Multi Master

14 January 2021 15:33

Deploy Multi Master using kubespray

https://kubernetes.io/docs/setup/production-environment/tools/kubespray/
https://github.com/kubernetes-sigs/kubespray

Pre-Requisites

- 5 VMs
  - 1 workstation
    - Workstation/ha-proxy
  - 4 for Kubernetes cluster (I am using 3)
- Password less authentication
  - Privilege user (Ansible)
- DNS (hosts file)
- pip3

Building Blocks

- Ansible
  - kubeadm
- Cloud/On-prem

Implementation Steps:

1) **On All VMs**
   a. **Add entry into /etc/hosts file**

```
[root@instance-1 ~]# cat /etc/hosts
instance-1 10.128.0.9
instance-2 10.128.0.10
instance-3 10.128.0.11
instance-4 10.128.0.12
workstation2 1 10.128.0.13
```

   b. **Reset root password, to configure password less**
      # sudo su -
      # passwd root

   c. **Enable IPv4 forwarding**
      echo "net.ipv4.ip_forward = 1" >> /etc/sysctl.conf
      sysctl -p

   d. **Disable Firewall**
      systemctl disable firewalld
      systemctl stop firewalld

   e. **Allow root login**
      vi /etc/ssh/sshd_config
      i. Change value to **yes**
         # Change to no to disable tunnelled clear text passwords
         PasswordAuthentication yes

      ii. Change value to **yes**
         PermitRootLogin yes

      iii. Restart sshd service
         #systemctl restart sshd

2) **Enable passwordless authentication**
   a. **Workstation VM**
      i. Generate Public key
         # ssh-keygen -t rsa

      ii. Copy sshkey to all the VMs

3) **Clone kubespray repository** - https://github.com/kubernetes-sigs/kubespray
4) **Install dependencies from ``requirements.txt``**
   # apt-get update
   # apt install python3-pip
   Git clone -
   [root@instance-4 ~]# git clone https://github.com/kubernetes-sigs/kubespray.git

cd kubespray
sudo pip3 install -r requirements.txt

5) **# Copy ``inventory/sample`` as ``inventory/mycluster``**
   cp -rfp inventory/sample inventory/mycluster
6) **# Update Ansible inventory file with inventory builder**
   declare -a IPS=(10.10.1.3 10.10.1.4 10.10.1.5)
   CONFIG_FILE=inventory/mycluster/hosts.yaml python3 contrib/inventory_builder/inventory.py ${IPS[@]}
7) **# Review and change parameters under ``inventory/mycluster/group_vars``**
   cat inventory/mycluster/group_vars/all/all.yml
   cat inventory/mycluster/group_vars/k8s-cluster/k8s-cluster.yml
8) **# Deploy Kubespray with Ansible Playbook - run the playbook as root**
   ansible-playbook -i inventory/mycluster/hosts.yaml cluster.yml

```
PLAY RECAP *********************************************************************************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
node1                      : ok=500  changed=112  unreachable=0    failed=0    skipped=941  rescued=0    ignored=0
node2                      : ok=579  changed=125  unreachable=0    failed=0    skipped=1086 rescued=0    ignored=0
node3                      : ok=421  changed=92   unreachable=0    failed=0    skipped=626  rescued=0    ignored=0

Thursday 14 January 2021  18:00:46 +0000 (0:00:00.082)       0:11:04.739 ******
===============================================================================
container-engine/docker : ensure docker packages are installed ---------------------------------------------------------- 56.67s
kubernetes/kubeadm : Join to cluster ------------------------------------------------------------------------------------ 24.95s
kubernetes/master : Joining control plane node to the cluster. ---------------------------------------------------------- 23.05s
kubernetes/master : kubeadm | Initialize first master ------------------------------------------------------------------- 21.09s
Gen_certs | Write etcd master certs ------------------------------------------------------------------------------------- 13.81s
Gen_certs | Write etcd master certs ------------------------------------------------------------------------------------- 13.16s
kubernetes/preinstall : Install packages requirements ------------------------------------------------------------------- 11.28s
reload etcd ------------------------------------------------------------------------------------------------------------- 10.81s
kubernetes-apps/ansible : Kubernetes Apps | Start Resources ------------------------------------------------------------- 9.18s
bootstrap-os : Install libselinux python package ------------------------------------------------------------------------- 8.90s
download_container | Download image if required ------------------------------------------------------------------------- 7.78s
kubernetes-apps/ansible : Kubernetes Apps | Lay Down CoreDNS Template --------------------------------------------------- 7.64s
download_container | Download image if required ------------------------------------------------------------------------- 7.15s
download_container | Download image if required ------------------------------------------------------------------------- 6.83s
Gen_certs | Gather etcd master certs ------------------------------------------------------------------------------------- 6.38s
Gen_certs | Gather etcd master certs ------------------------------------------------------------------------------------- 6.09s
download_container | Download image if required ------------------------------------------------------------------------- 6.07s
download | Download files / images ------------------------------------------------------------------------------------- 6.03s
wait for etcd up -------------------------------------------------------------------------------------------------------- 5.93s
container-engine/docker : Write docker dns systemd drop-in -------------------------------------------------------------- 5.81s
root@workstation2:~/kubespray#
root@workstation2:~/kubespray#
root@workstation2:~/kubespray#
```

