# microk8s 安装

microk8s 是 Canonical 的k8s 发行版，默认使用snap 进行安装，CentOS需要先安装 snap。

## CentOS安装 snap

snap安装需要在CentOS 7.6 + 才支持

1. 更换国内yum源

```
mv /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo.bak
wget -O /etc/yum.repos.d/CentOS-Base.repo
https://mirrors.aliyun.com/repo/Centos-7.repo
sed -i -e '/mirrors.cloud.aliyuncs.com/d' -e '/mirrors.aliyuncs.com/d'
/etc/yum.repos.d/CentOS-Base.repo
yum makecache
```

2. 关闭防火墙

```
systemctl stop firewalld.service
systemctl disable firewalld.service
```

3. 关闭selinux

```
setenforce 0
[root@node1 ~]# cat /etc/selinux/config
SELINUX=disabled
```

4. 安装 snap

```
$ yum install -y epel-release
$ yum install -y snapd
$ systemctl enable --now snapd.socket
$ systemctl enable --now snapd.service
$ snap install core
core 16-2.52.1 from Canonical✓ installed
$ ln -vfs /var/lib/snapd/snap /snap
```

5. 配置snap加速

```
# 添加hosts解析
91.189.91.43    darkbowser.canonical.com

snap install snap-store-proxy
snap install snap-store-proxy-client

# systemctl edit snapd.service
[Service]
Environment="HTTP_PROXY=http://10.11.12.123:10240"
Environment="HTTPS_PROXY=http://10.11.12.123:10240"
Environment="NO_PROXY=localhost,127.0.0.1,192.168.0.0/24,*.domain.ltd"
```

## 安装microk8s

```
snap install microk8s --classic

[root@node1 ~]# cat /etc/hosts
192.168.251.244 node1
192.168.251.98 node2
192.168.251.249 node3
```

## 下载被墙镜像

```
curl -L "https://raw.githubusercontent.com/OpsDocker/pullk8s/main/pullk8s.sh" -o
/usr/local/bin/pullk8s

https://www.ucloud.cn/yun/33182.html

docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/pause:3.1
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/pause:3.1
k8s.gcr.io/pause:3.1
docker save k8s.gcr.io/pause:3.1 -o pause.3.1.tar
microk8s.ctr image import pause.3.1.tar

docker pull registry.cn-hangzhou.aliyuncs.com/google_containers/metrics-
server:v0.5.2
docker tag registry.cn-hangzhou.aliyuncs.com/google_containers/metrics-
server:v0.5.2 k8s.gcr.io/metrics-server/metrics-server:v0.5.2
docker save k8s.gcr.io/metrics-server/metrics-server:v0.5.2 -o metrics-
server.0.5.2.tar
microk8s.ctr image import metrics-server.0.5.2.tar
```

## 启用RBAC

```
microk8s enable rbac
```

## 连接常用路径

```
$ ln -vsf /var/snap/microk8s/common/var/lib/kubelet /var/lib/
‘/var/lib/kubelet’ -> ‘/var/snap/microk8s/common/var/lib/kubelet’

$ ln -vsf /var/snap/microk8s/common/var/lib/containerd /var/lib/
‘/var/lib/containerd’ -> ‘/var/snap/microk8s/common/var/lib/containerd’
```

## 配置CLI

```
$ snap alias microk8s.kubectl kubectl
$ kubectl completion bash >/etc/bash_completion.d/kubectl
```

## 安装helm3

```
microk8s enable helm3

Fetching helm version v3.8.0.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0 12.9M    0 81920    0     0    876      0  4:19:15  0:01:33  4:17:42     0
curl: (56) OpenSSL SSL_read: SSL_ERROR_SYSCALL, errno 104
## 网络问题出现以上报错

# vim /var/snap/microk8s/common/addons/core/addons/helm3/enable
#!/usr/bin/env bash

set -e

source $SNAP/actions/common/utils.sh

echo "Enabling Helm 3"

if [ ! -f "${SNAP_DATA}/bin/helm3" ]
then
  #SOURCE_URI="https://get.helm.sh"
  SOURCE_URI="https://mirrors.huaweicloud.com/helm/v3.8.0"   ## 改华为的镜像地址
  HELM_VERSION="v3.8.0"
```

