tags: TLS, CA, x509

02. 创建 CA 根证书和秘钥

- 02. 创建 CA 根证书和秘钥
  - 安装 cfssl 工具集
  - 创建配置文件
  - 创建证书签名请求文件
  - 生成 CA 证书和私钥
  - 分发证书文件
  - 参考

为确保安全，kubernetes 系统各组件需要使用 x509 证书对通信进行加密和认证。

CA (Certificate Authority) 是自签名的根证书，用来签名后续创建的其它证书。

CA 证书是集群所有节点共享的，只需要创建一次，后续用它签名其它所有证书。

本文档使用 CloudFlare 的 PKI 工具集 cfssl 创建所有证书。

注意：如果没有特殊指明，本文档的所有操作均在 **zhangjun-k8s-01** 节点上执行。

安装 cfssl 工具集

```
sudo mkdir -p /opt/k8s/cert && cd /opt/k8s/work

wget
https://github.com/cloudflare/cfssl/releases/download/v1.4.1/cfssl_1.4.1_linux_amd64
mv cfssl_1.4.1_linux_amd64 /opt/k8s/bin/cfssl

wget
https://github.com/cloudflare/cfssl/releases/download/v1.4.1/cfssljson_1.4.1_linux_am
d64
mv cfssljson_1.4.1_linux_amd64 /opt/k8s/bin/cfssljson

wget https://github.com/cloudflare/cfssl/releases/download/v1.4.1/cfssl-
certinfo_1.4.1_linux_amd64
mv cfssl-certinfo_1.4.1_linux_amd64 /opt/k8s/bin/cfssl-certinfo

chmod +x /opt/k8s/bin/*
export PATH=/opt/k8s/bin:$PATH
```

创建配置文件

CA 配置文件用于配置根证书的使用场景 (profile) 和具体参数 (usage，过期时间、服务端认证、客户端认证、加密等):

```
cd /opt/k8s/work
cat > ca-config.json <<EOF
{
  "signing": {
    "default": {
      "expiry": "87600h"
    },
    "profiles": {
      "kubernetes": {
        "usages": [
            "signing",
            "key encipherment",
            "server auth",
            "client auth"
        ],
        "expiry": "876000h"
      }
    }
  }
}
EOF
```

- `signing`：表示该证书可用于签名其它证书（生成的 `ca.pem` 证书中 `CA=TRUE`）；
- `server auth`：表示 client 可以用该该证书对 server 提供的证书进行验证；
- `client auth`：表示 server 可以用该该证书对 client 提供的证书进行验证；
- `"expiry": "876000h"`：证书有效期设置为 100 年；

创建证书签名请求文件

```
cd /opt/k8s/work
cat > ca-csr.json <<EOF
{
  "CN": "kubernetes-ca",
  "key": {
    "algo": "rsa",
    "size": 2048
  },
  "names": [
    {
      "C": "CN",
      "ST": "BeiJing",
      "L": "BeiJing",
```

```
      "O": "k8s",
      "OU": "opsnull"
    }
  ],
  "ca": {
    "expiry": "876000h"
 }
}
EOF
```

- CN：Common Name：kube-apiserver 从证书中提取该字段作为请求的用户名 **(User Name)**，浏览器使用该字段验证网站是否合法；
- O：Organization：kube-apiserver 从证书中提取该字段作为请求用户所属的组 **(Group)**；
- kube-apiserver 将提取的 User、Group 作为 RBAC 授权的用户标识；

注意：

1. 不同证书 csr 文件的 CN、C、ST、L、O、OU 组合必须不同，否则可能出现 PEER'S CERTIFICATE HAS AN INVALID SIGNATURE 错误；
2. 后续创建证书的 csr 文件时，CN 都不相同（C、ST、L、O、OU 相同），以达到区分的目的；

生成 CA 证书和私钥

```
cd /opt/k8s/work
cfssl gencert -initca ca-csr.json | cfssljson -bare ca
ls ca*
```

分发证书文件

```
cd /opt/k8s/work
source /opt/k8s/bin/environment.sh
for node_ip in ${NODE_IPS[@]}
  do
    echo ">>> ${node_ip}"
    ssh root@${node_ip} "mkdir -p /etc/kubernetes/cert"
    scp ca*.pem ca-config.json root@${node_ip}:/etc/kubernetes/cert
  done
```

参考

1. 各种 CA 证书类型