

OCI-How to Install and Configure KVM Guest in Oracle Cloud Infrastructure

May 2023, Version 1.1

Sharath Kumar
EMEA Senior Cloud Specialist Engineer

Copyright © 2023, Oracle and/or its affiliates
Public

Prerequisites

You have proper VCN ready and fully configured the related security rules like allow SSH and ensure KVM hypervisor can access the internet.

You understand how to install an operating system as a guest, or you know how to copy a virtual disk image between systems.

Steps

Launch the KVM hypervisor host instance

[1] Review the OCI public documentation(<https://docs.oracle.com/en-us/iaas/Content/Compute/Tasks/launchinginstance.htm>) for current options and features you can use in the environment then select Oracle Images "Oracle Linux KVM Image and a BM shape for the KVM hypervisor host based on the workload requirements and provision it.

Browse all images

An image is a template of a virtual hard drive that determines the operating system and other software for an instance.

Image source

Oracle images

Compartment

SharathKumar

oraseemeaocids (root)/COMPUTE/SharathKumar

Choose from Oracle enterprise images and solutions enabled for Oracle Cloud Infrastructure.

kvm		
App name	Publisher	Price
<input checked="" type="checkbox"/> Oracle Linux KVM Image	Oracle	Free
<input type="checkbox"/> Oracle Linux KVM Image (Autonomous Linux)	Oracle	Free
1 Selected		Showing 2 Items < 1 c

Agreement for Oracle app **Oracle Linux KVM Image**

☒ I have reviewed and accept the [Oracle Standard Terms and Restrictions](#)

Select image

Cancel

Create compute instance

ORACLE
Linux

Oracle Linux KVM Image
Simplified Oracle Linux KVM hypervisor deployment

Change image

Shape



BM.Standard2.52
Bare metal machine, 52 core OCPU, 768 GB memory, 50 Gbps network bandwidth

Change shape

By default, the KVM image has included required KVM packages. You can also check if those packages are properly installed and configured:

```
[opc@j-kvm-host ~]$ rpm -q libvirt
```

```
libvirt-3.9.0-14.el7_5.5.x86_64
```

```
[opc@j-kvm-host ~]$ systemctl status libvirtd
```

- libvirtd.service - Virtualization daemon

KVM related modules:

```
[root@j-kvm-host opc]# modinfo kvm_intel |grep modules
```

```
filename: /lib/modules/4.1.12-124.15.2.el7uek.x86_64/kernel/arch/x86/kvm/kvm-intel.ko
```

```
[root@j-kvm-host opc]# modinfo kvm |grep modules
```

```
filename: /lib/modules/4.1.12-124.15.2.el7uek.x86_64/kernel/arch/x86/kvm/kvm.ko
```

The intel_iommu for SR-IOV pass-through (pt) mode in the kernel has configured enabled:

"intel_iommu=on" has added into GRUB command line:

```
[root@j-kvm-host opc]# cat /etc/default/grub
```

```
...
```

```
GRUB_CMDLINE_LINUX="crashkernel=auto LANG=en_US.UTF-8 transparent_hugepage=never console=tty0  
console=ttyS0,9600 rd.luks=0 rd.lvm=0 rd.md=0 rd.dm=0 netroot=iscsi:169.254.0.2:::1:ign.2015-  
02.oracle.boot:uefi iscsi_param=node.session.timeo.replacement_timeout=6000 net.ifnames=1  
nvme_core.shutdown_timeout=10 ipmi_si.tryacpi=0 ipmi_si.trydmi=0 ipmi_si.trydefaults=0 network-  
config=e2NvbmZpZzogZGlzYWJsZWR9Cg== intel_iommu=on" >>>>>>>>
```

```
[root@j-kvm-host opc]# dmesg | grep -e DMAR -e IOMMU
```

```
[ 0.000000] ACPI: DMAR 0x0000000006D1B9EC8 000260 (v01 ORACLE X7-2C 46040600 INTL 20091013)
```

```
[ 0.000000] Intel-IOMMU: enabled
```

Tuned service has been enabled for performance optimization:

```
[root@j-kvm-host opc]# systemctl status tuned
```

tuned.service - Dynamic System Tuning Daemon

Loaded: loaded (/usr/lib/systemd/system/tuned.service; enabled; vendor preset: enabled)

Active: active (running) since Wed 2019-03-06 04:31:29 GMT; 2h 8min ago

Docs: man:tuned(8)

man:tuned.conf(5)

man:tuned-adm(8)