9) On Workstation

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
deb https://apt.kubernetes.io/ kubernetes-xenial main
EOF
sudo apt-get update

apt-get install -y kubectl

scp root@10.128.0.9:/etc/kubernetes/admin.conf ~/.kube/config

Autoscaling:

- https://github.com/kubernetes-sigs/kubespray/blob/master/docs/getting-started.md
- Add entries into hosts.yaml file under required sections (kube-master, kube-node)
  inventory/mycluster/hosts.yml

ansible-playbook -i inventory/mycluster/hosts.yml scale.yaml

- **To Remove - run playbook first and then remove entry from inventory file**
  - # ansible-playbook -i inventory/mycluster/hosts.yml remove-node.yaml --extra-vars "node=nodename1,nodename2"

Reset your cluster:

ansible-playbook -i inventory/mycluster/hosts.yml reset.yaml

```
Friday 15 January 2021  06:10:04 +0000 (0:00:01.355)       0:00:43.512 ********
===============================================================================
Gather necessary facts ---------------------------------------------------------- 10.35s
reset | delete some files and directories ------------------------------------------ 9.93s
Gather minimal facts ------------------------------------------------------------ 3.08s
download | Download files / images --------------------------------------------- 2.33s
reset | remove all containers ----------------------------------------------------- 2.06s
reset | remove services ----------------------------------------------------------- 1.41s
reset | Restart network ----------------------------------------------------------- 1.36s
reset | stop services ------------------------------------------------------------- 1.22s
reset | unmount kubelet dirs ------------------------------------------------------ 0.96s
reset : flush iptables ---------------------------------------------------------- 0.85s
reset | remove docker dropins ----------------------------------------------------- 0.78s
reset | restart docker if needed -------------------------------------------------- 0.75s
reset | remove dns settings from dhclient.conf ------------------------------------ 0.63s
reset | stop etcd services -------------------------------------------------------- 0.60s
reset | remove remaining routes set by bird --------------------------------------- 0.60s
reset | check if crictl is present ------------------------------------------------ 0.45s
reset | systemctl daemon-reload --------------------------------------------------- 0.43s
reset | remove etcd services ------------------------------------------------------ 0.43s
reset | remove the network device created by calico ------------------------------- 0.33s
reset | check kube-ipvs0 network device ------------------------------------------- 0.28s
root@workstation2:~/kubespray#
```

For External HA
https://github.com/kubernetes-sigs/kubespray/blob/master/docs/ha-mode.md

1) **On Workstation (HAPROXY)**

   a. **Install ha-proxy and setup loadbalancer**
   # apt-get update && apt-get install -y haproxy

