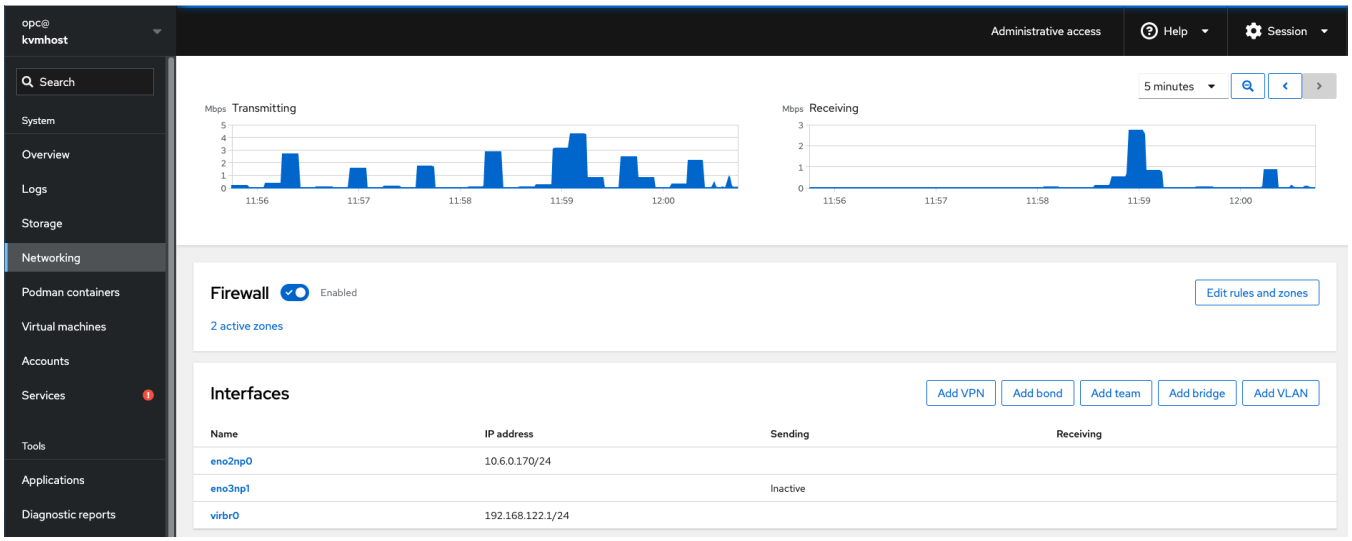
OCI instances are configured by default to only allow inbound ssh traffic. To access the cockpit web console, which is served on port 9090, you can create an ssh local port-forwarding tunnel as follows. On your local machine, create an SSH tunnel that allows you to connect to the cockpit port on the KVM Host:

```
[myhost] $ ssh -i ~/keys/private-key -L 9090:localhost:9090 opc@141.147.25.70
```

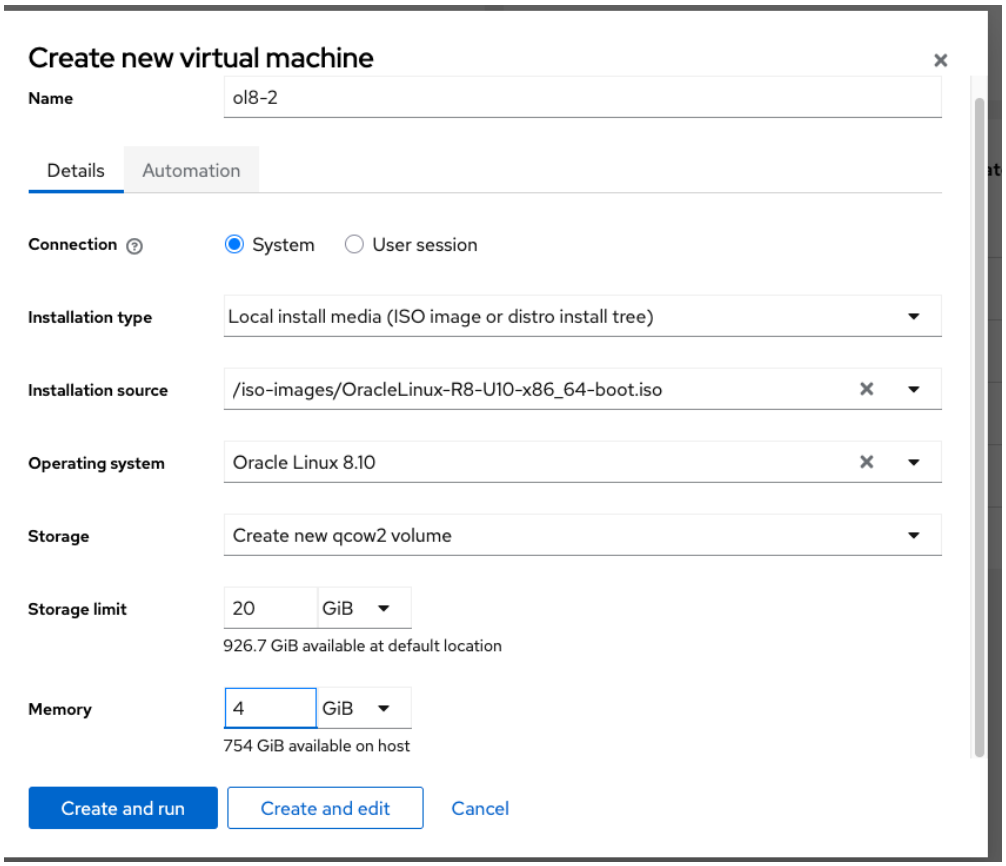
After creating the tunnel on your local system, open a browser to http://localhost:9090
Login using opc credentials and tick “Reuse my password for privileged tasks”
This document provides more detail about Cockpit:
https://github.com/mariusscholtz/Oracle-Cloud-Infrastructure-resources/tree/main/cockpit

The pictures below show Cockpit web console. The KVM server is called *kvmhost*.

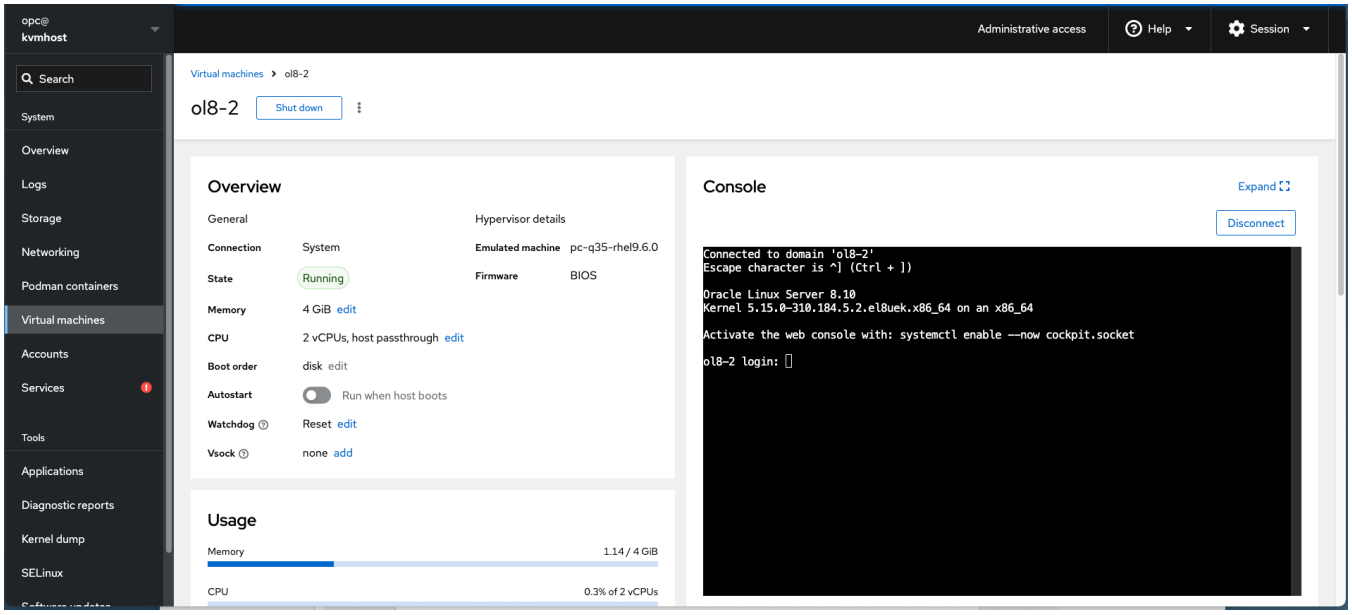


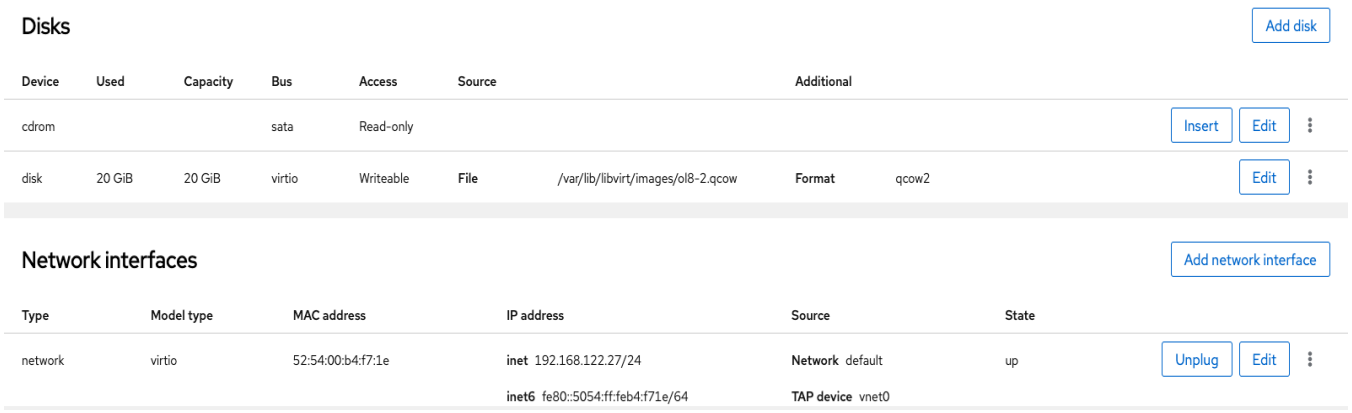


Creating a VM using Cockpit is easy. Clock on Virtual machines → Create VM.
The picture below shows an example on creating an Oracle Linux 8 VM.

