创建kubeconfig文件

注意：请先参考 安装kubectl命令行工具，先在 master 节点上安装 kubectl 然后再进行下面的操作。

kubelet 、kube-proxy 等Node机器上的进程与 Master 机器的kube-apiserver进程通信时需要认证和授权；

kubernetes 1.4 开始支持由kube-apiserver为客户端生成 TLS 证书的 TLS Bootstrapping功能，这样就不需要为每个客户端生成证书了；该功能**当前仅支持为kubelet生成证书** ；

因为master节点和node节点复用，所有在这一步其实已经安装了kubectl。参考安装kubectl命令行工具。

以下操作只需要在master节点上执行，生成的*.kubeconfig文件可以直接拷贝到node节点的/etc/kubernetes目录下。

创建TLS Bootstrapping Token

Token auth file

Token 可以是任意的包涵128 bit的字符串，可以使用安全的随机数发生器生成。

```
[root@vlnx251101 ssl]# export BOOTSTRAP_TOKEN=$(head -c 16 /dev/urandom | od -An -t x | tr -d ' ')
[root@vlnx251101 ssl]# cat > token.csv <<EOF
${BOOTSTRAP_TOKEN},kubelet-bootstrap,10001,"system:kubelet-bootstrap"
EOF
```

注意：在进行后续操作前请检查 `token.csv` 文件，确认其中的 `${BOOTSTRAP_TOKEN}` 环境变量已经被真实的值替换。

BOOTSTRAP_TOKEN 将被写入到 kube-apiserver 使用的 token.csv 文件和 kubelet 使用的 bootstrap.kubeconfig 文件，如果后续重新生成了 BOOTSTRAP_TOKEN，则需要：

1. 更新 token.csv 文件，分发到所有机器 (master 和 node) 的 /etc/kubernetes/ 目录下，分发到node节点上非必需；
2. 重新生成 bootstrap.kubeconfig 文件，分发到所有 node 机器的 /etc/kubernetes/ 目录下；
3. 重启 kube-apiserver 和 kubelet 进程；
4. 重新 approve kubelet 的 csr 请求；

将token.csv **发到所有机器（Master 和Node）**的 /etc/kubernetes/ 目录。

```
[root@vlnx251101 ssl]# scp token.csv
192.168.251.101:/etc/kubernetes/
```

创建kubelet bootstrapping kubeconfig 文件

执行下面的命令时需要先安装kubectl命令（参见master）

```
[root@vlnx251101 ~]# cd /etc/kubernetes
[root@vlnx251101 kubernetes]# export
KUBE_APISERVER="https://192.168.251.101:6443"
```

设置集群参数

```
[root@vlnx251101 kubernetes]# kubectl config set-cluster
kubernetes \
--certificate-authority=/etc/kubernetes/ssl/ca.pem \
--embed-certs=true \
```

```
--server=${KUBE_APISERVER} \
--kubeconfig=bootstrap.kubeconfig
```

设置客户端认证参数

```
[root@vlnx251101 kubernetes]# kubectl config set-
credentials kubelet-bootstrap \
--token=${BOOTSTRAP_TOKEN} \
--kubeconfig=bootstrap.kubeconfig
```

设置上下文参数

```
[root@vlnx251101 kubernetes]# kubectl config set-context
default \
--cluster=kubernetes \
--user=kubelet-bootstrap \
--kubeconfig=bootstrap.kubeconfig
```

设置默认上下文

```
[root@vlnx251101 kubernetes]# kubectl config use-context
default --kubeconfig=bootstrap.kubeconfig
```

- `--embed-certs` 为true时表示将 `certificate-authority`证书写入到生成的 `bootstrap.kubeconfig` 文件中；
- 设置客户端认证参数时没有指定密钥和证书，后续由`kube-apiserver`自动生成；

创建kube-proxy kubeconfig

```
[root@vlnx251101 kubernetes]# export
KUBE_APISERVER="https://192.168.251.101:6443"
```

设置集群参数

```
[root@vlnx251101 kubernetes]# kubectl config set-cluster
kubernetes \
--certificate-authority=/etc/kubernetes/ssl/ca.pem \
--embed-certs=true --server=${KUBE_APISERVER} \
--kubeconfig=kube-proxy.kubeconfig
```

设置客户端认证参数

```
[root@vlnx251101 kubernetes]# kubectl config set-
credentials kube-proxy \
--client-certificate=/etc/kubernetes/ssl/kube-proxy.pem \
--client-key=/etc/kubernetes/ssl/kube-proxy-key.pem \
--embed-certs=true \
--kubeconfig=kube-proxy.kubeconfig
```

设置上下文参数

```
[root@vlnx251101 kubernetes]# kubectl config set-context
default \
--cluster=kubernetes \
--user=kube-proxy \
--kubeconfig=kube-proxy.kubeconfig
```

设置默认上下文

```
[root@vlnx251101 kubernetes]# kubectl config use-context
default --kubeconfig=kube-proxy.kubeconfig
```

- 设置集群参数和客户端认证参数时`--embed-certs` 都为 `true` ，这会 将`certificate-authority` 、`client-certificate`和`client-key` 指向的证书文件内容写入到生成的`kube-proxy.kubeconfig`文件中

- kube-proxy.pem证书中CN为 system:kube-proxy， kube-apiserver 预定义的RoleBinding cluster-admin 将User system:kube-proxy 与 Role system:node-proxier 绑定，该 Role 授予了调用kube-apiserver Proxy 相关API的权限；

分发kubeconfig文件

将两个kubeconfig文件**分发到所有 Node 机器**的 /etc/kubernetes/目录

```
[root@vlnx251101 kubernetes]# scp bootstrap.kubeconfig
kube-proxy.kubeconfig 192.168.251.102:/etc/kubernetes/
```

创建kubectl kubeconfig文件

```
[root@vlnx251101 ~]# export
KUBE_APISERVER="https://192.168.251.101:6443"
```

设置集群参数

```
[root@vlnx251101 ~]# kubectl config set-cluster kubernetes
\
--certificate-authority=/etc/kubernetes/ssl/ca.pem \
--embed-certs=true \
--server=${KUBE_APISERVER}
```

设置客户端认证参数

```
[root@vlnx251101 ~]# kubectl config set-credentials admin
\
--client-certificate=/etc/kubernetes/ssl/admin.pem \
--embed-certs=true \
```

```
--client-key=/etc/kubernetes/ssl/admin-key.pem
```

设置上下文参数

```
[root@vlnx251101 ~]# kubectl config set-context kubernetes
\
--cluster=kubernetes \
--user=admin
```

设置默认上下文

```
[root@vlnx251101 ~]# kubectl config use-context kubernetes
```

- admin.pem证书 OU 字段值为system:masters， kube-apiserver预定义的 RoleBinding cluster-admin将 Group system:masters 与Role cluster-admin绑定，该 Role授予了调用 kube-apiserver相关 API的权限；
- 生成的kubeconfig 被保存到 ~/.kube/config 文件；

分发到所有node节点

```
[root@vlnx251101 ~]# scp ~/.kube/config
192.168.251.101:/etc/kubernetes/kubelet.kubeconfig
```