   # vi /etc/haproxy/haproxy.cfg

```
listen kubernetes-apiserver-https
  bind 10.128.0.13:8383
  mode tcp
  option log-health-checks
  timeout client 3h
  timeout server 3h
  server master1 10.128.0.9:6443 check check-ssl verify none inter 10000
  server master2 10.128.0.10:6443 check check-ssl verify none inter 10000
  balance roundrobin
~
~
~
~
```

   # systemctl restart haproxy

   # vi inventory/mycluster/group_vars/all/all.yml

```
## External LB example config
apiserver_loadbalancer_domain_name: "lb.example.com"
loadbalancer_apiserver:
  address: 10.128.0.13
  port: 8383

## Internal loadbalancers for apiservers
loadbalancer_apiserver_localhost: false
# valid options are "nginx" or "haproxy"
# loadbalancer_apiserver_type: nginx  # valid values "nginx" or "haproxy"
```

2) **Updated /etc/hosts file if you don't have DNS**

```
root@workstation2:~/kubespray# cat /etc/hosts
instance-1 10.128.0.9
instance-2 10.128.0.10
instance-3 10.128.0.11
instance-4 10.128.0.12
workstation2 10.128.0.13
lb.example.com 10.128.0.13
127.0.0.1 localhost
```

3) **Trigger the setup**

   # ansible-playbook -i inventory/mycluster/hosts.yaml cluster.yaml

```
Friday 15 January 2021  06:21:16 +0000 (0:00:00.093)       0:07:33.808 ********
===============================================================================
kubernetes/kubeadm : Join to cluster ------------------------------------------
----------------------------------------- 30.56s
kubernetes/master : kubeadm | Initialize first master ------------------------
----------------------------------------- 28.49s
reload etcd -------------------------------------------------------------------
----------------------------------------- 10.74s
Gen_certs | Write etcd master certs -----------------------------------------
----------------------------------------- 10.56s
Gen_certs | Write etcd master certs -----------------------------------------
----------------------------------------- 10.46s
kubernetes/master : Joining control plane node to the cluster. ---------------
----------------------------------------- 9.28s
kubernetes-apps/ansible : Kubernetes Apps | Start Resources ------------------
----------------------------------------- 7.36s
kubernetes/preinstall : Install packages requirements ------------------------
----------------------------------------- 6.85s
wait for etcd up --------------------------------------------------------------
----------------------------------------- 6.49s
network_plugin/calico : Start Calico resources --------------------------------
----------------------------------------- 6.41s
kubernetes-apps/ansible : Kubernetes Apps | Lay Down CoreDNS Template ---------
----------------------------------------- 5.36s
Configure | Check if etcd cluster is healthy ----------------------------------
----------------------------------------- 5.27s
Gen_certs | Gather etcd master certs -----------------------------------------
----------------------------------------- 4.76s
Gen_certs | Gather etcd master certs -----------------------------------------
----------------------------------------- 4.70s
download | Download files / images ------------------------------------------
----------------------------------------- 4.60s
kubernetes/preinstall : Get current calico cluster version -------------------
----------------------------------------- 4.06s
kubernetes/master : Master | wait for kube-scheduler --------------------------
----------------------------------------- 3.58s
network_plugin/calico : Calico | Create calico manifests ----------------------
----------------------------------------- 2.96s
download_file | Download item ----------------------------------------------------
----------------------------------------- 2.94s
container-engine/docker : ensure docker packages are installed ---------------
----------------------------------------- 2.74s
root@workstation2:~/kubespray#
root@workstation2:~/kubespray# ansible-playbook -i inventory/mycluster/hosts.yaml cluster.yml
```

