kubeconfig 简介

使用 kubeconfig 文件来组织有关集群、用户、命名空间和身份认证机制的信息。`kubectl` 命令行工具使用 kubeconfig 文件来查找选择集群所需的信息，并与集群的 API 服务器进行通信。

默认情况下，`kubectl` 在 `$HOME/.kube` 目录下查找名为 `config` 的文件。 您可以通过设置 `KUBECONFIG` 环境变量或者设置 `--kubeconfig` 参数来指定其他 kubeconfig 文件。

```
kubectl config SUBCOMMAND

选项
      --kubeconfig="": 使用特定的配置文件。

继承自父命令的选项
      --alsologtostderr[=false]: 同时输出日志到标准错误控制台和文件。
      --api-version="": 和服务端交互使用的API版本。
      --certificate-authority="": 用以进行认证授权的.cert文件路径。
      --client-certificate="": TLS使用的客户端证书路径。
      --client-key="": TLS使用的客户端密钥路径。
      --cluster="": 指定使用的kubeconfig配置文件中的集群名。
      --context="": 指定使用的kubeconfig配置文件中的环境名。
      --insecure-skip-tls-verify[=false]: 如果为true，将不会检查服务器凭证的有效性，这会导致你的HTTPS链接变得不安全。
      --kubeconfig="": 命令行请求使用的配置文件路径。
      --log-backtrace-at=:0: 当日志长度超过定义的行数时，忽略堆栈信息。
      --log-dir="": 如果不为空，将日志文件写入此目录。
      --log-flush-frequency=5s: 刷新日志的最大时间间隔。
      --logtostderr[=true]: 输出日志到标准错误控制台，不输出到文件。
      --match-server-version[=false]: 要求服务端和客户端版本匹配。
      --namespace="": 如果不为空，命令将使用此namespace。
      --password="": API Server进行简单认证使用的密码。
  -s, --server="": Kubernetes API Server的地址和端口号。
      --stderrthreshold=2: 高于此级别的日志将被输出到错误控制台。
      --token="": 认证到API Server使用的令牌。
      --user="": 指定使用的kubeconfig配置文件中的用户名。
      --username="": API Server进行简单认证使用的用户名。
      --v=0: 指定输出日志的级别。
      --vmodule=: 指定输出日志的模块，格式如下: pattern=N，使用逗号分隔。
```

生成kubeconfig的配置步骤

1、定义变量

```
export KUBE_APISERVER="https://172.20.0.2:6443"
```

2、设置集群参数

```
kubectl config set-cluster kubernetes   --certificate-
authority=/etc/kubernetes/ssl/ca.pem   --embed-certs=true   --
server=${KUBE_APISERVER} --kubeconfig=/root/config.conf
```

> 说明：集群参数主要设置了所需要访问的集群的信息。
>
> 使用set-cluster设置了需要访问的集群，如上为kubernetes；
>
> --certificate-authority设置了该集群的公钥；
>
> --embed-certs为true表示将--certificate-authority证书写入到kubeconfig中；
>
> --server则表示该集群的kube-apiserver地址

3、设置客户端认证参数

```
kubectl config set-credentials admin   --client-
certificate=/etc/kubernetes/ssl/admin.pem   --embed-certs=true   --
client-key=/etc/kubernetes/ssl/admin-key.pem --
kubeconfig=/root/config.conf
```

> 说明：用户参数主要设置用户的相关信息，主要是用户证书。
>
> 如上的用户名为admin，证书为：/etc/kubernetes/ssl/admin.pem，私钥为：/etc/kubernetes/ssl/admin-key.pem。
>
> 注意客户端的证书首先要经过集群CA的签署，否则不会被集群认可。
>
> 此处使用的是ca认证方式，也可以使用token认证，如kubelet的 TLS Boostrap机制下的bootstrapping使用的就是token认证方式。

4、设置上下文参数

```
kubectl config set-context kubernetes   --cluster=kubernetes   --
user=admin --kubeconfig=/root/config.conf
```

> 说明：上下文参数将**集群参数**和**用户参数**关联起来。

> 如上面的上下文名称为kubenetes，集群为kubenetes，用户为admin，表示使用admin的用户凭证来访问kubenetes集群的default命名空间，也可以增加--namspace来指定访问的命名空间。

5、设置默认上下文

```
kubectl config use-context kubernetes  --kubeconfig=/root/config.conf
```

