kubectl 与 apiserver https 安全端口通信，apiserver 对提供的证书进行认证和授权。
kubectl 作为集群的管理工具，需要被授予最高权限。这里创建具有最高权限的 admin
证书。

创建证书签名请求

```
[root@k8s-master2 pki]# cat admin-csr.json
{
   "CN":"admin",
   "hosts":[

   ],
   "key":{
      "algo":"rsa",
      "size":2048
   },
   "names":[
      {
         "C":"CN",
         "ST":"BeiJing",
         "L":"BeiJing",
         "O":"system:masters",
         "OU":"k8s"
      }
   ]
}
```

O 为 system:masters ， kube-apiserver 收到该证书后将请求的 Group 设置为
system:masters；

生成证书和私钥：

../cfssl gencert -ca=ca.pem -ca-key=ca-key.pem --config=ca-config.json -profile=kubernetes admin-csr.json | ../cfssljson -bare admin

创建 kubeconfig 文件

kubeconfig 为 kubectl 的配置文件，包含访问 apiserver 的所有信息，如 apiserver 地址、CA 证书和自身使用的证书；

```
kubectl config set-cluster kubernetes --certificate-authority=/etc/kubernetes/pki/ca.pem --embed-certs=true --server=http://172.16.102.100:8443
kubectl config set-credentials admin --client-certificate=/etc/kubernetes/pki/admin.pem --embed-certs=true --client-key=/etc/kubernetes/pki/admin-key.pem
kubectl config set-context kubernetes --cluster=kubernetes --user=admin
kubectl config use-context kubernetes
```

--certificate-authority：验证 kube-apiserver 证书的根证书；
--client-certificate 、 --client-key：刚生成的 admin 证书和私钥，连接 kube-apiserver 时使用；
--embed-certs=true：将 ca.pem 和 admin.pem 证书内容嵌入到生成的 kubectl.kubeconfig 文件中(不加时，写入的是证书文件路径)；