4) **Access your Kubernetes cluster**
   a. **On workstation VM**

```
root@workstation2:~/kubespray# scp root@10.128.0.9:/etc/kubernetes/admin.conf ~/.kube/config
admin.conf                                                                                    100% 5582     5.5KB/s   00:00
root@workstation2:~/kubespray#
root@workstation2:~/kubespray# cat ~/.kube/config
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUM1ekNDQWMrZ0F3SUJBZ01CQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdwcmRXSm
wKY201bGRHVnpNQjRYRFRJeE1ERXhOVEEyTVRnd09Wb1hEVE14TURFeE16QTJNVGd3T1Zwd0ZURVRQkVHQTFVRQpBeE1LYTNWaVpYSnVaWFJsY3pDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBB
RENDQVFvQ2dnRUJBT242ClRDTEEwOTBObUdMYXRtdlRBdVA2RjN2SmIrVEFac1JPSFN1bkJPRDNGTTRHNEM1TEVEUzVZRm9uOS9XcXN5UzEKVGlpTU40UVJac1FiT3N0bDlRbXUxbFg2RFhCWFBzdU1uMS
t4ZGY4SzFwa29ERDF2NUd2cXV3S31HRGkwR01XSApoNStyL3dlQ1FVRzhnV2FUdWVQWVNhTlNWS3RNZ2Mzc1lSaXR6MjI4K1NzeVdmNmxQQzRZRjhCcERVT0pVWktBCkNxNFVhSzhPd0V3YnFpWmtaMDR0
cUt0eisvS01kWE96Q0U0QlNJam16d3lKQzFPR3B6TmxsMWlHaUJWUHc5dWIKbTZmb2xLR1dxWXRFZDJKTkwzT1lzNlVscjlUTFQzaFNpZkhpdDFYS2I4WHZPMFpLM3AzM3J6UGVETW02ZU9rQgpxWjRmc0
U2MUtVS0VsR2JLTnhrQ0F3RUFBYU5DTUVBd0RnWURWUjBQQVFIL0JBUURBZ0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0hRWURWUjBPQkJZRUZIS3RBYjNQYXNwd01qWUFZbEg5bmxud0JjcmVNQTBH
Q1NxR1NJYjMKRFFFQkN3VUFBNElCQVFBcVhMcjRZamFYZ3lmbnY0WTI0dXF0Y3VtblZyYW9PWDNZNzBwTVg2MlF3a2tmM1FabAo5Mm1SR28zWG9TZS9oUmhXMU1HZW5pNHByNVpiQT13dmRwK1NTZ1AwMl
J6WkhlcitEQkl3SVY4MnNSTFdvUVVGCkowZVdzSEhESGRpTWErL1pSZDVCZ3dGR3FrWDhnck5LcytWUWp1NFFxL2U4VVFFSDF3dGRkdndmOWNKeXdETkwKVys1cy9DR1d2ZXZtbmJQb3hDZW1rSm95WWE0
UGloelZOYlNFV1RDS2UxYVdiemFPRGEzdXh0blh4RGZPQjRZcAp4bUNtSnVzZ0hnY0xHck41QjF1NWhuc1VadnZwS0REWXJxeC9kNTZyNlhYZ3VSU0tqNkdqOTV3c3JFTWF0NlFUCjRyS0Q1NjRKUE9scU
5ySFVhYmtoMzRhYzNNVzM3V09sWUJyQgotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
    server: https://lb.example.com:8383
  name: cluster.local
contexts:
- context:
    cluster: cluster.local
    user: kubernetes-admin
  name: kubernetes-admin@cluster.local
current-context: kubernetes-admin@cluster.local
kind: Config
```

