Install Kubernetes Using Kubespray (Offline)

Last Updated September 11, 2020

This section provides the information that is required to install Kubernetes using Kubespray:

> **Note**
>
> Perform the following steps only if there is no proxy. The disk space that is required for the RPMs and images is at least 20 GB.

Software Requirements

The following table lists the software that is required for the Kubernetes installation:

Requirement	Version
RHEL	7.3 and 7.4
CentOS Linux	7.4.1708 (Core)
Kubernetes	1.10.11, 1.11.7, 1.12.5, 1.13.2 By default, Kubespray-2.6.0 comes with v1.10.4, use v1.10.11 for CVE-2018-1002105 Fix.
Kubespray	Release 2.6.0

Identify and SSH to an Internet Connected Host

Perform the following steps on all the hosts that are part of the cluster:

> **Warning**
>
> Consider the following points:
>
> - These steps are applicable when there is no option for proxy on the deployment servers.
> - At least 5 GB disk space is required for rpm and images.
> - Ensure that the RHEL versions of the internet connected host and offline host are similar

- Ensure that the RHEL versions of the internet connected host and offline host are similar.

Install Yum Packages

Run the following command to install the Yum packages for Kubespray:

```
yum install -y wget epel-release python-pip python36 python-setuptools python36 docker
systemctl start docker
```

Download the RPM Packages for Offline Installation

Follow these steps:

1. Create a directory named rpm.

   ```
   mkdir /root/rpm
   ```

2. Run the following command to download the rpm packages:

   ```
   repotrack -p /root/rpm wget epel-release openssl-libs openssl openssl-devel libsepol-devel libselinux-python
   device-mapper-libs ebtables python-httplib2 openssl curl rsync bash-completion socat unzip python-setuptools
   python-pip python36 python36-libs docker-ce-17.03.2.ce-1.el7.centos.x86_64 docker-ce-selinux-17.03.2.ce-
   1.el7.centos.noarch vsftpd deltarpm python-deltarpm createrepo
   ```

Download the Kubespray Installation Scripts and Untar

Run the following command to clone the repo:

```
git clone git@github.gwd.broadcom.net:ESD/kubespray-offline.git
```

Install the PIP Packages for Kubespray

Navigate to the directory where Kubespray is downloaded and run the following command:

```
cd kubespray-offline
pip install -r requirements.txt
```

Download the PIP Packages for Kubespray Offline Installation

Follow these steps:

1. Create a directory named pip-packages.

   ```
   mkdir /root/pip-packages/
   ```

2. Run the following command to download the packages:

   ```
   pip download -d /root/pip-packages/ -r kubespray-offline/requirements.txt
   ```

Generate the hosts.ini File for Ansible

Follow these steps:

1. Navigate to the **kubespray-offline** directory.

```
cd kubespray-offline
```

2. Copy the sample inventory folder.

```
cp -r inventory/local/ inventory/mycluster

rm -f inventory/mycluster/hosts.ini
```

3. Declare the hosts and generate hosts.ini.

> Note
>
> While running the Kubespray installation, your node hostname is changed to the one generated in the hosts.ini. To skip changing the hostname, you can edit the **roles/bootstrap-os/defaults/main.yml** file. In this file, change **override_system_hostname: true** to **override_system_hostname: false**.

```
##For Single host:
declare -a IPS=(10.131.158.143)

CONFIG_FILE=inventory/mycluster/hosts.ini /usr/bin/python3.6m contrib/inventory_builder/inventory.py ${IPS[@]}
```

```
[all]
node1 ansible_host=10.238.32.23 ip=10.238.32.23
[kube-master]
node1
[kube-node]
node1
[etcd]
node1
[k8s-cluster:children]
kube-node
kube-master
[calico-rr]
[vault]
node1
```

> Note
>
> **More Information: Symptom:** "Config file" creation error with python error code. **Solution:** Remove hosts.ini and try running the config file command again.

Set Up SSH for Passwordless Access to Ansible Host

Copy the public key of the Ansible host from the **~/.ssh/id_rsa.pub** file and paste the key in the **~/.ssh/authorized_keys** file of all other hosts. Alternatively, you can run the following command to set up the SSH:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub $host;
```

Provide Input Values to Kubespray

Edit the **inventory/mycluster/group_vars/k8s-cluster.yml** file to include the directory where the docker images get downloaded.

