

---

## Creating a KVM virtual machine (CLI method)

If you do not want to use the **virt-manager** GUI application to create a KVM virtual machine, you can use these CLI commands instead.

### About this task

To create a KVM virtual machine, complete the following steps:

### Procedure

1. Create a base qcow2 image for the Central Server and Region Server:  
`qemu-img create -f qcow2 -o cluster_size=2048k /home/sco-base.qcow2 200G`
2. Create the guests:

- a. Run the **virt-install** command with the appropriate parameters. The syntax of the **virt-install** command is as follows:

```
virt-install --connect=qemu:///system \  
--name=sco-base \  
--ram=2048 \  
--vcpus=2 \  
--arch=x86_64 \  
--os-type=linux \  
--os-variant=rhel6 \  
--hvm \  
--virt-type kvm \  
--cdrom=/opt/RHEL6.x-xxxxxxx.x-Server-x86_64-DVD1.iso \  
--disk path=/home/sco-base.qcow2,format=qcow2,bus=virtio \  
--network bridge=br0,model=virtio \  
--accelerate \  
--vnc
```

where

**--arch** Specifies the bits of the guest operating system. Use `x86_64` for IBM® Cloud Orchestrator virtual machines.

**--os-variant**

Specifies the detailed variant of the guest operating system. You can display the full list by running the **man virt-install** command.

**--disk** Specifies the virtual disk or disks for installing the guest operating system. If supported, enable `virtio`.

**--network**

Specifies the bridge device name. If supported, enable `virtio`.

- b. In the `virt-viewer` window, follow the steps in the installation wizard to complete the Red Hat Enterprise Linux installation. Install the operating system with base mode. For operating systems that support `virtio`, customize the partition layout with only `/filesystem`.
3. Configure the image file:
    - a. Log on to the operating system by using the **virt-viewer** command:  
`virtviewer <ID or name>`

For example:

```
virtviewer sco-base
```

- Run the **virsh list --all** command to display a list of all IDs and names.
- b. Edit the `/etc/adjtime` file to ensure that the hardware clock is configured as UTC, as shown in the following example:
 

```
619.737272 1411131873 0.000000
1411131873
UTC
```
  - c. Disable SELinux by editing the `/etc/selinux/config` file and specifying the following value:
 

```
SELINUX=disabled
```
  - d. Disable the firewall:
 

```
service iptables stop
service ip6tables stop
chkconfig iptables off
chkconfig ip6tables off
```
  - e. Edit the `/etc/hosts` file and specify the following value:
 

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 localhost.localdomain localhost
```
  - f. Edit the `/etc/sysconfig/network` file and specify the following value:
 

```
NETWORKING=yes
```
  - g. Edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file and specify the following values:
 

```
# Intel Corporation 82562GT 10/100 Network Connection
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
TYPE=Ethernet
PERSISTENT_DHCLIENT=1
```
  - h. Remove the write net rules:
 

```
mv /lib/udev/write_net_rules /lib/udev/write_net_rules.bak
mv /etc/udev/rules.d/70-persistent-net.rules /etc/udev/rules.d/70-persistent-net.rules.bak
mv /lib/udev/rules.d/75-persistent-net-generator.rules
/lib/udev/rules.d/75-persistent-net-generator.rules.bak
```
  - i. Shut down the virtual machine:
 

```
shutdown -h now
```
4. Create an image file for each server:
 

```
qemu-img create -b <path>/sco-base.qcow2 -f qcow2 -F qcow2 <path>/<SERVER_NAME>.qcow2
```
  5. For each server, create an XML template file called `<SERVER_NAME>.xml` with the following content:
 

```
<domain type='kvm'>
  <name><SERVER_NAME></name>
  <memory unit='KiB'><SERVER_RAM></memory>
  <currentMemory unit='KiB'><SERVER_RAM></currentMemory>
  <vcpu placement='static'><CPU_NUM></vcpu>
  <os>
    <type arch='x86_64'>hvm</type>
    <boot dev='hd'>/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <clock offset='utc'>/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='qcow2' cache='none'>/>
    </disk>
  </devices>
</domain>
```

```

        <source file='IMAGE_PATH' />
        <target dev='vda' bus='virtio' />
    </disk>
    <controller type='ide' index='0'>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1' />
    </controller>
    <controller type='usb' index='0'>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x2' />
    </controller>
    <interface type='bridge'>
        <source bridge='br0' />
        <model type='virtio' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' />
    </interface>
    <serial type='pty'>
        <target port='0' />
    </serial>
    <console type='pty'>
        <target type='serial' port='0' />
    </console>
    <input type='tablet' bus='usb' />
    <input type='mouse' bus='ps2' />
    <graphics type='vnc' port='-1' autoport='yes' />
    <video>
        <model type='cirrus' vram='9216' heads='1' />
        <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0' />
    </video>
    <memballoon model='virtio'>
        <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0' />
    </memballoon>
</devices>
</domain>

```

**Note:** To define the *SERVER\_RAM* and *CPU\_NUM* values for each server, see "Hardware requirements" ([http://www-01.ibm.com/support/knowledgecenter/#!/SS4KMC\\_2.4.0.1/com.ibm.sco.doc\\_2.4/c\\_hw\\_prerequisites.html](http://www-01.ibm.com/support/knowledgecenter/#!/SS4KMC_2.4.0.1/com.ibm.sco.doc_2.4/c_hw_prerequisites.html)) in the *IBM Cloud Orchestrator User's Guide*. Replace *SERVER\_NAME* and *IMAGE\_PATH* with the actual values.

6. Create the virtual machines by using the generated XML files:

```

virsh define <IMAGE_PATH>/<SERVER_NAME>.xml
virsh start <SERVER_NAME>

```

7. Configure the virtual machines for the Deployment Server and Central Servers:

- a. Log on to each virtual machine:

```

virt-viewer <SERVER_NAME>

```

- b. If there is no DHCP server, edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file, as shown in the following example:

```

#Intel Corporation 82562GT 10/100 Network Connection
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
TYPE=Ethernet
IPADDR=192.168.1.3
NETMASK=255.255.255.0
GATEWAY=192.16.1.1

```

- c. If no DNS is in use, update the `/etc/hosts` file as shown in the following example:

```

# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1 localhost.localdomain localhost
192.0.2.3 central-server1.example.com central-server1

```