Main PID: 3757 (tuned)

CGroup: /system.slice/tuned.service

└─3757 /usr/bin/python -Es /usr/sbin/tuned -l -

oci-utils package has been installed and ocid.service is enabled by default:

```
[root@j-kvm-host opc]# rpm -q oci-utils
```

oci-utils-o.6-34.el7.noarch

```
[root@j-kvm-host opc]# systemctl status ocid.service
```



ocid.service - Oracle Cloud Infrastructure utilities daemon

Loaded: loaded (/etc/systemd/system/ocid.service; enabled; vendor preset: enabled)

Attach block volumes for KVM hypervisor and Guest further usage

Follow below document steps to attach two block volumes(one for KVM host and another for KVM Guest) and mount them:

<https://docs.oracle.com/en-us/iaas/Content/Block/Tasks/attachingavolume.htm>

Attach Block Volume				
 BV ATTACHED	J KVM Host OCID: ...vxkn5a Show Copy	Attachment Type: iscsi Attachment Access: Read/Write Block Volume Compartment: <input type="text"/>	Size: 1.0 TB Device Path: - ⓘ	In-transit Encryption: Disabled
 BV ATTACHED	J KVM Guest OCID: ...5bu6ga Show Copy	Attachment Type: iscsi Attachment Access: Read/Write Block Volume Compartment: <input type="text"/>	Size: 500.0 GB Device Path: - ⓘ	In-transit Encryption: Disabled

Examine Storage: In the KVM host, use the `oci-iscsi-config -s` command to display attached Block Volumes:

```
[root@j-kvm-host opc]# oci-iscsi-config -s
```

```
[root@j-kvm-host opc]# oci-iscsi-config -s
```

Currently attached iSCSI devices:

Target iqn.2015-12.com.oracleiaas:d3b517a5-5f28-4914-85b0-d26ca1860bb0

Persistent portal: 169.254.2.3:3260

Current portal: 169.254.2.3:3260

State: LOGGED_IN

Attached device: sdc

Size: 500G

File system type: Unknown

Mountpoint: Not mounted

Target iqn.2015-12.com.oracleiaas:bbd692ef-9822-4b43-9206-34913a5be5b4

Persistent portal: 169.254.2.2:3260

Current portal: 169.254.2.2:3260

State: LOGGED_IN

Attached device: sdb

Size: 1T

File system type: Unknown

Mountpoint: Not mounted

Create lvm on the block volume and put the guest ISO on it:

```
[root@j-kvm-host opc]# ls -hl /mnt/myhostlv1/
```

total 3.2G

drwx-----. 2 root root 16K 14:04 lost+found

-rw-r--r--. 1 qemu qemu 3.2G 14:21 OracleLinux-R9-U0-Server-x86_64-dvd.iso

Create a secondary vNIC on the KVM hypervisor host for the KVM guest to use

Follow below document steps to add the second vNIC:

OCI: How to configure a secondary VNICS in OL7 Instance via oci-network-config ([Doc ID 2487934.1](#))

Note down the **"MAC Address"** and **"IP Address"** information here which will be used in later.

NIC 0

VNIC

ATTACHED

J KVM Host (Primary VNIC)

OCID: ...6x4yta Show Copy

Attached: Wed, 06 Mar 2019 12:56:05 GMT

Compartment:

Private IP Address: 10.0.0.45

Fully Qualified Domain Name: j-kvm-host... Show Copy

Public IP Address:

NIC 1

VNIC

ATTACHED

KVM Guest 1 VNIC

OCID: ...4cajq Show Copy

Attached: Wed, 06 Mar 2019 13:31:03 GMT

Compartment:

Private IP Address: 10.0.0.46

Fully Qualified Domain Name: Unavailable

Public IP Address:

Examine VNICs: In the KVM host, use the command `sudo oci-network-config -s` to display the VNICs.