```
local_release_dir: "/root/docker-images/"
dashboard_enabled: true
registry_enabled: true
local_volume_provisioner_enabled: true
ingress_nginx_enabled: true
```

Download the Kubernetes Docker Images

Run the following command to download the docker images:

```
export ANSIBLE_INVALID_TASK_ATTRIBUTE_FAILED=False
ansible-playbook -i inventory/mycluster/hosts.ini cluster.yml -e download_run_once=true -e download_localhost=true
--tags download --skip-tags upload,upgrade
```

Once the images get downloaded into the **/root/offline-installation/containers** folder, copy the images to the airgapped/offline host. Choose any one of the nodes as the Ansible and Kubernetes master.

Identify and SSH to an Airgapped or Offline Host

Perform the following tasks:

> **Warning**
>
> - Choose any one of the provisioned nodes as the Ansible and Kubernetes master.
> - Ensure that enough disk space is available for the images and rpms.
> - Copy the rpms, pip-packages and container images to the airgapped Ansible host.
> - Copy the kubespray folder from the previous steps which includes the inventory files and settings.

Install Dependencies for FTP Yum Repo

```
cd /root/rpm
rpm -ivh deltarpm* python-deltarpm* createrepo* vsftpd*
```

Set Up the Offline FTP Yum Repo on Ansible Node

Before you perform the following steps, disable SELinux or use **setsebool -P ftpd_full_access 1** command to allow vsftpd.

Follow these steps:

1. Create a directory named localrepo.

   ```
   mkdir -p /var/ftp/pub/localrepo
   ```

2. Copy RPM packages to the directory you created in the earlier step.

2. Copy RPM packages to the directory you created in the earlier step:

```
cp -ar /root/rpm/*.* /var/ftp/pub/localrepo/
```

3. Create a repo.

```
createrepo -v /var/ftp/pub/localrepo/
```

4. Replace with IP address of the Ansible Master.

```
vi /etc/yum.repos.d/localrepo.repo
[localrepo]
name=Local Repository
baseurl=ftp://<ip-of-ansible-host>/pub/localrepo
gpgcheck=0
enabled=1
```

5. Start the vsftpd service on the master node:

```
systemctl restart vsftpd
systemctl enable vsftpd

yum clean all
yum repolist
```

Install Python-Pip and Python36

Install python-pip and python36 to generate hosts.ini file for Kubespray (hosts.ini generation script works only with python3) packages:

```
# yum --disablerepo=* --enablerepo=localrepo install -y python-pip python36 python-setuptools python36 python36-libs
```

If yum does not pick the packages, then run following command to clear the cache:

```
yum --disablerepo=* --enablerepo=localrepo clean expire-cache
```

Install PIP Offline Packages

Navigate to the kubespray-offline directory and run the following command to install the packages:

```
pip install --no-index --find-links="/root/pip_packages" -r requirements.txt
```

Generate the hosts.ini File for Ansible

Follow these steps:

1. Navigate to the **kubespray-offline** directory.

```
cd kubespray-offline
```

2. Copy the sample inventory folder.

```
cp -r inventory/local/ inventory/mycluster
```

3. Declare the hosts and generate hosts.ini.

```
##For Single host:
declare -a IPS=(10.131.158.143)

##For multi host:
declare -a IPS=(10.238.32.23 10.238.32.58 10.238.34.106)

CONFIG_FILE=inventory/mycluster/hosts.ini /usr/bin/python3.6m contrib/inventory_builder/inventory.py ${IPS[@]}
```

> Note
>
> This step changes your node hostname to the one generated in the hosts.ini. To skip changing the hostname, edit the **roles/bootstrap-os/defaults/main.yml** file. In this file, change **override_system_hostname: true** to **override_system_hostname: false**.

```
[all]
node1 ansible_host=10.238.32.23 ip=10.238.32.23
node2 ansible_host=10.238.32.58 ip=10.238.32.58
node3 ansible_host=10.238.34.106 ip=10.238.34.106
[kube-master]
node1
[kube-node]
node1
node2
node3
[etcd]
node1
node2
node3
[k8s-cluster:children]
kube-node
kube-master
[calico-rr]
[vault]
node1
node2
node3
```

> Note
>
> Error: "Config file" creation error with python error code.
> **More Information**: Fix: Remove hosts.ini and try again running the **config file** command.

Provide Input Values to Kubespray

Edit the **inventory/mycluster/group_vars/k8s-cluster.yml** file to include the directory where the docker images get downloaded.

```
local_release_dir: "/root/docker-images/"      ## Folder where you keep copied images from internet connected host
offline_images: "/tmp/offline_images/"                ## Temporary folder on the nodes for kubespray to upload
docker images
kube_version: v1.10.11
docker_daemon_graph: "/var/lib/docker"                ## Make sure to set this option with your preferred storage
path
```

Set SELinux Contexts

Set the SELinux contexts for Kubernetes and the etcd folder.

Follow these steps:

1. Create a directory named kubernetes.