   a. **From Master node**

```
[root@node1 ~]# kubectl  get pods -A
NAMESPACE     NAME                                       READY   STATUS    RESTARTS   AGE
kube-system   calico-kube-controllers-8b5ff5d58-m8j86    1/1     Running   0          25m
kube-system   calico-node-5tszh                          1/1     Running   0          26m
kube-system   calico-node-d5zcj                          1/1     Running   0          26m
kube-system   calico-node-pqhf8                          1/1     Running   0          26m
kube-system   coredns-85967d65-4qwq5                     1/1     Running   0          25m
kube-system   coredns-85967d65-84491                     1/1     Running   0          25m
kube-system   dns-autoscaler-5b7b5c9b6f-86p2r            1/1     Running   0          25m
kube-system   kube-apiserver-node1                       1/1     Running   0          27m
kube-system   kube-apiserver-node2                       1/1     Running   0          27m
kube-system   kube-controller-manager-node1              1/1     Running   0          27m
kube-system   kube-controller-manager-node2              1/1     Running   0          27m
kube-system   kube-proxy-6htrz                           1/1     Running   0          27m
kube-system   kube-proxy-8jppb                           1/1     Running   0          26m
kube-system   kube-proxy-mlxtf                           1/1     Running   0          27m
kube-system   kube-scheduler-node1                       1/1     Running   0          27m
kube-system   kube-scheduler-node2                       1/1     Running   0          27m
kube-system   nodelocaldns-bp4n4                         1/1     Running   0          25m
kube-system   nodelocaldns-clf7c                         1/1     Running   0          25m
kube-system   nodelocaldns-djr7b                         1/1     Running   0          25m
[root@node1 ~]# cat .kube/config | head - 15
==> standard input <==
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUM1ekNDQWMrZ0F3SUJBZ01CQ
wKY201bGRHVnpNQjRYRFRJeE1ERXhOVEEyTVRnd09Wb1hEVE14TURFeE16QTJNVGd3T1Zvd0ZURVRNQkVHQTFVRQpBeE1LYTN
RENDQVFvQ2dnRUJBT242ClRDTEEvOTBObUdMYXRtdlRBdVA2RjN2SmIrVEFaclJPSFN1bkJPRDNGTTRHNEM1TEVEUzVZRm9uC
c4ZGY4SzFwa29ERDF2NUd2cXV3S3lHRGkwR01XSApoNStyL3dlQ1FVRzhnV2FUdWVQWVNhTlNWS3RNZ2Mzc1lSaXR6MjI4K1N
cUt0eisvS01kWE96Q0U0Q1NJam16d3lKQzFPR3B6TmxsMWlHaUJWUHc5dWIKbTZmb2xLR1dxWXRFZDJKTkwzT1lzNlVscjlUT
U2MUtVS0VsR2JLTnhrQ0F3RUFBYU5DTUVBd0RnWUdWUjBQQVFIL0JBUURBZ0trTUE4R0ExVWRFd0VCCi93UUZNQU1CQWY4d0h
Q1NxR1NJYjMKRFFFQkN3VUFBNElCQVFBcVhMcjRZamFYZ3lmbnY0WTI0dXF0Y3VtblZyYW9PWDNZNzBwTVg2M1F3a2tmM1Fab
J6WkhlcitEQkl3SVY4MnNSTFdvUVVGCkowZVdzSEhESGRpTWErL1pSZDVCZ3dGR3FrWDhnck5LcytWUWp1NFFxL2U4VVFFSDB
JGloelZOY1NFV1RDS2UxYVdiemFPRGEzdXh0b1h4RGZPQjRZcAp4bUNtSnVzZ0hnY0xHck41QjF1NWhuc1VadnZwS0REWXJxe
5ySFVhYmtoMzRhYzNNVzM3V09sWUJyQgotLS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==
    server: https://lb.example.com:8383
  name: cluster.local
contexts:
- context:
    cluster: cluster.local
    user: kubernetes-admin
head: cannot open ‘15’ for reading: No such file or directory
[root@node1 ~]#
```

