CS571
Mansi Shah
(19526)
Week12

Talking to the API server from within a pod

Running a pod to try out communication with the API server

```
mansishah@macbookpro downward % vim curl.yaml
mansishah@macbookpro downward % cat curl.yaml
apiVersion: v1
kind: Pod
metadata:
  name: curl
spec:
  containers:
  - name: main
    image: tutum/curl
    command: ["sleep", "9999999"]
mansishah@macbookpro downward %
```

```
    command: ["sleep", "9999999"]
mansishah@macbookpro downward % kubectl create -f curl.yaml
pod/curl created
mansishah@macbookpro downward % kubectl get po
NAME          READY   STATUS              RESTARTS   AGE
curl          0/1     ContainerCreating   0          9s
downward      1/1     Running             0          119m
my-job-nglj9  0/1     Completed           0          41m
mansishah@macbookpro downward % kubectl get po
NAME          READY   STATUS      RESTARTS   AGE
curl          1/1     Running     0          62s
downward      1/1     Running     0          120m
my-job-nglj9  0/1     Completed   0          42m
mansishah@macbookpro downward %
```

Finding the API server's address

```
mansishah@macbookpro downward % kubectl get po
NAME           READY   STATUS      RESTARTS   AGE
curl           1/1     Running     0          3m32s
downward       1/1     Running     0          3h29m
my-job-nglj9   0/1     Completed   0          131m
mansishah@macbookpro downward % kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   17d
mansishah@macbookpro downward % kubectl exec -it curl bash
root@curl:/# env | grep KUBERNETES_SERVICE
KUBERNETES_SERVICE_PORT=443
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT_HTTPS=443
root@curl:/#
```

After creating the pod, run kubectl exec to run a bash shell inside its container:

```
root@curl:/# curl https://kubernetes
curl: (60) SSL certificate problem: unable to get local issuer certificate
More details here: http://curl.haxx.se/docs/sslcerts.html

curl performs SSL certificate verification by default, using a "bundle"
```

Verifying the server's identity

```
root@curl:/# ls /var/run/secrets/kubernetes.io/serviceaccount/
ca.crt  namespace  token
```

```
root@curl:/# curl --cacert /var/run/secrets/kubernetes.io/serviceaccount/ca.crt https://kubernetes
{
  "kind": "Status",
  "apiVersion": "v1",
  "metadata": {

  },
  "status": "Failure",
  "message": "forbidden: User \"system:anonymous\" cannot get path \"/\"",
  "reason": "Forbidden",
  "details": {

  },
  "code": 403
}root@curl:/#
```

```
}root@curl:/# export CURL_CA_BUNDLE=/var/run/secrets/kubernetes.io/serviceaccount/ca.crt
root@curl:/# curl https://kubernetes
{
  "kind": "Status",
  "apiVersion": "v1",
  "metadata": {

  },
  "status": "Failure",
  "message": "forbidden: User \"system:anonymous\" cannot get path \"/\"",
  "reason": "Forbidden",
  "details": {

  },
  "code": 403
}root@curl:/#
```

Authenticating with the API server

```
}root@curl:/# TOKEN=$(cat /var/run/secrets/kubernetes.io/serviceaccount/token)
root@curl:/# curl -H "Authorization: Bearer $TOKEN" https://kubernetes
{
  "paths": [
    "/api",
    "/api/v1",
    "/apis",
    "/apis/",
    "/apis/admissionregistration.k8s.io",
    "/apis/admissionregistration.k8s.io/v1",
    "/apis/admissionregistration.k8s.io/v1beta1",
    "/apis/apiextensions.k8s.io",
    "/apis/apiextensions.k8s.io/v1",
    "/apis/apiextensions.k8s.io/v1beta1",
    "/apis/apiregistration.k8s.io",
    "/apis/apiregistration.k8s.io/v1",
    "/apis/apiregistration.k8s.io/v1beta1",
    "/apis/apps",
    "/apis/apps/v1",
    "/apis/authentication.k8s.io",
    "/apis/authentication.k8s.io/v1",
    "/apis/authentication.k8s.io/v1beta1",
    "/apis/authorization.k8s.io",
    "/apis/authorization.k8s.io/v1",
    "/apis/authorization.k8s.io/v1beta1",
    "/apis/autoscaling",
    "/apis/autoscaling/v1",
```

- Disabling Role-Based Access Control (RBAC)
- Getting the namespace the pod is running in

```
mansishah@macbookpro downward % kubectl create clusterrolebinding permissive-binding \
> --clusterrole=cluster-admin \
> --group=system:serviceaccounts
Error from server (AlreadyExists): clusterrolebindings.rbac.authorization.k8s.io "permissive-binding" already exists
mansishah@macbookpro downward % kubectl exec -it curl bash
root@curl:/# curl -H "Authorization: Bearer $TOKEN" https://kubernetes/api/v1/namespaces/$NS/pods
curl: (60) SSL certificate problem: unable to get local issuer certificate
More details here: http://curl.haxx.se/docs/sslcerts.html

curl performs SSL certificate verification by default, using a "bundle"
 of Certificate Authority (CA) public keys (CA certs). If the default
 bundle file isn't adequate, you can specify an alternate file
 using the --cacert option.
If this HTTPS server uses a certificate signed by a CA represented in
 the bundle, the certificate verification probably failed due to a
 problem with the certificate (it might be expired, or the name might
 not match the domain name in the URL).
If you'd like to turn off curl's verification of the certificate, use
 the -k (or --insecure) option.
root@curl:/#
```