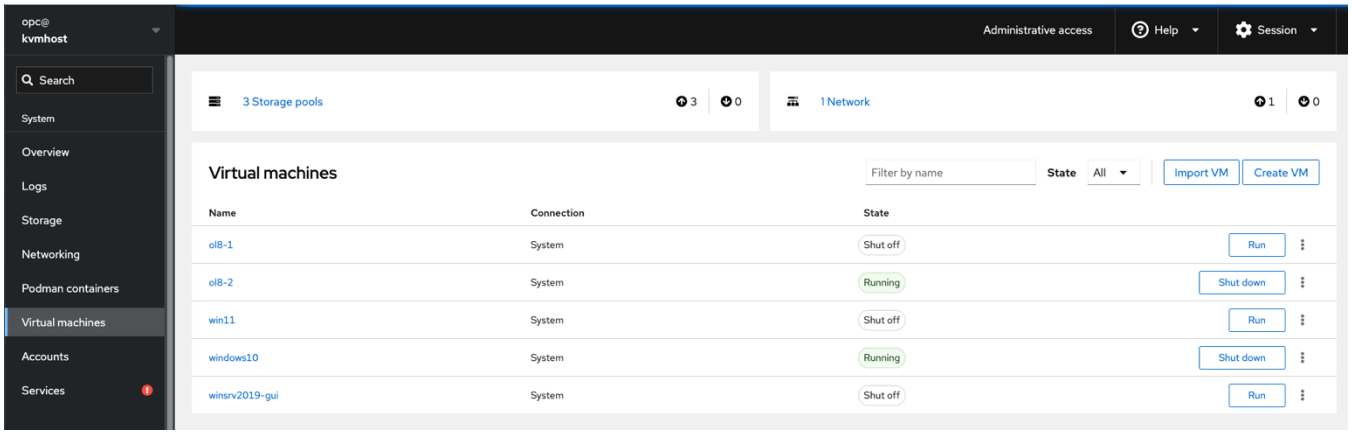


This picture shows detail of the virtual machine named *ol8-2*.

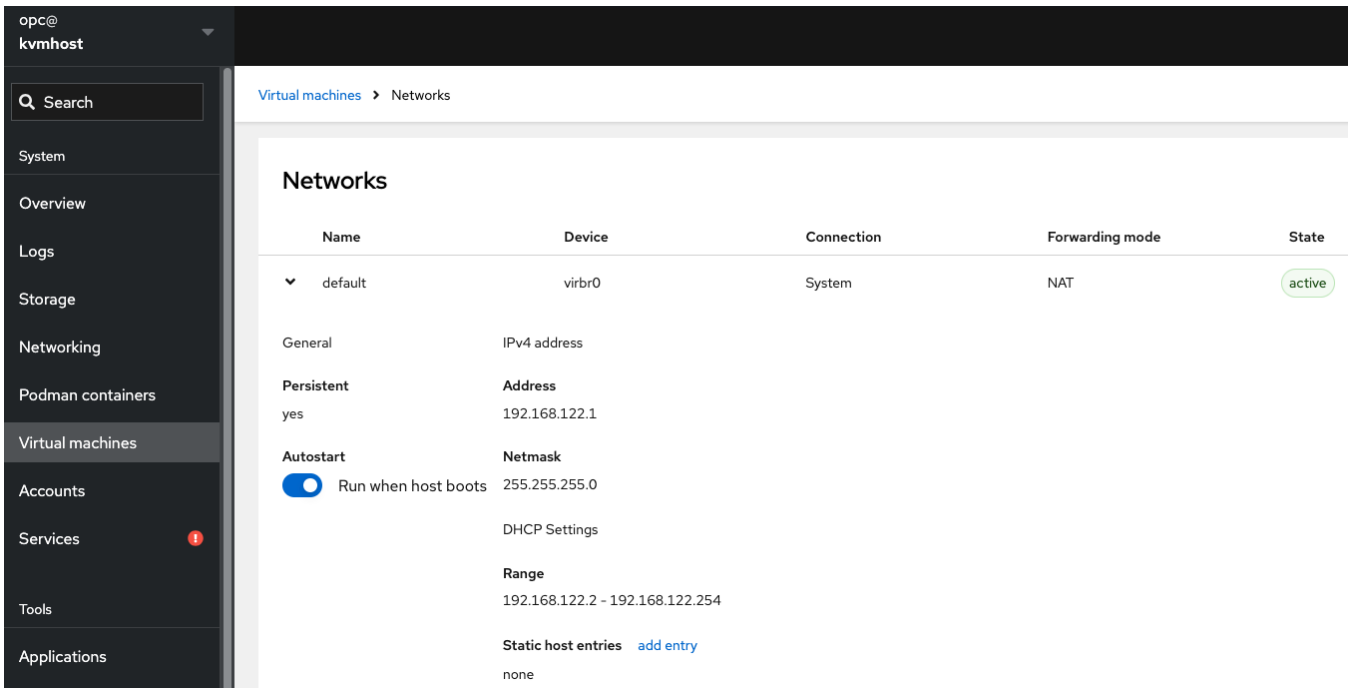




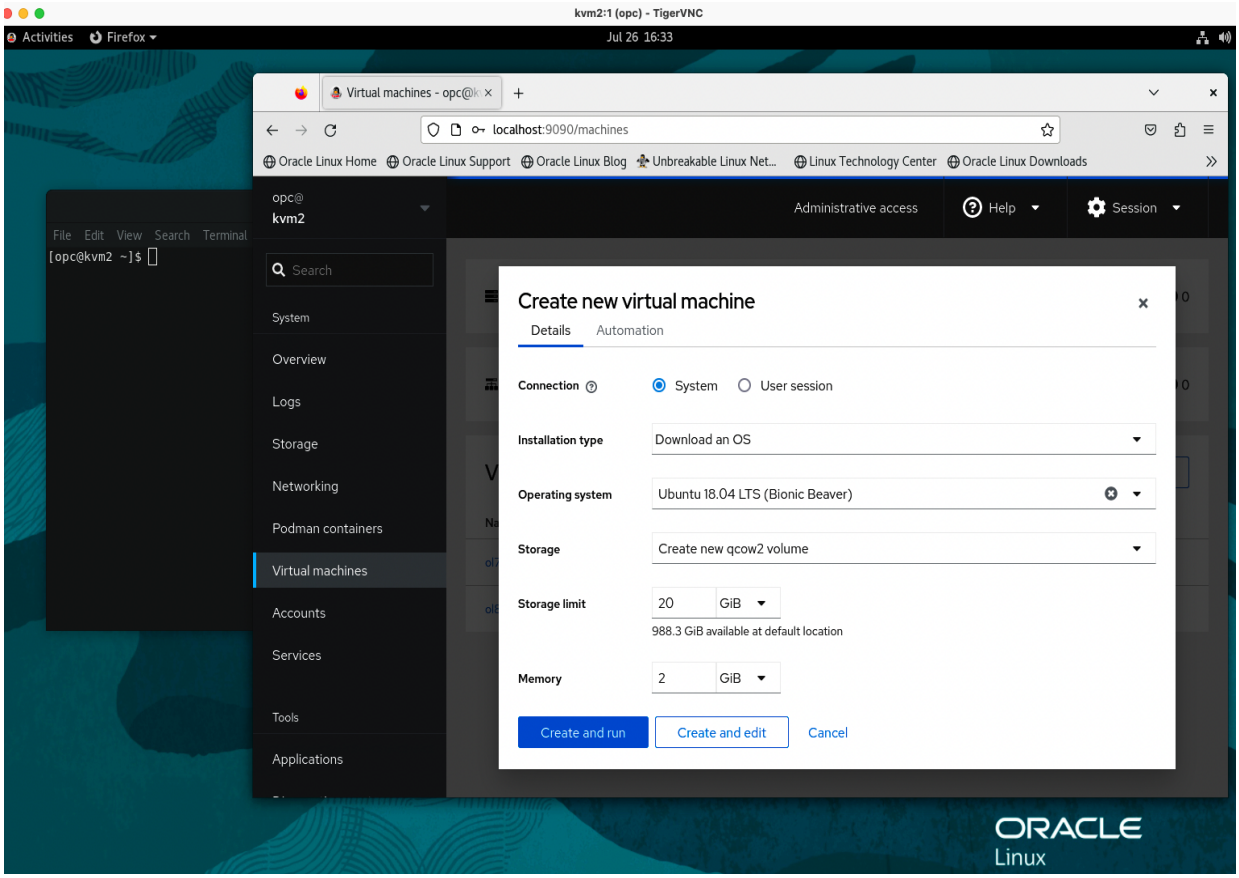
The picture below shows the Cockpit functionality to list all virtual machines (VMs). The vm ol8-2 and windows 10 are running.



