14.2.创建 kubeconfig 文件

要创建 kubeconfig 文件的话，我们需要一个私钥，以及集群 CA 授权颁发的证书。如同我们要到公安局(权威机构)去申请身份证，公安局审核之后给我们颁发身份证，这个身份证可以作为证明身份的有效证件，而不能自己随便印一张名片作为有效证件。

同理我们不能直接用私钥生成公钥，而必须是用私钥生成证书请求文件（申请书），然后根据证书请求文件向 CA（权威机构）申请证书(身份证)，CA 审核通过之后会颁发证书。

下面开始创建创建整个过程。

14.2.1.申请证书

```
[root@vms10 ~]# mkdir role ; cd role
```

创建私钥，名字为 john.key

```
[root@vms10 role]# openssl genrsa -out john.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
[root@vms10 role]#
```

利用刚生成的私有 john.key 生成证书请求文件 john.csr：

```
[root@vms10 role]# openssl req -new -key john.key -out john.csr -subj
"/CN=john/O=cka2020"
[root@vms10 role]#
```

特别注意，这里 CN 的值 john，就是后面我们授权的用户。

对证书请求文件进行 base64 编码：

```
[root@vms10 role]# cat john.csr | base64 | tr -d "\n"
LS0tLS1CRUdJTiB...大量输出...TVC0tLS0tCg==[root@vms10 role]#
```

编写申请证书请求文件的 yaml 文件

```
[root@vms10 role]# cat csr.yaml
apiVersion: certificates.k8s.io/v1beta1
kind: CertificateSigningRequest
metadata:
  name: john
spec:
  groups:
  - system:authenticated
  #signerName: kubernetes.io/legacy-aa #注意这行是被注释掉的
  request:  LS0tLS1CRUdJTiB...大量内容...TVC0tLS0tCg==
  usages:
```

```
  - client auth
[root@vms10 role]#
```

注意这里 apiVersion 要带 beta1，否则 signerName 那行就不能注释掉，但这样的话后面的操作就不能获取到证书。这里 request 里的是 base64 编码之后的证书请求文件。

申请证书

```
[root@vms10 role]# kubectl apply -f csr.yaml
      …输出…
certificatesigningrequest.certificates.k8s.io/john created
[root@vms10 role]#
```

查看已经发出证书申请请求：

```
[root@vms10 role]# kubectl get csr
NAME     AGE     SIGNERNAME                            REQUESTOR           CONDITION
john     36s     kubernetes.io/legacy-unknown     kubernetes-admin     Pending
[root@vms10 role]#
```

批准证书：

```
[root@vms10 role]# kubectl certificate approve john
certificatesigningrequest.certificates.k8s.io/john approved
[root@vms10 role]#
```

查看审批通过的证书：

```
[root@vms10 role]# kubectl get csr
NAME     AGE     SIGNERNAME                            REQUESTOR           CONDITION
john     75s     kubernetes.io/legacy-aa     kubernetes-admin           Approved,Issued
[root@vms10 role]#
```

查看证书：

```
[root@vms10 role]# kubectl get csr/john    -o jsonpath='{.status.certificate}'
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCakNDQWU2Z0F3SUJBZ0IRQmhRN0x
WMThFTz…大量输出…LS0tLUVORCBDRVJUSUZJQ0FURS0tLS0tCg==[root@vms10 role]#
```

导出证书文件：

```
[root@vms10 role]# kubectl get csr/john    -o jsonpath='{.status.certificate}' | base64 -d >
john.crt
[root@vms10 role]#
```

给用户授权，这里给 john 一个集群角色 cluster-role，这样 john 具有管理员权限，具体权限管理后面会讲。

```
[root@vms10 role]# kubectl create clusterrolebinding test1 --clusterrole=cluster-admin
--user=john
clusterrolebinding.rbac.authorization.k8s.io/test1 created
[root@vms10 role]#
```

14.2.2.创建 kubeconfig 文件

创建 kubeconfig 模板文件

```
[root@vms10 role]# cat kc1
apiVersion: v1
```

```
kind: Config
preferences: {}
clusters:
- cluster:
  name: cluster1
users:
- name: john
contexts:
- context:
  name: context1
  namespace: default
current-context: "context1"
[root@vms10 role]#
```

这个模板文件里，clusters 字段指定 kubernetes 集群的信息，users 指定用户，contexts 用于指定上下文包括用户默认所在的命名空间等信息。

拷贝 CA 证书

```
[root@vms10 role]# cp /etc/kubernetes/pki/ca.crt .
[root@vms10 role]#
```

设置集群字段

```
[root@vms10 role]# kubectl config --kubeconfig=kc1 set-cluster cluster1
--server=https://192.168.26.10:6443 --certificate-authority=ca.crt --embed-certs=true
Cluster "cluster1" set.
[root@vms10 role]#
```

这里--embed-certs=true 的意思是把证书内容写入到此 kubeconfig 文件里。

设置用户字段

```
[root@vms10 role]# kubectl config --kubeconfig=kc1 set-credentials john
--client-certificate=john.crt --client-key=john.key --embed-certs=true
User "john" set.
[root@vms10 role]#
```

设置上下文字段

```
[root@vms10 role]# kubectl config --kubeconfig=kc1 set-context context1
--cluster=cluster1 --namespace=default --user=john
Context "context1" modified.
[root@vms10 role]#
```

这样 kubeconfig 文件就创建完毕了，下面开始验证 kubeconfig 文件。

14.2.3.验证 kubeconfig 文件

检查 john 是否具有 list 当前命名空间里的 pod 的权限

```
[root@vms10 role]# kubectl auth can-i list pods --as john
yes
[root@vms10 role]#
```

检查 john 是否具有 list 命名空间 kube-system 里 pod 的权限

```
[root@vms10 role]# kubectl auth can-i list pods -n kube-system --as john
yes
[root@vms10 role]#
```

把这个 kubeconfig 文件拷贝到 vms11 上

```
[root@vms10 role]# scp kc1 vms11:~
kc1                           100% 5506         4.1MB/s     00:00
[root@vms10 role]#
```

在 vms11 上用此 kubeconfig 文件执行集群命令

```
[root@vms11 ~]# kubectl --kubeconfig=kc1 get nodes
NAME                STATUS     ROLES       AGE     VERSION
vms10.rhce.cc       Ready      master      8d      v1.19.2
vms11.rhce.cc       Ready      <none>      8d      v1.19.2
vms12.rhce.cc       Ready      <none>      8d      v1.19.2
[root@vms11 ~]#
```

可以正常使用。

上面是使用 kubeconfig 方式来登录，当然我们也能用用户名密码的登录方式来登录 kubernetes。

下面就开始来讲如何用用户及密码的方式来登录 kubernetes。