5) **How Controller and Scheduler working - LEADER**

```
[root@node1 ~]# kubectl get endpoints -n kube-system
NAME                      ENDPOINTS                                                     AGE
coredns                   10.233.90.1:53,10.233.96.2:53,10.233.90.1:53 + 3 more...   27m
kube-controller-manager   <none>                                                        29m
kube-scheduler            <none>                                                        29m
[root@node1 ~]#
```

```
[root@node1 ~]# kubectl  get endpoints kube-scheduler -n kube-system -o yaml
apiVersion: v1
kind: Endpoints
metadata:
  annotations:
    control-plane.alpha.kubernetes.io/leader: '{"holderIdentity":"node2_64676e9b-f0e5-48f0-bf62-a8cc9e26e467","leaseDurationSeconds":15,"acquireTime":"202
1-01-15T06:19:27Z","renewTime":"2021-01-15T06:49:27Z","leaderTransitions":1}'
  creationTimestamp: "2021-01-15T06:18:25Z"
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
```

```
[root@node1 ~]# kubectl  get endpoints kube-controller-manager -n kube-system -o yaml
apiVersion: v1
kind: Endpoints
metadata:
  annotations:
    control-plane.alpha.kubernetes.io/leader: '{"holderIdentity":"node2_3d8b20ab-5577-445d-b56f-a80721bd4cb7","leaseDurationSeconds":15,"acquireTime":"202
1-01-15T06:19:26Z","renewTime":"2021-01-15T06:51:03Z","leaderTransitions":1}'
  creationTimestamp: "2021-01-15T06:18:24Z"
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
```

6) **API Server:**

```
[root@node1 ~]# kubectl get nodes
NAME    STATUS   ROLES    AGE   VERSION
node1   Ready    master   33m   v1.19.7
node2   Ready    master   33m   v1.19.7
node3   Ready    <none>   32m   v1.19.7
[root@node1 ~]#
```

```
[root@node1 ~]# kubectl  describe -n kube-system pod kube-apiserver-node1 | more
Name:                 kube-apiserver-node1
Namespace:            kube-system
Priority:             2000001000
Priority Class Name:  system-node-critical
Node:                 node1/10.128.0.9
```

- **Communication with "etcd"**

```
[root@node1 ~]# kubectl  describe -n kube-system pod kube-apiserver-node1 | grep etcd-servers
      --etcd-servers=https://10.128.0.9:2379,https://10.128.0.10:2379,https://10.128.0.11:2379
[root@node1 ~]#
```

```
[root@node1 ~]# kubectl get pods -A -o wide | grep node3
kube-system   calico-kube-controllers-8b5ff5d58-m8j86   1/1     Running   0          33m   10.128.0.11   node3   <none>           <none>
kube-system   calico-node-5tszh                         1/1     Running   0          33m   10.128.0.11   node3   <none>           <none>
kube-system   kube-proxy-8jppb                          1/1     Running   0          34m   10.128.0.11   node3   <none>           <none>
kube-system   nodelocaldns-djr7b                        1/1     Running   0          32m   10.128.0.11   node3   <none>           <none>
[root@node1 ~]#
```

- **General opertaions**

```
[root@node1 ~]# kubectl create deployment my-web --image nginx --replicas=5
deployment.apps/my-web created
```

```
[root@node1 ~]# kubectl  get svc
NAME         TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.233.0.1      <none>        443/TCP   54m
svcclus      ClusterIP   10.233.36.138   <none>        80/TCP    35s
[root@node1 ~]# kubectl  describe svc svcclus
Name:              svcclus
Namespace:         default
Labels:            app=my-web
Annotations:       <none>
Selector:          app=my-web
Type:              ClusterIP
IP:                10.233.36.138
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         10.233.90.2:80,10.233.90.3:80,10.233.92.1:80 + 2 more...
Session Affinity:  None
Events:            <none>
```

```
[root@node1 ~]# kubectl  get svc
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes   ClusterIP      10.233.0.1      <none>        443/TCP        57m
my-web       ClusterIP      10.233.2.126    <none>        80/TCP         85s
svcclus1b    LoadBalancer   10.233.32.163   <pending>     80:30600/TCP   5s
[root@node1 ~]#
```

7) Conclusion

- **etcd instance**: all instances will be clustered together using consensus;
- **API server:** each server will talk to local etcd - all API servers in the cluster will be available;
- controllers, scheduler, and cluster auto-scaler: will use lease mechanism - only one instance of each of them will be active in the cluster;
- **add-on manager:** each manager will work independently trying to keep add-ons in sync.