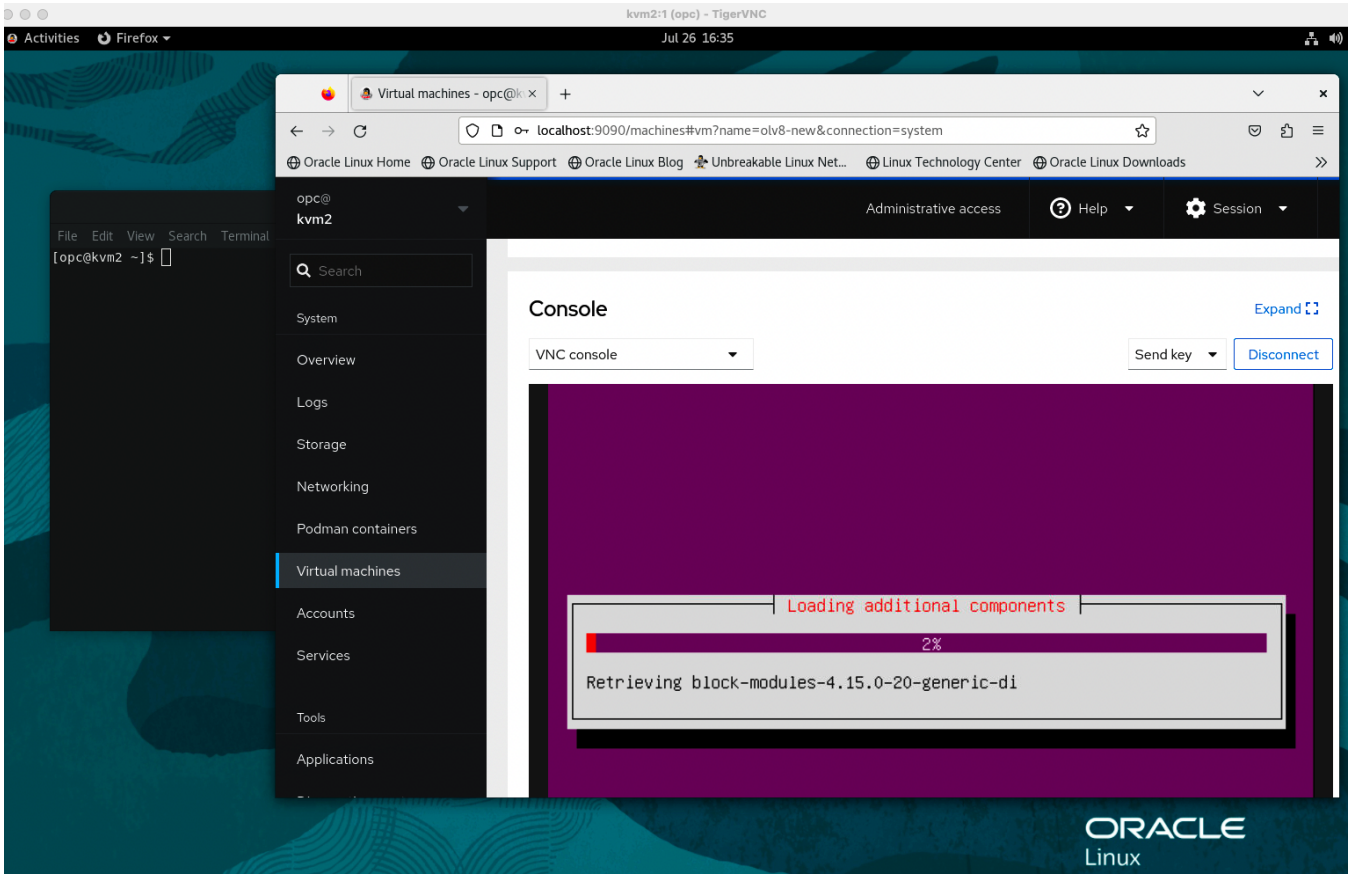
The picture below shows the Cockpit functionality to list the network, Click Virtual Machines → Click Network. NAT allows guest OS to connect to KVM host, and other guests on the network. Allow all guest VMs to connect to the Internet using the built-in virbr0 bridge.



Another example of how to Create a new VM using Cockpit:



Create a new VM ubuntu using cockpit:
In this example the boot file is located on /vm/ubuntu.qcow2.



CREATE KVM GUEST VMS USING VIRT-INSTALL

The command line tool virt-install is part of the KVM stack and allows for command line creation of virtual machines. For further reading, the virt-install documentation provides detail on KVM Guest creation: https://docs.oracle.com/en/operating-systems/oracle-linux/9/kvm-user/kvm-CreatingaNewVirtualMachine.html

The example below shows the command virt-install for creating a KVM guest (VM) running Oracle Linux 8: The hostname is vmol3. Oracle Linux 8 requires at least 14 GB of disk space, 40 GB recommended. This will allow you to access the console and perform a text install of the OS. eno3 is an interface on the KVM server. Installation of VM takes about 15 min.

```
[opc@kvmhost ~]$ sudo  virt-install --name=vmol3 --memory=2048 --vcpus=1 --location=/iso-images/OracleLinux-R8-U10-x86_64-dvd.iso --disk /vm/vmol2.qcow,size=14 --network type=direct,source=eno3 --os-variant=ol8.0 --nographics --console pty,target_type=serial --extra-args='console=tty0 console=ttyS0,115200n8 serial'
```

The picture below shows the console of the VM guest being installed.

```
================================================================================
Installation

1) [x] Language settings                 2) [x] Time settings
       (English (United States))                (Africa/Johannesburg timezone)
3) [x] Installation source               4) [x] Software selection
       (Local media)                            (Server with GUI)
5) [x] Installation Destination          6) [x] Kdump
       (Automatic partitioning                  (Kdump is enabled)
       selected)
7) [ ] Network configuration             8) [x] Root password
       (Not connected)                          (Password is set.)
9) [x] User creation
       (Administrator ms76152 will be
       created)

Please make a selection from the above ['b' to begin installation, 'q' to quit,
'r' to refresh]: b
================================================================================
================================================================================
Progress

.
Setting up the installation environment
Setting up org_fedora_oscap addon
Setting up com_redhat_kdump addon
..
Configuring storage
Creating disklabel on /dev/vda
Creating swap on /dev/vda2
Creating xfs on /dev/vda3
...
Running pre-installation scripts
.
Running pre-installation tasks
...
Installing.
Starting package installation process
Downloading packages
Preparing transaction from installation source
```

Access the vm console from the kvm server:

[opc@kvmhost ~]$ virsh console vmol3
Connected to domain vmol3
Escape character is ^] → This is Control and]
[ms76152@vmol3 ~]$ exit
logout
Oracle Linux Server 8.10
Kernel 5.15.0-206.153.7.1.el8uek.x86_64 on an x86_64
Activate the web console with: systemctl enable --now cockpit.socket
vmol3 login:

The picture below shows the console and startup of the guest VM vmol3:

```
[  OK  ] Reached target Network is Online.
         Starting Open-iSCSI...
         Starting Notify NFS peers of a restart...
         Starting System Logging Service...
[  OK  ] Started Notify NFS peers of a restart.
[   69.855402] Loading iSCSI transport class v2.0-870.
[  OK  ] Started System Logging Service.
[  OK  ] Started Open-iSCSI.
         Starting Logout off all iSCSI sessions on shutdown...
[  OK  ] Started Logout off all iSCSI sessions on shutdown.
[  OK  ] Reached target Remote File Systems (Pre).
[  OK  ] Reached target Remote File Systems.
         Starting Permit User Sessions...
         Starting Virtualization daemon...
         Starting Crash recovery kernel arming...
[  OK  ] Reached target Remote Encrypted Volumes.
[  OK  ] Started Permit User Sessions.
[  OK  ] Started Command Scheduler.
[  OK  ] Started Serial Getty on ttyS0.
         Starting Terminate Plymouth Boot Screen...
[  OK  ] Started Getty on tty1.
[  OK  ] Reached target Login Prompts.
[  OK  ] Started Job spooling tools.
[  OK  ] Started Terminate Plymouth Boot Screen.
[  OK  ] Started Virtualization daemon.
[  OK  ] Reached target Multi-User System.
         Starting Update UTMP about System Runlevel Changes...
[  OK  ] Started Update UTMP about System Runlevel Changes.
         Starting Network Manager Script Dispatcher Service...
[  OK  ] Started Network Manager Script Dispatcher Service.

Oracle Linux Server 8.10
Kernel 5.15.0-206.153.7.1.el8uek.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

vmol3 login:
```

The picture below shows VM detail displayed on the console.

```
vmol3 login:
Oracle Linux Server 8.10
Kernel 5.15.0-206.153.7.1.el8uek.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

vmol3 login: ms76152
Password:
[ms76152@vmol3 ~]$ lsmem
RANGE                                  SIZE  STATE REMOVABLE BLOCK
0x0000000000000000-0x000000007fffffff   2G online       yes  0-15

Memory block size:       128M
Total online memory:       2G
Total offline memory:      0B
[ms76152@vmol3 ~]$ lscpu
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              1
On-line CPU(s) list: 0
Thread(s) per core:  1
Core(s) per socket:  1
Socket(s):           1
NUMA node(s):        1
Vendor ID:           GenuineIntel
CPU family:          6
Model:               85
Model name:          Intel Xeon Processor (Skylake, IBRS)
Stepping:            4
CPU MHz:             1995.340
BogoMIPS:            3990.68
Hypervisor vendor:   KVM
Virtualization type: full
L1d cache:           32K
L1i cache:           32K
L2 cache:            4096K
L3 cache:            16384K
NUMA node0 CPU(s):   0
Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat
```

Here is another example of creating Oracle Linux 8 guest VM using virt-install:

Create ol8-2 using command line
20 GB disk space, 4GB mem, 2 vcpu, networked

[opc@kvmhost ~]$ sudo virt-install --name=ol8-2 --memory=4096 --vcpus=2 --location=/iso-images/OracleLinux-R8-U10-x86_64-boot.iso --disk /var/lib/libvirt/images/ol8-2.qcow,size=20 --network default --os-variant=ol8.0 --nographics --console pty,target_type=serial --extra-args='console=tty0 console=ttyS0,115200n8 serial'

From kvmhost you can now ssh to the new ol8-2 host with IP address 192.168.122.27