[root@j-kvm-host opc]# **oci-network-config -s | column -t**

```
[root@j-kvm-host opc]# oci-network-config -s | column -t
CONFIG ADDR SPREFIX SBITS VIRT RT NS IND IFACE VLTAG VLAN STATE MAC VNIC
- 10.0.0.45 10.0.0.0 24 10.0.0.1 - 1 eno2 0 - UP 00:10:e0:e5:2e:84
ocid1.vnic.oc1.iad.abuwcljrol5eyleobsqjcaqsq5vbl6vx17zed314h2q7oewrif642n6x4yta
ADD 10.0.0.46 10.0.0.0 24 10.0.0.1 - 2 eno3d1 0 - UP 00:10:e0:e5:2e:85
ocid1.vnic.oc1.iad.abuwcljrm4osymxnrrnnfrmlonzo6rgksskva3q7bxcise2jxdofph4cajqc

[root@j-kvm-host images]# ifconfig

eno2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9000
inet 10.0.0.45 netmask 255.255.255.0 broadcast 10.0.0.255
ether xx.xx.xx.xx txqueuelen 1000 (Ethernet)
RX packets 3055676 bytes 4011517595 (3.7 GiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1931520 bytes 10300725519 (9.5 GiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eno3d1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
ether <MAC ADDR> txqueuelen 1000 (Ethernet)
RX packets 60 bytes 3832 (3.7 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
```

Note: -

If more than one secondary VNIC's are attached, `oci-network-config` utility will not add IFACE Name and it will be empty. Check output of command "`oci-network-config -s | column -t`" and IFACE column will be empty for additional secondary NIC. Please be noted for the first secondary VNIC it will be proper, problem can be witnessed for subsequent secondary VNIC addition.

KVM instance cannot be created using that interface, so IFACE name has to be manually updated.

```
#ip link add link <physical nic interface name> name <new vnic name> address <mac address> type vlan id <vlan id>
```

Create the KVM Guest VM

There are two ways to install KVM guest:

virsh-install command

This article will mainly focus on this method.

Use oci-kvm

oci-kvm is a tool provided by Oracle KVM image to install and remove KVM guests on OCI instances. This article won't focus on this method but will give an example usages for oci-kvm:

```
# oci-kvm create -D J_KVM_GUEST1 -V --vcpu 4 --memory 8192 --boot cdrom,hd --location /u01/OracleLinux-R6-U0-Server-x86_64-dvd.iso --nographics --console pty,target_type=virtio --noautoconsole --os-variant=rhel6 --extra-args "console=tty0 console=ttyS0,115200n8 serial"
```

virsh-install command

Create the KVM guest via virsh-install command. Create the domain, attach the network device following the appropriate network type which will be hostdev for KVM BM hypervisor and direct networking for KVM VM hypervisor:

```
# virt-install --name=J_KVM_GUEST1 --memory=8196 --vcpus=1 --location=/mnt/myhostlv1/OracleLinux-R6-U0-Server-x86_64-dvd.iso --disk /mnt/myguestlv1/images/J_KVM_GUEST1_ol6.qcow2,device=disk,bus=virtio,size=8 --network type=direct,source=en03d1,source_mode=passthrough,mac=00:10:E0:E5:2E:85 --os-type=linux --os-variant=rhel6 --nographics --console pty,target_type=serial --extra-args='console=tty0 console=ttyS0,115200n8 serial'
```

After KVM guest is created, check network connection:

From hypervisor:

```
[root@j-kvm-host opc]# virsh domiflist J_KVM_GUEST1
```

```
Interface Type Source Model MAC
```

```
-----
```

```
macvtap0 direct eno3d1 virtio 00:10:e0:e5:2e:85
```

```
[root@j-kvm-host opc]# ip -d link show macvtap0
```

```
57: macvtap0@eno3d1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN mode DEFAULT group default qlen 500
```

```
link/ether <MAC ADDR> brd ff:ff:ff:ff:ff:ff promiscuity 0
```

```
macvtap mode passthru addrngenmode eui64 numtxqueues 1 numrxqueues 1
```

```
[root@j-kvm-host opc]# virsh dumpxml J_KVM_GUEST1 |grep "interface type" -A4
```

```
interface type='direct'>
```

```
<mac address='<MAC ADDR>'>/>
```

```
<source dev='eno3d1' mode='passthrough'>/>
```

```
<target dev='macvtap0'>/>
```

```
<model type='virtio'>/>
```

From KVM guest

```
[root@localhost ~]# ifconfig
```

```
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
```

Associating your BMCS vNIC with your KVM VM by editing /etc/sysconfig/network-scripts/ifcfg-etho and resolv.conf and add the below information:

```
root@localhost ~]# cat /etc/sysconfig/network-scripts/ifcfg-etho
```

```
HWADDR="<MAC ADDR>" >>>>>>>>BMCS second VNIC MAC
DEVICE="etho"
BOOTPROTO=static
IPADDR=10.0.0.46 >>>>>>>>BMCS second VNIC IP Address
NETMASK=255.255.255.0
GATEWAY=10.0.0.1
#NM_CONTROLLED="yes"
ONBOOT="yes"
```

```
[root@localhost ~]# cat /etc/resolv.conf
```

```
nameserver 169.254.169.254
```