## 配置dashboard

```
[root@node1 yaml]# cat admin.yml
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: admin
roleRef:
  kind: ClusterRole
  name: cluster-admin
  apiGroup: rbac.authorization.k8s.io
subjects:
- kind: ServiceAccount
  name: admin
  namespace: kube-system
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin
  namespace: kube-system

[root@node1 yaml]# kubectl apply -f admin.yml    ## 创建sa  admin
```

```
[root@node1 yaml]# kubectl get svc -A
NAMESPACE     NAME                        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                  AGE
default       kubernetes                  ClusterIP   10.152.183.1     <none>        443/TCP                  41h
kube-system   kube-dns                    ClusterIP   10.152.183.10    <none>        53/UDP,53/TCP,9153/TCP   23h
kube-system   metrics-server              ClusterIP   10.152.183.121   <none>        443/TCP                  20h
kube-system   kubernetes-dashboard        ClusterIP   10.152.183.110   <none>        443/TCP                  20h
kube-system   dashboard-metrics-scraper   ClusterIP   10.152.183.225   <none>        8000/TCP                 20h
default       nginx                       NodePort    10.152.183.26    <none>        80:32530/TCP             17h
edoc2         edoc2                       ClusterIP   10.152.183.254   <none>        80/TCP,5002/TCP          16h
[root@node1 yaml]# kubectl edit svc kubernetes-dashboard -n kube-system
service/kubernetes-dashboard edited
[root@node1 yaml]# kubectl get svc -A
NAMESPACE     NAME                        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                  AGE
default       kubernetes                  ClusterIP   10.152.183.1     <none>        443/TCP                  41h
kube-system   kube-dns                    ClusterIP   10.152.183.10    <none>        53/UDP,53/TCP,9153/TCP   23h
kube-system   metrics-server              ClusterIP   10.152.183.121   <none>        443/TCP                  20h
kube-system   dashboard-metrics-scraper   ClusterIP   10.152.183.225   <none>        8000/TCP                 20h
default       nginx                       NodePort    10.152.183.26    <none>        80:32530/TCP             17h
edoc2         edoc2                       ClusterIP   10.152.183.254   <none>        80/TCP,5002/TCP          16h
kube-system   kubernetes-dashboard        NodePort    10.152.183.110   <none>        443:30284/TCP            20h
```

```
service type 改为 NodePort
```

```
[root@node1 yaml]# kubectl -n kube-system create token admin
# 对 admin 创建token
```

# microk8s 问题

1. microk8s是Canonical 的k8s发行版，默认使用snap 进行安装，CentOS需要先安装 snap，CentOS7.6+才支持snap的安装，snap的安装依赖众多。
2. microk8s主要的定位是在本地开发、物联网和边缘计算上进行使用，
3. 使用snap安装microk8s时由于国内没有snap仓库，导致在线安装十分缓慢，使用离线方式安装需要安装两个snap包（microk8s.snap 和 core.snap 两个包总计 330M，还不包括依赖的k8s运行组件镜像）
4. microk8s整个集群的状态数据存储保存在dqlite中，而不是标准k8s的etcd中；服务以snap进行管理，使用自有的kubelite 对k8s 的标准服务进行类封装，增加了后期维护成本和难度。

```
[root@node1 microk8s]# ps -ef | grep microk8s
root      2845     1  0 11:07 ?        00:00:15 /bin/bash /var/lib/snapd/snap/microk8s/3272/apiservice-kicker
root      2854     1  0 11:07 ?        00:00:00 /bin/bash /var/lib/snapd/snap/microk8s/3272/run-cluster-agent-with-args
root      2866     1  1 11:07 ?        00:02:23 /snap/microk8s/3272/bin/containerd --config /var/snap/microk8s/3272/args/containerd.toml --root /var/snap/microk8s/common/var/lib/containerd
--state /var/snap/microk8s/common/run/containerd --address /var/snap/microk8s/common/run/containerd.sock
root      2909  2854  0 11:07 ?        00:00:00 /snap/microk8s/3272/bin/cluster-agent --bind 0.0.0.0:25000 --keyfile /var/snap/microk8s/3272/certs/server.key --certfile /var/snap/microk8s/3
272/certs/server.crt --timeout 240
root      3369     1  4 11:07 ?        00:07:40 /snap/microk8s/3272/bin/k8s-dqlite --storage-dir=/var/snap/microk8s/3272/var/kubernetes/backend/ --listen=unix:///var/snap/microk8s/3272/var/
kubernetes/backend/kine.sock:12379
root      3400     1  9 11:07 ?        00:17:40 /snap/microk8s/3272/kubelite --scheduler-args-file=/var/snap/microk8s/3272/args/kube-scheduler --controller-manager-args-file=/var/snap/micro
k8s/3272/args/kube-controller-manager --proxy-args-file=/var/snap/microk8s/3272/args/kube-proxy --kubelet-args-file=/var/snap/microk8s/3272/args/kubelet --apiserver-args-file=/var/snap/micr
ok8s/3272/args/kube-apiserver --kubeconfig-file=/var/snap/microk8s/3272/credentials/client.config --start-control-plane=true
root      4212     1  0 11:07 ?        00:00:00 /var/lib/snapd/snap/microk8s/3272/bin/containerd-shim-runc-v1 -namespace k8s.io -id 89aa411410c47d51f2f8f1853a6a65583cca44955bcac367a17cb8598
568d5b1 -address /var/snap/microk8s/common/run/containerd.sock
root      4352     1  0 11:07 ?        00:00:00 /var/lib/snapd/snap/microk8s/3272/bin/containerd-shim-runc-v1 -namespace k8s.io -id a17e9679a8e4a00da599dea8fc0fbad007275462ce80913da3c18a859
67765a5 -address /var/snap/microk8s/common/run/containerd.sock
root      4370     1  0 11:07 ?        00:00:04 /var/lib/snapd/snap/microk8s/3272/bin/containerd-shim-runc-v1 -namespace k8s.io -id 408a98ecf532ed77ed2f20a03994cb18e0f7b37ca67d4d6d6b5d453d8
cc23f7e -address /var/snap/microk8s/common/run/containerd.sock
```