More examples of virt-install:
https://docs.oracle.com/en/operating-systems/oracle-linux/9/kvm-user/kvm-CreatingaNewVirtualMachine.html#reference_gmx_mbt_ncc

CREATE KVM GUEST VMS USING VIRT-MANAGER AND TIGERVNC DISPLAY

The virt-manager application is a desktop user interface for managing virtual machines through libvirt. It primarily targets KVM VMs, but also manages Xen and LXC Linux containers. https://virt-manager.org

virt-manager is installed automatically as part of the KVM software suite:

[opc@kvmhost ~]$ sudo dnf group install "Virtualization Host" "Virtualization Client"

If virt-manager is not installed, you can manually install virt-manager using:

[opc@kvmhost ~]$ sudo dnf install -y virt-manager

Refer to https://docs.oracle.com/en/virtualization/oracle-linux-virtualization-manager/

To access virt-manager, install the desktop on the KVM host and access it using VNC Server:
Install virt-manager - KVM GUI. Create VNC session to kvmhost
The kvm host is running Oracle Linux 9.6

[opc@kvmhost ~]$ sudo dnf groupinstall "Server with GUI" - See Mos Doc 2153562.1
Start the local console and reboot the KVM host: sudo init 6: OCI Console → OS Management → Console Connection → Launch Cloud Shell Connection

Next create a console connection on the OCI compute instance or configure tigervnc to the VM:

[opc@kvmhost ~]$ sudo dnf install tigervnc-server -y
As opc user, configure vncpassword: $ vncpasswd
Copy tigervnc libraries:
[opc@kvmhost ~]$ sudo cp /lib/systemd/system/vncserver@.service /etc/systemd/system/vncserver@\:1.service
[opc@kvmhost ~]$ sudo vi /etc/systemd/system/vncserver@\:1.service and replace USER with opc
Add a user mapping for opc user:
[opc@kvmhost ~]$ sudo vi /etc/tigervnc/vncserver.users
:1=opc

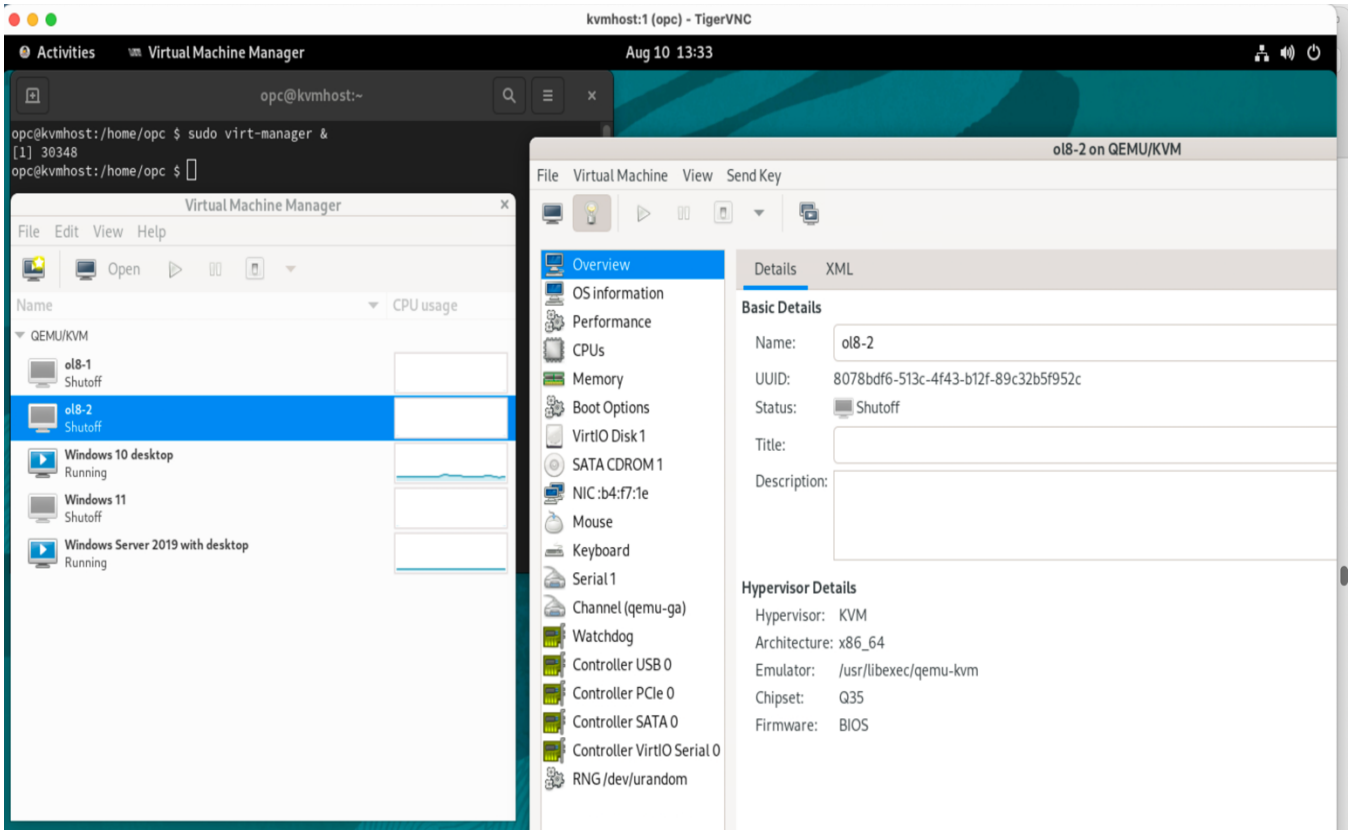
[opc@kvmhost ~]$ sudo systemctl daemon-reload
[opc@kvmhost ~]$ sudo systemctl enable vncserver@\:1.service
[opc@kvmhost ~]$ sudo systemctl start vncserver@\:1.service
[opc@kvmhost ~]$ sudo systemctl status vncserver@\:1.service
[opc@kvmhost ~]$ sudo firewall-cmd --list-all
[opc@kvmhost ~]$ sudo firewall-cmd --add-service=vnc-server --permanent
[opc@kvmhost ~]$ sudo firewall-cmd --reload

Open Tigervnc on your laptop to ip-address-of-kvmhost:5901, example:
141.147.25.70:5901

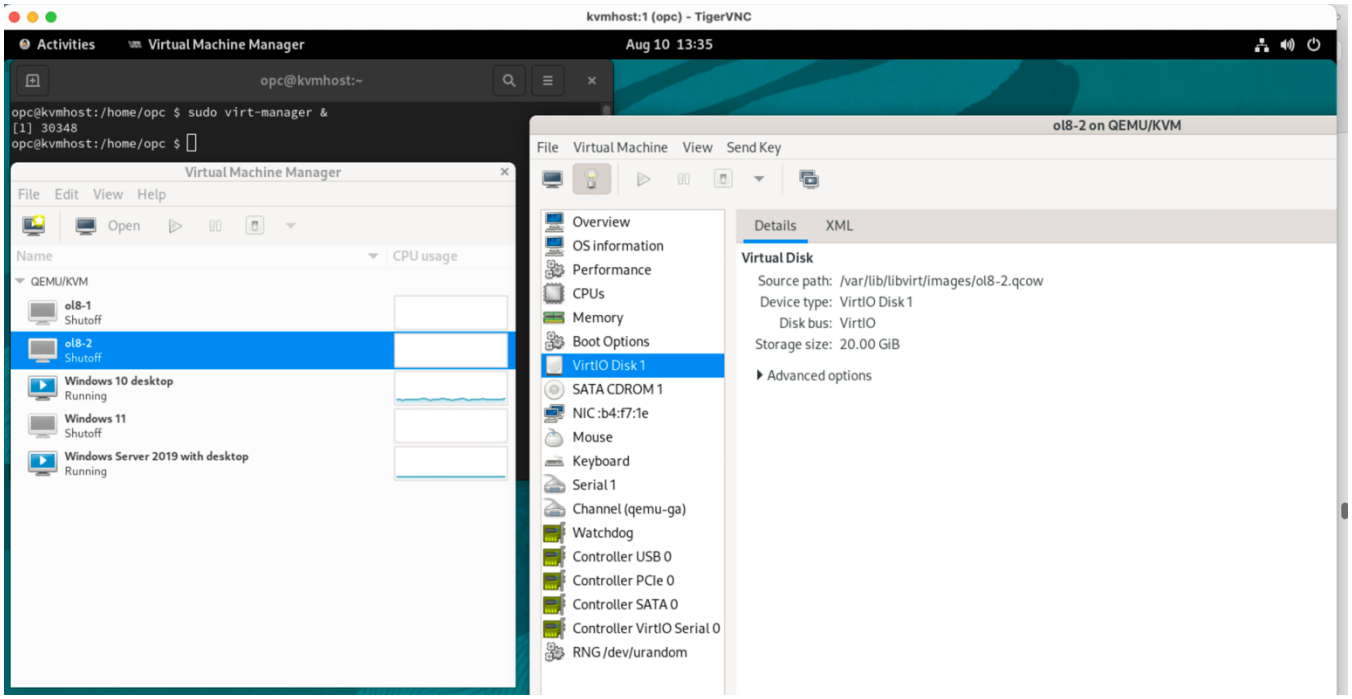
Start virt-manager GUI:
From a terminal on Oracle Linux KVM host, run:

[opc@kvmhost ~]$ sudo virt-manager

The pictures below show the virt-manager GUI.