实操

设置k8s dashboard 的访问

```
#  创建命名空间 devops
# $ kubectl create namespace devops
# namespace/devops created


# 在命名空间 devops 下创建 k8s-dashboard-admin 用户
$ kubectl create sa k8s-dashboard-admin -n kube-system
serviceaccount/k8s-dashboard-admin created


# 查看 集群角色 cluster-admin 这个角色具有所有权限
$kubectl get clusterrole
NAME                                                   CREATED AT
calico-kube-controllers                                2021-08-09T02:50:10Z
calico-node                                            2021-08-09T02:50:10Z
coredns                                                2021-08-09T03:25:13Z
system:aggregated-metrics-reader                       2021-08-09T03:27:02Z
system:metrics-server                                  2021-08-09T03:27:03Z
cluster-admin                                          2021-08-09T03:28:17Z
system:discovery                                       2021-08-09T03:28:17Z
system:monitoring                                      2021-08-09T03:28:17Z
```

```
system:basic-user
     2021-08-09T03:28:17Z
... ...
... ...


# 在 devops 命名空间下创建一个角色绑定 k8s-dashboard-admin-rolebinding，绑定 cluster-admin ，赋予 devops 下 k8s-dashboard-admin cluster-admin 的所有权限
$ kubectl create rolebinding k8s-dashboard-admin-rolebinding -n kube-system --clusterrole=cluster-admin --serviceaccount=kube-system:k8s-dashboard-admin
rolebinding.rbac.authorization.k8s.io/k8s-dashboard-admin-rolebinding created


# 查看 devops 名称空间下 secret
$ kubectl get secret -n kube-system
NAME                                TYPE
     DATA    AGE
default-token-s4zf6                 kubernetes.io/service-account-token    3      5m33s
k8s-dashboard-admin-token-5df6v     kubernetes.io/service-account-token    3      4m42s


# 查看 k8s-dashboard-admin-token-5df6v 这个 secret 的详细信息，以及对应的token
$ kubectl describe secret k8s-dashboard-admin-token-5df6v -n kube-system
Name:         k8s-dashboard-admin-token-5df6v
Namespace:    devops
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: k8s-dashboard-admin
              kubernetes.io/service-account.uid: a7a32093-1215-4d06-9ab4-dddd2cd31b2d

Type:  kubernetes.io/service-account-token

Data
====
```

```
token:
eyJhbGciOiJSUzI1NiIsImtpZCI6InFvYVB3NO5FbjUwRDJWRm5oXzJxOE1RRHB0dm
N2bON5MkYzRHhQUW55RXcifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2Nvd
W5OIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZXZ
vcHMiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW5OL3NlY3JldC5uYW1lIjoia
zhzLWRhc2hib2FyZC1hZG1pbi10b2tlbi01ZGY2diIsImt1YmVybmV0ZXMuaW8vc2V
ydmljZWFjY291bnQvc2VydmljZS1hY2NvdW5OLm5hbWUiOiJrOHMtZGFzaGJvYXJkL
WFkbWluIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY29
1bnQudWlkIjoiYTdhMzIwOTMtMTIxNS00ZDA2LTlhYjQtZGRkZDJjZDMxYjJkIiwic
3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW5OOmRldm9wczprOHMtZGFzaGJvYXJkLWF
kbWluIn0.epzryCp4kgAWsHifUrFGj4SSxrFksBw8NyghC_y7uZ07PaYFakZuyKxgy
qi12FYcAMfXmFtbMWM_Fu7WFI6uQRZKkF4ruiqyLOqrXjrc0kjHx0Cn8NoquuAXjvM
pme2c-
FZ4SOBxtKJ3I40r04nkXFNQAgvgrAvInKvQqEcEhYuVqR0MwWYCCy2k87O8lOv9wBW
gX5EN7_-SKSWbKsaG3aF_rEOxJF3oUBvxzzoKrqjRuavs7r8e4TQp6-
Q8QNBrQ1i7TR1Ud93eWH1BYK_8rxakK8AHny1BFLWBfIsBiuH112-
6yKygg_dDL6yy8btLwQxeRbbQCe7p_4633YGC_g
ca.crt:     1123 bytes
namespace:  6 bytes
```

```
# 设置集群
$ kubectl config set-cluster microk8s-local --embed-certs --
certificate-authority=/var/snap/microk8s/current/certs/ca.crt --
server=https://127.0.0.1:16443 --kubeconfig=./dashboard.conf
Cluster "microk8s-local" set.

# 设置用户认证，这里使用 token 方式
$ kubectl config set-credentials k8s-dashboard-admin --
token=$KD_TOKEN --kubeconfig=./dashboard.conf
User "k8s-dashboard-admin" set.

# 设置上下文
$ kubectl config set-context k8s-dashboard --cluster=microk8s-
local --user=k8s-dashboard-admin --namespace=kube-system --
kubeconfig=./dashboard.conf
Context "k8s-dashboard" created.

# 设置默认上下文
$ kubectl config use-context k8s-dashboard --
kubeconfig=./dashboard.conf
Switched to context "k8s-dashboard".
```