    - snap熟悉学习成本
    - microk8s 自有的封装组件的原理熟悉及后续问题排错成本
    - 官方文档及社区不够完善
5. 使用microk8s v1.24/stable版本验证搭建集群，添加节点存在问题 cluster-agent这个25000端口不通，无法添加集群，文档没有找到解决方案

```
[root@node2 ~]# microk8s join 192.168.251.244:25000/88a04d75d5c98ab5889ee6a3c9d4469d/bfa0841ea0e3
Contacting cluster at 192.168.251.244
Traceback (most recent call last):
  File "/snap/microk8s/3272/scripts/cluster/join.py", line 968, in <module>
    join(prog_name="microk8s join")
  File "/snap/microk8s/3272/usr/lib/python3/dist-packages/click/core.py", line 722, in __call__
    return self.main(*args, **kwargs)
  File "/snap/microk8s/3272/usr/lib/python3/dist-packages/click/core.py", line 697, in main
    rv = self.invoke(ctx)
  File "/snap/microk8s/3272/usr/lib/python3/dist-packages/click/core.py", line 895, in invoke
    return ctx.invoke(self.callback, **ctx.params)
  File "/snap/microk8s/3272/usr/lib/python3/dist-packages/click/core.py", line 535, in invoke
    return callback(*args, **kwargs)
  File "/snap/microk8s/3272/scripts/cluster/join.py", line 961, in join
    join_dqlite(connection_parts, verify, worker)
  File "/snap/microk8s/3272/scripts/cluster/join.py", line 739, in join_dqlite
    worker=worker,
  File "/snap/microk8s/3272/scripts/cluster/join.py", line 152, in get_connection_info
    conn.connect()
  File "/snap/microk8s/3272/usr/lib/python3.6/http/client.py", line 1438, in connect
    super().connect()
  File "/snap/microk8s/3272/usr/lib/python3.6/http/client.py", line 956, in connect
    (self.host,self.port), self.timeout, self.source_address)
  File "/snap/microk8s/3272/usr/lib/python3.6/socket.py", line 724, in create_connection
    raise err
  File "/snap/microk8s/3272/usr/lib/python3.6/socket.py", line 713, in create_connection
    sock.connect(sa)
OSError: [Errno 113] No route to host
```

```
[root@node2 ~]# telnet 192.168.251.244 25000
Trying 192.168.251.244...
telnet: connect to address 192.168.251.244: No route to host
```

```
[root@node1 ~]# ss -lnpt | grep 25000
LISTEN     0      128          :::25000                   :::*                   users:(("cluster-agent",pid=2909,fd=6))
[root@node1 ~]# ps -ef |grep cluster-agent
root      2504 28842  0 15:23 pts/2    00:00:00 grep --color=auto cluster-agent
root      2854     1  0 11:07 ?        00:00:00 /bin/bash /var/lib/snapd/snap/microk8s/3272/run-cluster-agent-with-args
root      2909  2854  0 11:07 ?        00:00:00 /snap/microk8s/3272/bin/cluster-agent --bind 0.0.0.0:25000 --keyfile /var/snap/microk8s/3272/certs/server.key --certfile /var/snap/microk8s/3
272/certs/server.crt --timeout 240
```