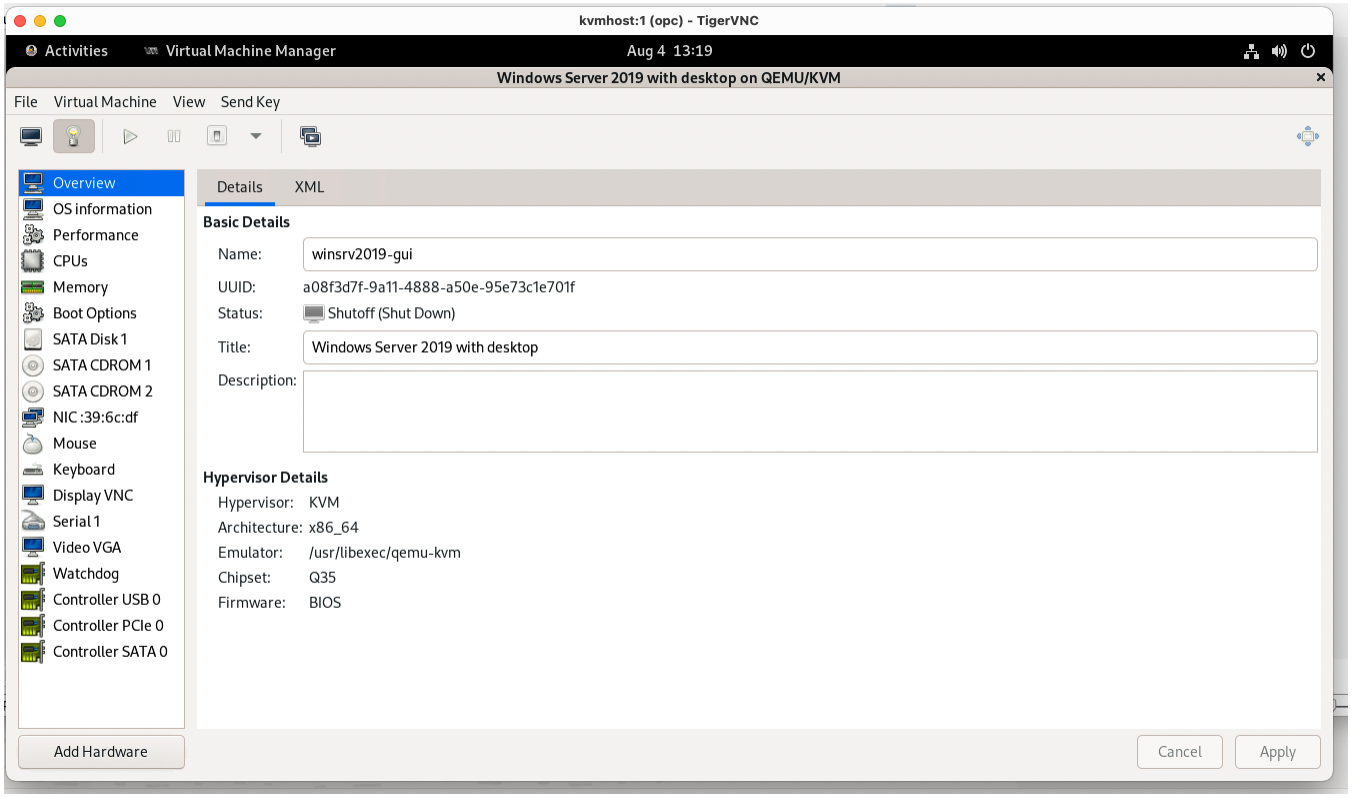
Use virt-manager to display detail of a guest VM (or guest OS):

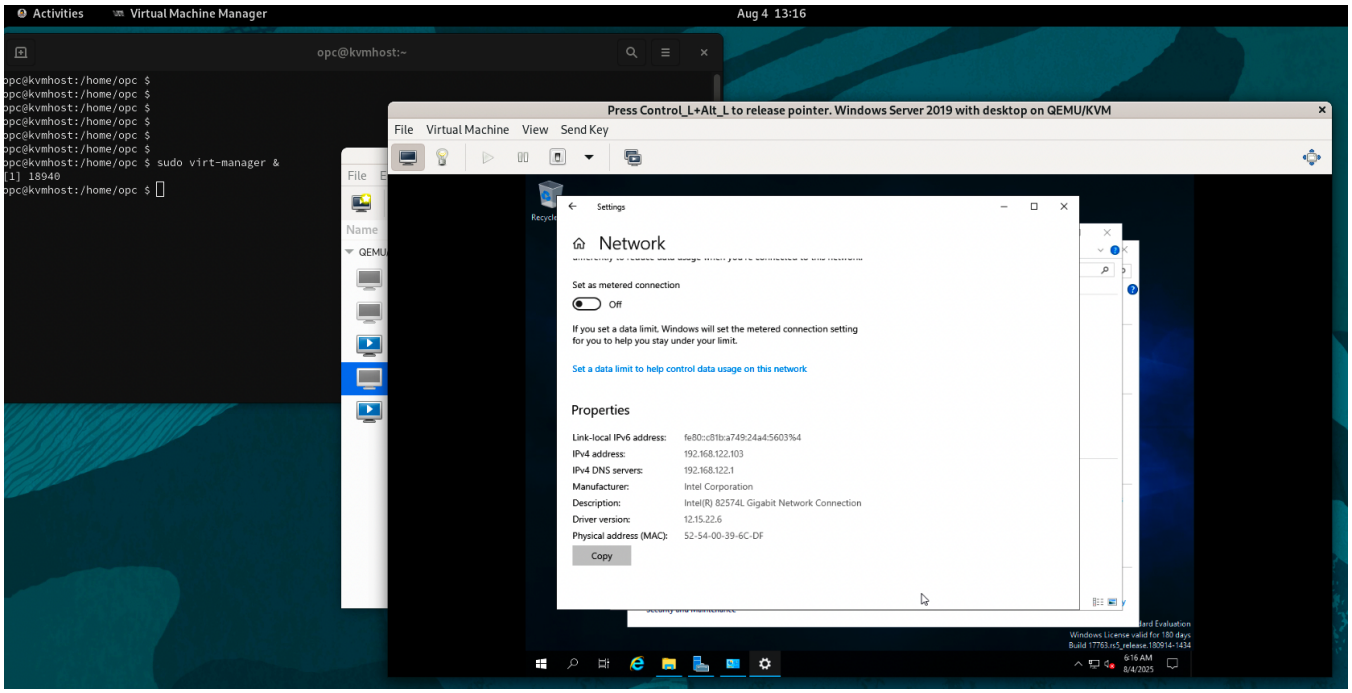


Install Windows Server 2019 guest OS on KVM using virt-manager

To install Windows Server 2019 on KVM host, ensure the following during configuration of the VM:

- Firmware is set to BIOS.
- Boot disk has 80GB or more.
- For GUI select Windows server 2019 Desktop Experience option during installation.
- Perform custom installation





Install Windows 11 guest OS on KVM using virt-manager

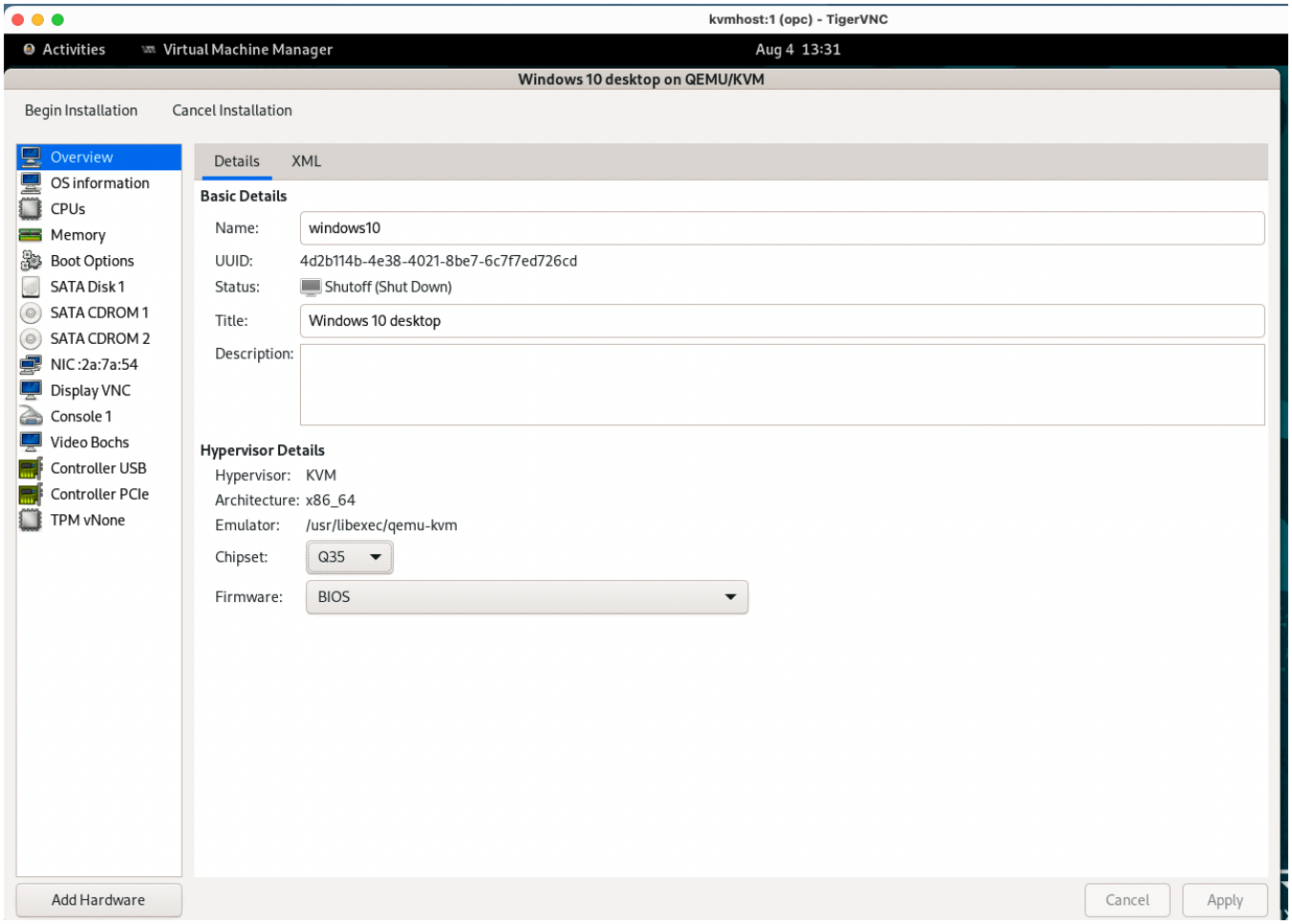
Most videos suggest using virt-manager instead of virt-install to install Windows guest on KVM.
To install Windows 11 guest OS on KVM see, https://www.youtube.com/watch?v=7tqKBy9r9b4
The installation of Windows 11 failed on this server because the CPU was too old. The Bare Metal server is BM.Standard2.52 with Intel Xeon Platinum 8167M, Base frequency 2.0 GHz, and does not meet the minimum requirements for Windows 11.

