节点IP地址：192.168.30.141

节点主机名：k8s-master-01

证书存放目录：/etc/kubernetes/kube-apiserver/ssl/

配置文件目录：/etc/kubernetes/kube-apiserver/conf/

日志存放目录：/var/log/kubernetes/kube-apiserver/

创建所需目录

mkdir -p /etc/kubernetes/kube-apiserver/{ssl,conf} \

/var/log/kubernetes/kube-apiserver/

创建证书签名请求

cd /usr/local/src/ssl/

cat > apiserver-csr.json <<EOF

{

"CN": "kubernetes",

"hosts": ["127.0.0.1",

"10.96.0.1",

"192.168.30.139",

"192.168.30.140",

"192.168.30.141",

"192.168.30.142",

"192.168.30.143",

"kubernetes",

"kubernetes.default",

"kubernetes.default.svc",

"kubernetes.default.svc.cluster",

"kubernetes.default.svc.cluster.local"],

"key": {

"algo": "rsa",

"size": 2048

},

"names": [{

"C": "CN",

"ST": "BeiJing",

"L": "BeiJing",

"OU": "System"

}]

}

EOF

注意：使用10.96.0.0/12做为集群Service网段，必须将10.96.0.1此IP地址加在hosts字段中

生成证书和私钥

cd /usr/local/src/ssl/

cfssl gencert -ca=/etc/kubernetes/ssl/ca.pem \

-ca-key=/etc/kubernetes/ssl/ca-key.pem \

-config=/etc/kubernetes/ssl/ca-config.json \

-profile=kubernetes apiserver-csr.json | cfssljson -bare apiserver

cp apiserver.pem apiserver-key.pem /etc/kubernetes/kube-apiserver/ssl/

下载二进制文件

cd /usr/local/src/

wget https://dl.k8s.io/v1.18.3/kubernetes-server-linux-amd64.tar.gz

tar xf kubernetes-server-linux-amd64.tar.gz

cp kubernetes/server/bin/{kube-apiserver,kube-scheduler,kube-controller-manager,kubectl} /usr/bin/

创建配置文件

cat > /etc/kubernetes/kube-apiserver/conf/kube-apiserver.conf <<EOF

KUBE\_APISERVER\_OPTS="--bind-address=192.168.30.141 \

--enable-admission-plugins=NamespaceLifecycle,LimitRanger,ServiceAccount,DefaultStorageClass,ResourceQuota,NodeRestriction \

--secure-port=6443 --authorization-mode=Node,RBAC \

--runtime-config=rbac.authorization.k8s.io/v1 \

--anonymous-auth=false \

--enable-bootstrap-token-auth=true \

--token-auth-file=/etc/kubernetes/kube-apiserver/conf/bootstrap-token.csv \

--service-cluster-ip-range=10.96.0.0/12 \

--service-node-port-range=30000-36000 \

--kubelet-client-certificate=/etc/kubernetes/kube-apiserver/ssl/apiserver.pem \

--kubelet-client-key=/etc/kubernetes/kube-apiserver/ssl/apiserver-key.pem \

--tls-cert-file=/etc/kubernetes/kube-apiserver/ssl/apiserver.pem \

--tls-private-key-file=/etc/kubernetes/kube-apiserver/ssl/apiserver-key.pem \

--client-ca-file=/etc/kubernetes/ssl/ca.pem \

--service-account-key-file=/etc/kubernetes/ssl/ca-key.pem \

--etcd-cafile=/etc/kubernetes/ssl/ca.pem \

--etcd-certfile=/etc/kubernetes/etcd/ssl/etcd.pem \

--etcd-keyfile=/etc/kubernetes/etcd/ssl/etcd-key.pem \

--etcd-servers=https://192.168.30.141:2379,https://192.168.30.142:2379, https://192.168.30.143:2379 \

--allow-privileged=true \

--audit-log-maxage=30 \

--audit-log-maxbackup=3 \

--audit-log-maxsize=100 \

--audit-log-path=/var/log/kubernetes/kube-apiserver/api-audit.log \

--v=2 \

--logtostderr=false \

--log-dir=/var/log/kubernetes/kube-apiserver/"

EOF

启用TLS Bootstrapping机制，文件内容格式：token，用户名，UID，用户组

TOKEN=$(head -c 16 /dev/urandom | od -An -t x | tr -d ' ')

cat > /etc/kubernetes/kube-apiserver/conf/bootstrap-token.csv <<EOF

$TOKEN,kubelet-bootstrap,10001,"system:node-bootstrap"

EOF

echo $TOKEN && awk -F',' '{print $1}' /etc/kubernetes/kube-apiserver/conf/bootstrap-token.csv

创建服务文件

cat > /usr/lib/systemd/system/kube-apiserver.service <<EOF

[Unit]

Description=Kubernetes API Server

Documentation=https://github.com/GoogleCloudPlatform/kubernetes

After=network.target

[Service]

EnvironmentFile=-/etc/kubernetes/kube-apiserver/conf/kube-apiserver.conf

ExecStart=/usr/bin/kube-apiserver \$KUBE\_APISERVER\_OPTS

Restart=on-failure

LimitNOFILE=65536

[Install]

WantedBy=multi-user.target

EOF

启动服务并设置为开机自启动

systemctl daemon-reload

systemctl enable kube-apiserver

systemctl start kube-apiserver

systemctl status kube-apiserver

授权kubelet-bootstrap用户允许请求证书

kubectl create clusterrolebinding kubelet-bootstrap \

--clusterrole=system:node-bootstrapper --user=kubelet-bootstrap