Installing Cisco Elastic Services Controller on a Kernel-based Virtual Machine

This chapter describes how to install Cisco Elastic Services Controller on a Kernel-based Virtual Machine and includes the following sections:

Installing Cisco Elastic Services Controller in a Kernel-based Virtual Machine

Cisco Elastic Services Controller can be installed in a Kernel-based Virtual Machine. You can install Cisco Elastic services controller in a Kernel-based Virtual Machine using libvirt.

Note
For the KVM host to support UEFI, user may require to install "ovmf" package on the host. Command to install ovmf package differs based on the host operating system.

Preparing to Install Cisco Elastic Services Controller on a Kernel-based Virtual Machine

If you plan to run Cisco Elastic Services Controller on a kernel-based virtual machine, make sure the following are setup:

	Notes
Python 2.7 or 3.x	Installed by default on Linux
python-setuptools	Installed by default on Linux

	Notes

pip	**On RHEL 6.x/7.x::**
	
	`# easy_install pip`
	
	Since the installation using pip compiles source files, the gcc and python development packages are also required on RHEL. To install these packages on RHEL:
	
	`# yum install gcc python-devel`
	
	On RHEL8.x:
	
	`# yum install gcc`
	
	On Ubuntu: Installed by default. Since the installation using pip compiles source files, the gcc and python development packages are also required on Ubuntu. To install these packages on Ubuntu:
	
	`# apt-get install python-dev`

OpenStack clients	`# pip install python-keystoneclient`
	`# pip install python-cinderclient`
	`# pip install python-novaclient`
	`# pip install python-neutronclient`

genisoimage	**On RHEL**:
	
	`# yum install  genisoimage`
	
	On Ubuntu:
	
	`# apt-get install genisoimage`

libvirt and virtinst	For Python 2
	
	On RHEL 6.x:
	
	`# yum install libvirt-python python-virtinst`
	
	On RHEL 7.x:
	
	`# yum install libvirt-python virt-install`
	
	On Ubuntu:
	
	`# apt-get install libvirt-dev`
	`# pip install libvirt-python`
	
	For Python 3:
	
	On Ubuntu
	
	`# sudo apt-get install -y python3-libvirt`
	
	On RHEL
	
	`# yum install -y python3-libvirt`
	
	Virt-Manager
	
	`git clone`
	`https://github.com/virt-manager/virt-manager.git`
	`/usr/share/virt-manager-py3`

	Notes

Other	**On RHEL 8.x**
	
	`# yum install libcurl-devel`
	
	`# yum install openssl-devel`
	
	`# pip3 install pycurl`
	
	`# pip3 install urlgrabber`

Note   libvirt will create the default network automatically.

Installing Elastic Services Controller on a Kernel-Based Virtual Machine

To install standalone Elastic Services Controller (ESC) on a kernel-based virtual machine, do the following:

Procedure

Step 1   Load the variables from the openerc file that contains OpenStack credentials:

```
cat ./openrc.sh
export OS_TENANT_NAME='<OS tenant username>'
export OS_USERNAME='<OS username>'
export OS_PASSWORD='<OS password>'
export OS_AUTH_URL='http://<Openstack Host>:5000/v2.0/'

source ./openrc.sh
```

Step 2   Copy the ESC qcow2 image and the bootvm.py into the kernel-based VM.

Step 3   Boot ESC on a kernel-based VM on the default network that was created when libvirt was installed, use one of the following command:

```
./bootvm.py --user_pass <username>:<password> --user_confd_pass <username>:<password>
--libvirt --image <image_name> esc-vm --net <default network>
```

Step 4   Boot ESC on a kernel-based VM on the default network with static IP, using the following command:

```
./bootvm.py --user_pass <username>:<password> --user_confd_pass <username>:<password>
--libvirt --image <image_name> esc-vm --net <network> --ipaddr <ip_address>
```

Step 5   Get a list of used IP addresses in your network. Use IP addresses that are not in the list for both HA Active/Standby bootvm.py command and for kad_vip. Determine the first 3 octets of your network (i.e. 192.168.122) and pass it in the below command :

```
arp -an | grep 192.168.122
```

Step 6   To install ESC on a kernel-based VM in high availability, use the following command twice for both the HA nodes:

Note   For the second bootvm.py command, use the other HA instance name.

```
./bootvm.py --user_pass <username>:<password> --user_confd_pass <username>:<password>
--libvirt --image <image_name> --ha_mode drbd --gateway_ip <default_gateway_ip_address>
--ipaddr <ip_address>
 --ha_node_list <ha peer ip addresses separated by comma> --kad_vip <vip address> esc-ha-1
 --net <network>
```

Next Steps: Cisco Elastic Services Controller Kernel-based Virtual Machine

Logging in to Cisco Elastic Services Controller Portal

Verifying ESC installation for a Kernel-based Virtual Machine (KVM)

After deploying ESC on a Kernel-based virtual machine, use the following procedure to verify the deployment.

Procedure

Step 1 Check that the ESC VMs have booted using the following command:

```
$ virsh list
```

Step 2 Get the IP address of the ESC VM, using the following command:

```
$ arp -an | grep <ip_address>
```

Step 3 Connect to ESC using SSH and verify the processes are running:

```
$ ssh USERNAME@ESC_IP
```

Troubleshooting Tips

When SSH access is not available, due to network conditions or ESC startup failures, you can connect to ESC through console(if enabled in ESC VM image) or VNC access. To access ESC VM through VNC, do the following:

1. Identify the vnc port.

    ```
    virsh dumpxml 10 | fgrep vnc
    ```

2. Create a ssh tunnel to the local vnc port to allow connection from your remote VNC client.