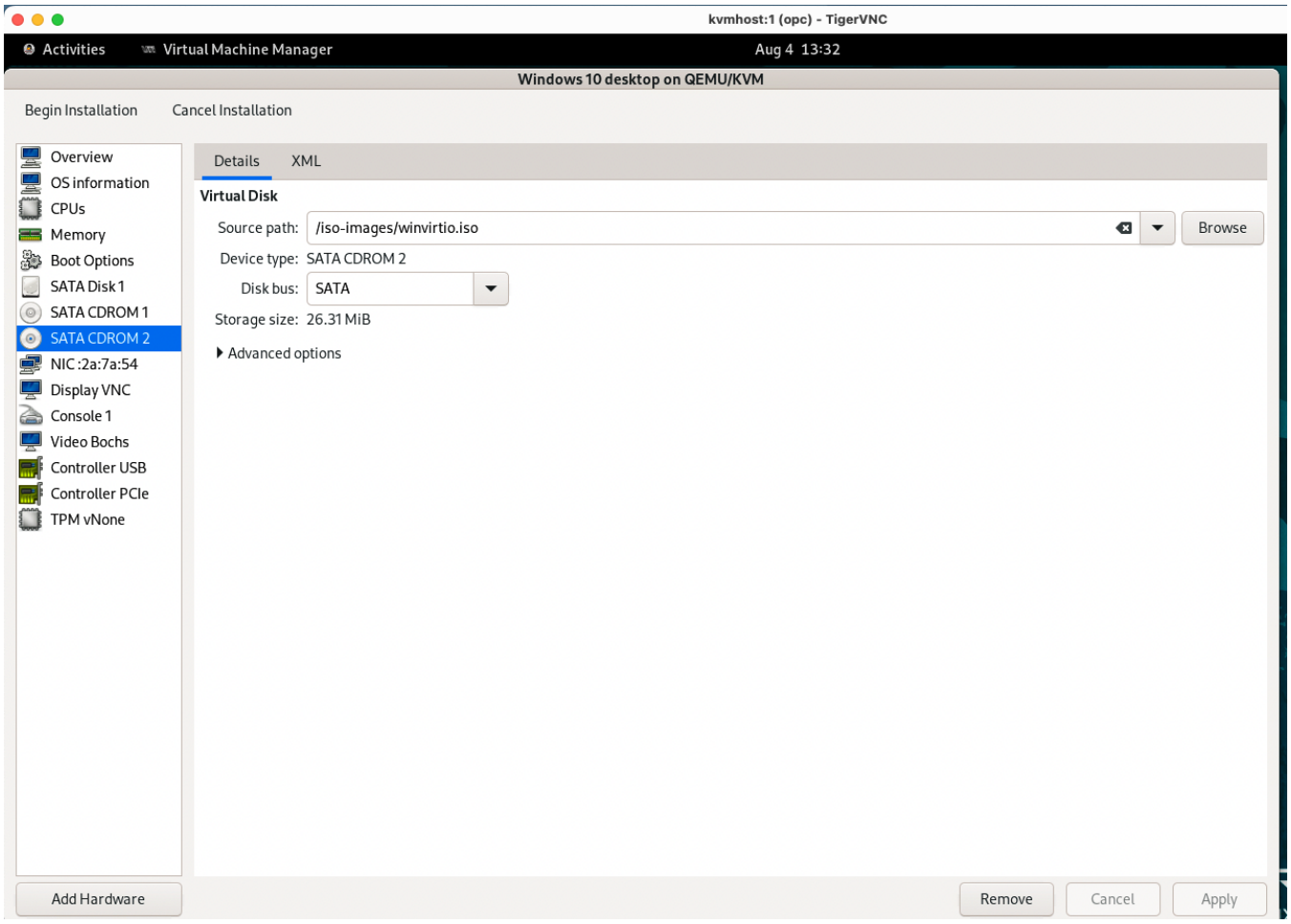
Install Windows 10 guest OS on KVM using virt-manager

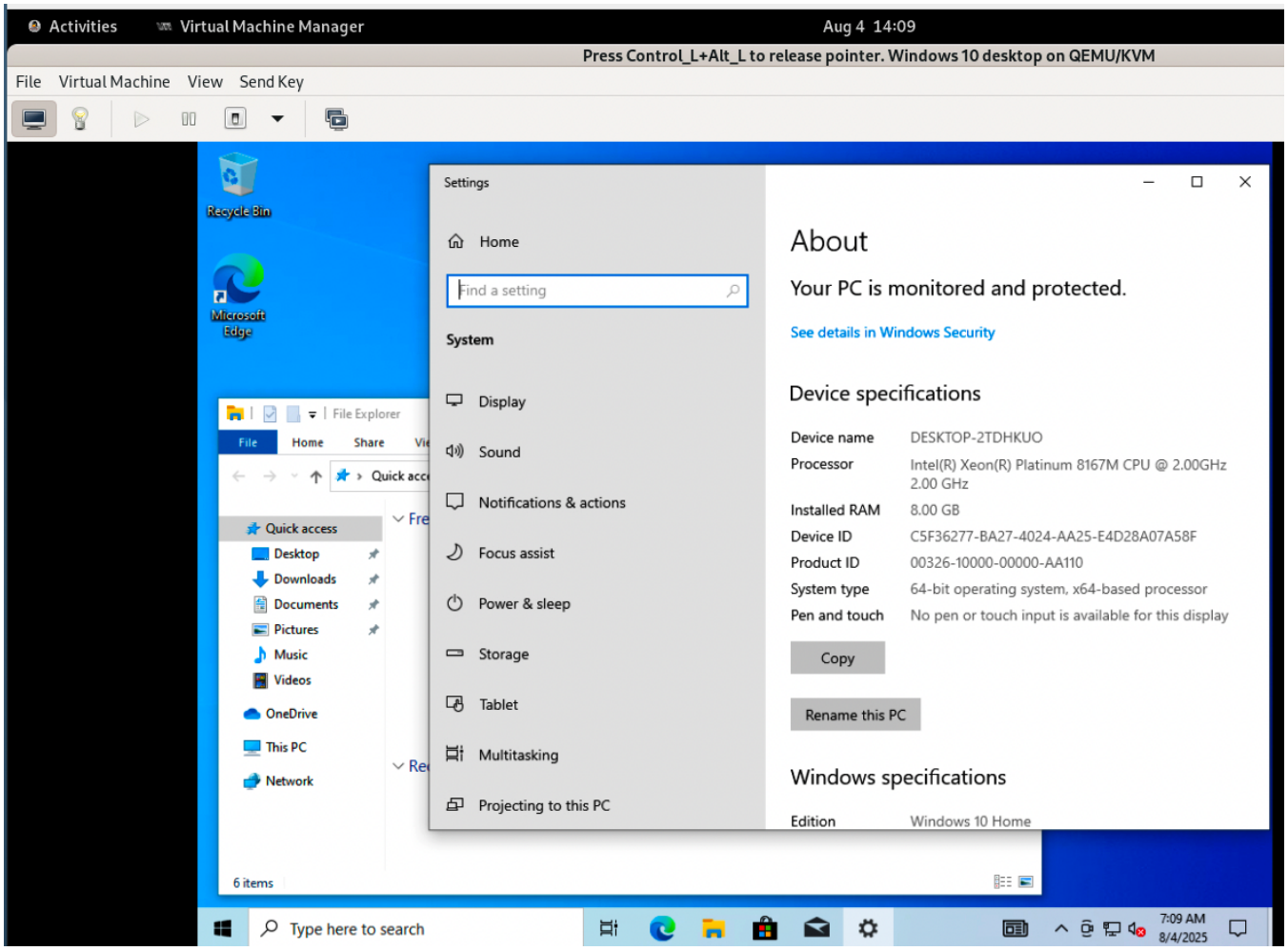
To install Windows 10 on KVM host, ensure the following:

- Firmware is set to BIOS.
- Boot disk has 80GB or more.
- Perform custom installation

The pictures below show Windows 10 creation detail using virt-manager.