Restart KVM guest network service:

```
[root@localhost ~]# service network restart
```

```
Shutting down loopback interface: [ OK ]
IPv6 over IPv4 tunneling driver
sit0: Disabled Privacy Extensions
Bringing up loopback interface: lo: Disabled Privacy Extensions
[ OK ]
```

```
[root@localhost ~]# ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
```

Do ping test to the hypervisor and Internet from KVM guest:

```
root@localhost ~]# ping 10.0.0.45
```

```
PING 10.0.0.45 (10.0.0.45) 56(84) bytes of data.
64 bytes from 10.0.0.45: icmp_seq=1 ttl=64 time=0.828 ms
64 bytes from 10.0.0.45: icmp_seq=2 ttl=64 time=0.366 ms
--- 10.0.0.45 ping statistics ---
```


2 packets transmitted, 2 received, 0% packet loss, time 1921ms
rtt min/avg/max/mdev = 0.366/0.597/0.828/0.231 ms

[root@localhost ~]# **ping 8.8.8.8**

PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

64 bytes from 8.8.8.8: icmp_seq=1 ttl=123 time=14.9 ms

64 bytes from 8.8.8.8: icmp_seq=2 ttl=123 time=14.5 ms

[root@localhost ~]# **ping www.google.com**

PING www.google.com (172.217.164.68) 56(84) bytes of data.

64 bytes from atl26s18-in-f4.1e100.net (172.217.164.68): icmp_seq=1 ttl=54 time=28.3 ms

Do ping test from hypervisor to the KVM guest:

[root@j-kvm-host opc]# **ping 10.0.0.46**

PING 10.0.0.46 (10.0.0.46) 56(84) bytes of data.

64 bytes from 10.0.0.46: icmp_seq=1 ttl=64 time=1.01 ms

64 bytes from 10.0.0.46: icmp_seq=2 ttl=64 time=0.351 ms

Attaching the secondary vnic after KVM guest installation complete

As aforementioned, you can also configure the network after KVM guest installation complete with --nonetwork parameter. The steps are as follows:

Virsh install the KVM guest:

```
# virt-install --name=J_KVM_GUEST1 --memory=8196 --vcpus=1 --location=/mnt/myhostlv1/OracleLinux-R6-U0-Server-x86_64-dvd.iso --disk /mnt/myguelstlv1/images/J_KVM_GUEST1_ol6.qcow2,device=disk,bus=virtio,size=8 --nonetwork --os-type=linux --os-variant=rhel6 --nographics --console pty,target_type=serial --extra-args='console=tty0 console=ttyS0,115200n8 serial'
```

Once installation done and make sure there is no network available in the guest, destroying the VM:

```
# virsh destroy J_KVM_GUEST1
```

Create a xml file with information of the new added vnic for attaching the secondary vnic:

```
# cat /tmp/kvm-vmcli-vnic-attach.xml
<interface type='direct'>
<mac address='<MAC ADDR>'/>
<source dev='ens4' mode='passthrough'/>
<model type='virtio'/>
</interface>
```

```
# virsh attach-device J_KVM_GUEST1 /tmp/kvm-vmcli-vnic-attach.xml --config
```

Start and login the VM, checking if the new vnic is added (eth0):

```
# virsh start J_KVM_GUEST1
```

Configuring the network of eth0:

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
HWADDR="<MAC ADDR>" >>>>>>>>BMCS second VNIC MAC
BOOTPROTO=static
IPADDR=10.0.0.46 >>>>>>>>BMCS second VNIC IP Address
NETMASK=255.255.255.0
GATEWAY=10.0.0.1
#NM_CONTROLLED="yes"
ONBOOT="yes"
```

Restart KVM guest network service:

```
# service network restart
```

Do the same ping tests as above to check network access.

Connect with us

Visit **oracle.com**. find your local office at: **oracle.com/contact**.

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2023, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

Disclaimer: If you are unsure whether your data sheet needs a disclaimer, read the revenue recognition policy. If you have further questions about your content and the disclaimer requirements, e-mail REVREC_US@oracle.com.