```
mkdir -p /etc/kubernetes/
```

2. Run the following command to set context for the Kubernetes files:

```
chcon -R -t svirt_sandbox_file_t /etc/kubernetes/
```

3. Create a directory named etcd.

```
mkdir -p /var/lib/etcd
```

4. Run the following command to set context for the etcd files:

```
chcon -R -t svirt_sandbox_file_t /var/lib/etcd
```

5. If SELinux is not working as expected, then disable SELinx:

```
setenforce 0
sed -i --follow-symlinks 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux
```

If SELinux is not disabled after editing, then restart all the hosts.

Configure the Firewalld

Configure the firewalld on the master node and on all the nodes.

```
firewall-cmd --permanent --add-port=6443/tcp
firewall-cmd --permanent --add-port=2379-2380/tcp
firewall-cmd --permanent --add-port=10250-10255/tcp
firewall-cmd --reload
firewall-cmd --list-ports
6443/tcp 2379-2380/tcp 10250/tcp 10251/tcp 10252/tcp 10255/tcp 30000-32767/tcp 6783/tcp 21/tcp
```

```
firewall-cmd --permanent --add-port=10250/tcp
firewall-cmd --permanent --add-port=10255/tcp
firewall-cmd --permanent --add-port=30000-32767/tcp
firewall-cmd --permanent --add-port=6783/tcp
firewall-cmd --reload
firewall-cmd --list-ports
10255/tcp 10250/tcp 30000-32767/tcp 6783/tcp
```

Run the following command to stop and disable firewalld if there are any issues:

```
systemctl stop firewalld && systemctl disable firewalld
```

Set Up SSH for Passwordless Access for Ansible Host

Manual Setup:

Use any of the hosts as the Ansible host. Then, copy the public key of that host from the **.ssh/id_rsa.pub** file and paste in to the **.ssh/authorized_keys** file of all other hosts.

Run the following command on all hosts to create SSH private and public keys:

```
ssh-keygen
```

Automated Setup:

Run the following command to copy the public key of the Ansible host to all hosts. Replace host2, host3, host4 with the selected IPs or hostnames.

```
for host in host2 host3 host4; do ssh-copy-id -i ~/.ssh/id_rsa.pub $host; done
```

Installation and Set Up

After you have downloaded and have installed all the dependencies, perform the following tasks:

Follow these steps:

1. Run the following command to create the yum repo file on all the nodes.

   ```
   ansible-playbook -v -i inventory/mycluster/hosts.ini cluster.yml --flush-cache --tags=yum-local
   ```

2. Run the following command to copy, upload, and download images from all the nodes. Ensure that the **variable local_release_dir** is set in the **inventory/mycluster/group_vars/k8s-cluster.yml** file.

   ```
   ansible-playbook -v -i inventory/mycluster/hosts.ini cluster.yml --flush-cache --tags=upload-images
   ```

   Check if all the images are loaded correctly on all the nodes. Any missing images make Kubespray to do a docker pull image, which will eventually fail and exit. Re-run the playbook to load the images correctly.

3. Run the following command to install Kubespray Kubernetes using Ansible Playbook:

   ```
   export ANSIBLE_INVALID_TASK_ATTRIBUTE_FAILED=False

   ansible-playbook -v -i inventory/mycluster/hosts.ini cluster.yml --flush-cache
   ```

> Note
>
> **More Information: Symptom:** Ansible failure for using multiple nameservers: **Solution:**
> vi inventory/mycluster/group_vars/all.yml
> uncomment line - docker_dns_servers_strict: false

4. Reset the clusters if there are any errors in the installation:

```
ansible-playbook -v -i  inventory/mycluster/hosts.ini reset.yml --flush-cache
```

5. Set up the dashboard.
   a. Log in into the master node and set up the service account for the dashboard:

```
kubectl create serviceaccount cluster-admin-dashboard-sa
kubectl get sa
kubectl create clusterrolebinding cluster-admin-dashboard-sa --clusterrole=cluster-admin --
serviceaccount=default:cluster-admin-dashboard-sa
TOKEN=$(kubectl describe secret $(kubectl -n kube-system get secret | awk '/^cluster-admin-dashboard-sa-
token-/{print $1}') | awk '$1=="token:"{print $2}') && echo $TOKEN
```

   b. Copy and save the token code.

```
kubectl cluster-info
kubernetes-dashboard is running at https://master1:6443/api/v1/namespaces/kube-
system/services/https:kubernetes-dashboard:/proxy
```

   c. Open the browser and enter the dashboard URL. Paste the Token code in the dashboard by selecting the Token option.

Comments

Please log in to post comments.