IMPORT KVM GUEST VMS USING QEMU-IMG AND VIRT-MANAGER

QCOW2 is a storage format for virtual disks. QCOW stands for QEMU copy-on-write.
You can import an existing qcow2 disk into KVM and create a new Guest VM from this imported disk.
https://docs.google.com/document/d/1X8TaBP1v_rh8e2QXDGFmkmzoPB13Pvp2V0FQ630VCsU/edit
and see https://access.redhat.com/solutions/641193

Migrate from VMWare to KVM by converting vmdk to qemu
Convert the disk image using this command: *sudo qemu-img convert -O qcow2 source.vmdk target.qcow2*

[opc@kvmhost ~]$ sudo qemu-img convert -O qcow2 CentOS_7.9.vmdk Centos79-new.qcow2
https://foswiki.org/Support/HowToRunVirtualMachineImageOnKvmQemu

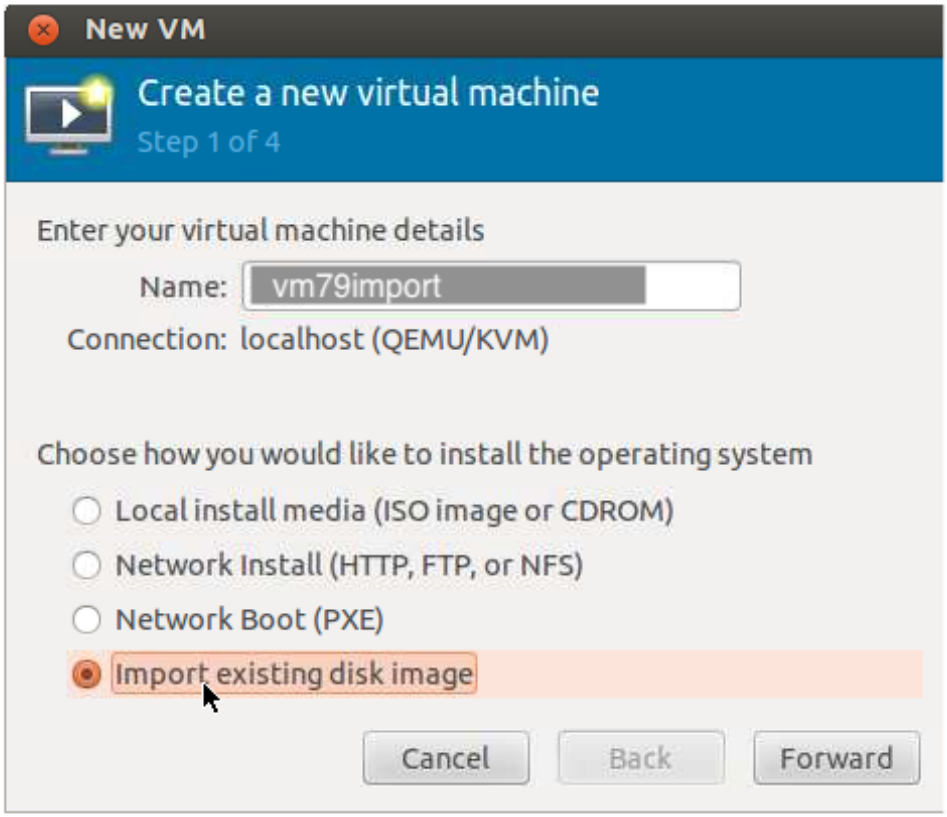
In this example I have used an existing qcow2 disk images from an older Oracle Linux 7.9 VM. This is image is called */vm/vmol79.*

[opc@kvmhost ~]$ ls -lh vmol79
-rw-------. 1 qemu qemu 21G Jul 30 11:27 vmol79

To import the vm, start virt-manager and create new VM from existing disk. See virt-manager section in this document.

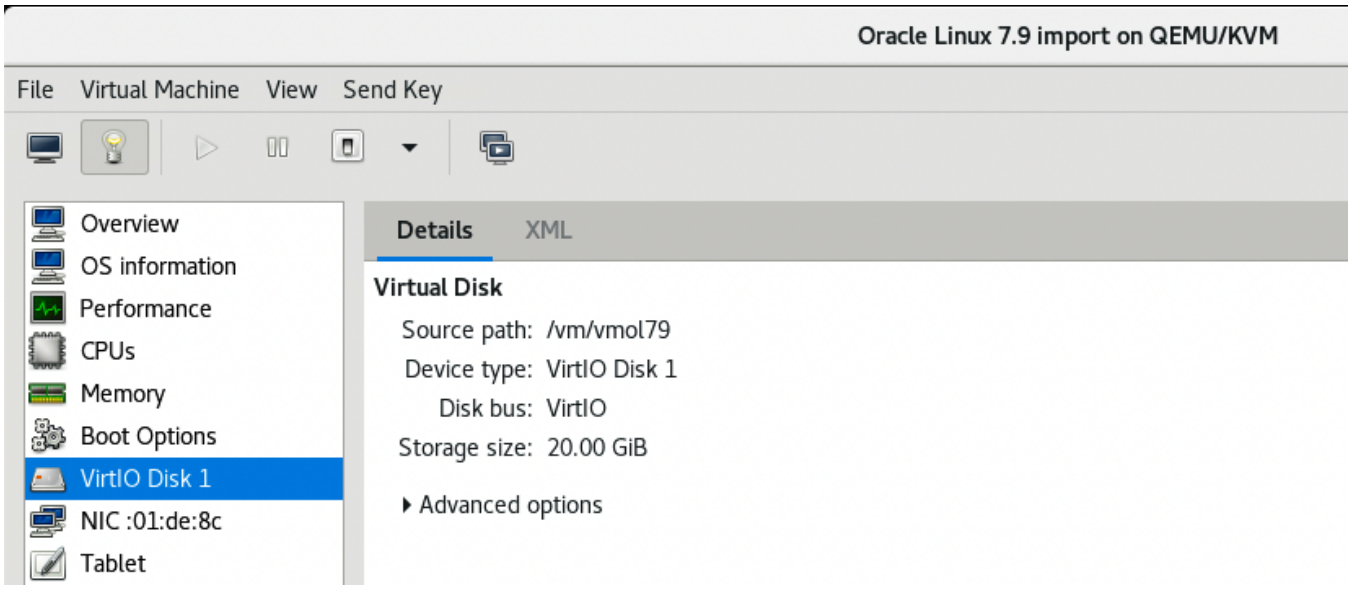
Provision a VM from virt-manager GUI interface

Step 1: Create a new VM from [Import existing disk image] option

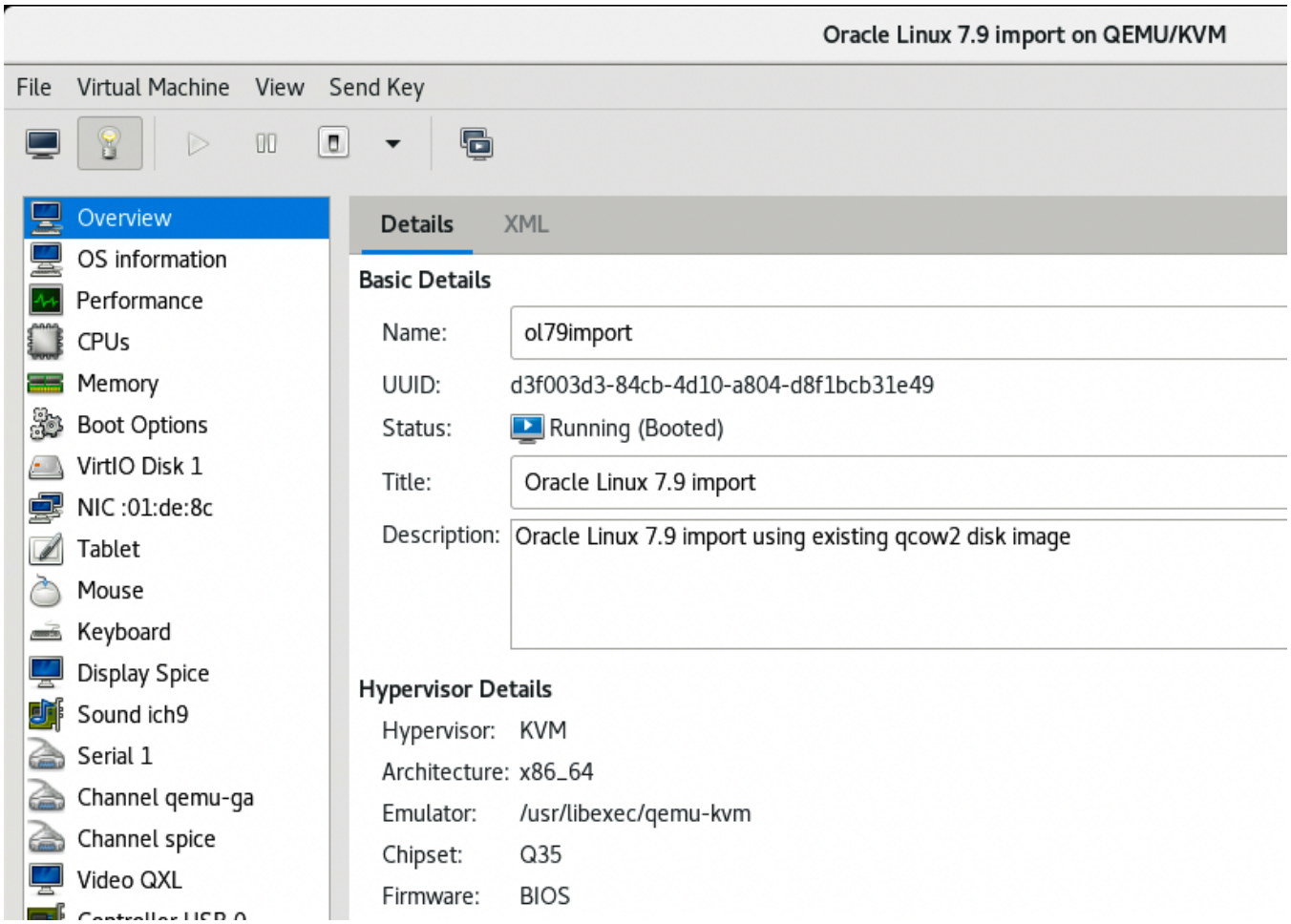


Step 2: Choose your qcow2 image

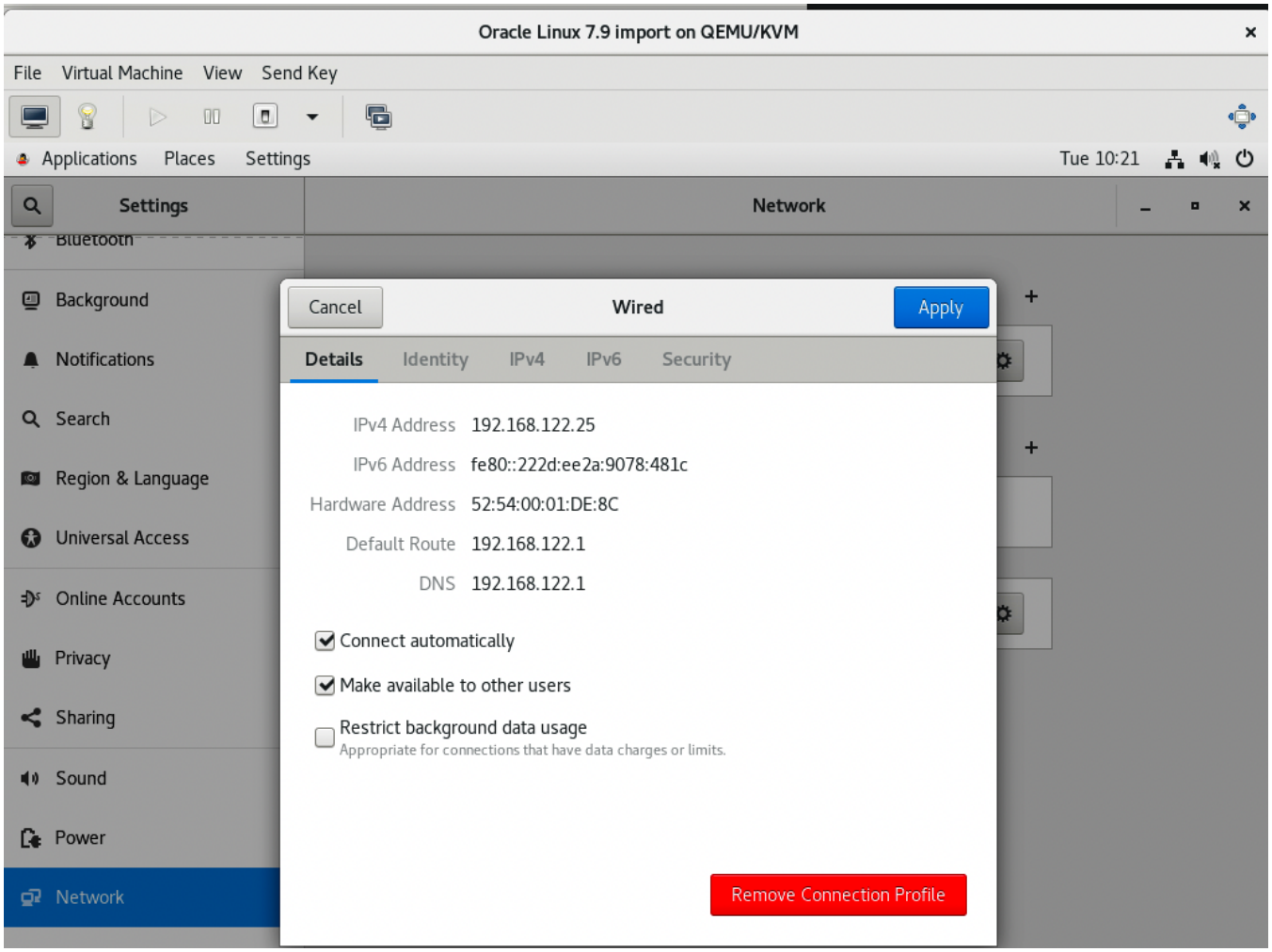
This picture shows detail about the VM boot disk:



This picture shows detail about the VM:



This picture shows detail about the VM networking:



LICENSING

Running Windows Server as KVM Guest OS requires a license. Customers are to ensure they have proper licensing. See https://docs.oracle.com/en-us/iaas/Content/Compute/References/licensing-options-microsoft.htm

Oracle Linux KVM or Oracle VM Server may be used as hard partitioning technology only as described in the following documents: Oracle Linux KVM, only if specific cores are allocated per the following document: https://www.oracle.com/a/ocom/docs/linux/ol-kvm-hard-partitioning.pdf

Bring Your Own License (BYOL): You can use your existing Microsoft licenses (e.g., Windows Server, SQL Server) on OCI KVM Bare Metal if your licenses have applicable Software Assurance and License Mobility rights.

License Included (LI): OCI typically provides "license included" options for its Compute virtual machines, **not** for Bare Metal or custom KVM deployments. For KVM Bare Metal, you usually need to bring your own licenses.

KVM ADMINISTRATION

See Oracle Linux KVM User's Guide: https://docs.oracle.com/en/operating-systems/oracle-linux/kvm-user/kvm-AboutOracleLinuxKVM.html
virsh commands: https://www.redhat.com/sysadmin/virsh-subcommands

For admin tasks, use virt-manager or cockpit.

CLI virsh can also be used: List of virsh commands: https://www.redhat.com/sysadmin/virsh-subcommands

To start a VM, run the following command:

[opc@kvmhost ~]$ sudo virsh start ol8-2

Access the vm console from the kvm server:

[opc@kvmhost ~]$ virsh console ol8-2
Escape character is ^] (Ctrl +])

To reboot a VM, run the following command:

[opc@kvmhost ~]$ sudo virsh reboot ol8-2

Lists all running domains (VMs):

```
[opc@kvmhost ~]$ sudo virsh list --all
 Id   Name      State
--------------------------------
 -    ol8-1     shut off
 -    ol8-2     shut off
 -    win11     shut off
 -    windows10 shut off
 -    winsrv2019-gui shut off
```

Show VM information:

```
[opc@kvmhost ~]$ sudo virsh dominfo ol8-2
Id:             1
Name:           ol8-2
UUID:           8078bdf6-513c-4f43-b12f-89c32b5f952c
OS Type:        hvm
State:          running
CPU(s):         2
CPU time:       47.6s
Max memory:     4194304 KiB
Used memory:    4194304 KiB
Persistent:     yes
Autostart:      disable
Managed save:   no
Security model: selinux
Security DOI:   0
Security label: system_u:system_r:svirt_t:s0:c341,c875 (enforcing)
```

CPU en mem info of VM Guest:

```
[opc@kvmhost ~]$ sudo virsh vcpuinfo ol8-2
VCPU:           0
CPU:            N/A
```

State: N/A
CPU time N/A
CPU Affinity: yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy...

VCPU: 1
CPU: N/A
State: N/A
CPU time N/A
CPU Affinity: yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy...

IP Address info of VM Guest:

```
[opc@kvmhost ~]$  sudo virsh domifaddr ol8-2
Name MAC address Protocol Address
-------------------------------------------------------------------
vnet0 52:54:00:b4:f7:1e ipv4 192.168.122.27/24

[opc@kvmhost ~]$  sudo ip -d link show
```

modify prompt for KVM host or guest VM Oracle Linux: Add to ~/.bashrc

```
PS1=$LOGNAME@`uname -n`:'$PWD $ ';export PS1
```

Delete a guest VM:

```
[opc@kvmhost ~]$  sudo virsh destroy ol8-2
```

The dumpxml subcommand dumps the XML configuration for a given domain.

```
[opc@kvmhost ~]$  sudo virsh dumpxml ol8-2
```

Hypervisor information:

```
[opc@kvmhost ~]$  sudo virsh sysinfo
[opc@kvmhost ~]$  sudo virsh version
```

Ensure KVM modules are loaded:

```
[opc@kvmhost ~]$ sudo modinfo kvm |grep modules
filename: /lib/modules/6.12.0-101.33.4.3.el9uek.x86_64/kernel/arch/x86/kvm/kvm.ko.xz

[opc@kvmhost ~]$  rpm -q libvirt
libvirt-8.0.0-23.2.0.1.module+el8.10.0+90363+955e9a81.x86_64

[opc@kvmhost ~]$  sudo systemctl status libvirtd

[opc@kvmhost ~]$  cat /etc/resolv.conf
nameserver 169.254.169.254
```

Restart KVM guest network service:

```
[opc@kvmhost ~]$  sudo service network restart
```

Reboot KVM Host

if you reboot kvm host, does the vm also reboot gracefully? No, No, KVM does not automatically shut down guests gracefully when the host reboots; the guest VMs are instead abruptly terminated, which is similar to pulling the power plug and risks data corruption. To achieve a graceful shutdown, you must manually shut down each virtual machine before rebooting the host or implement a custom script to send shutdown signals to the guest VMs and wait for them to complete before proceeding with the host shutdown.

Form the login prompt of the KVM host, stop the VMs first:

[opc@kvmhost ~]$ sudo virsh shutdown <VM_Name>
[opc@kvmhost ~]$ sudo vistsh list –all
Once all VMs are stopped, then reboot the KVM host:
[opc@kvmhost ~]$ sudo init 6 or do sudo shutdown -r now

You can write a script that runs before the host's shutdown process begins.

RESOURCES

Oracle Linux KVM user guide:
https://docs.oracle.com/en-us/iaas/oracle-linux/kvm/index.htm#introduction and see
https://docs.oracle.com/en/operating-systems/oracle-linux/9/kvm-user/

Note: Marketplace image for KVM is End of Life. It is recommended to perform the manual KVM deployment.
https://cloudmarketplace.oracle.com/marketplace/en_US/listing/48972344

Supported Guest VMs (See user guide):
https://docs.oracle.com/en/operating-systems/oracle-linux/9/kvm-user/kvm-GuestOperatingSystemRequirements.html#support

Installation Guide for KVM on OCI user guide:
https://docs.oracle.com/en/operating-systems/oracle-linux/kvm-user/kvm-AboutOracleLinuxKVM.html and
https://docs.oracle.com/en/operating-systems/oracle-linux/kvm-user/OL-KVM-USER.pdf

---oOo---

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.
Outside North America, find your local office at oracle.com/contact.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Oracle Cloud Infrastructure Bare Metal and KVM
August 2525
Author: [OPTIONAL]
Contributing Authors: [OPTIONAL]