
Star 898 Fork 152 Watch 52 Follow @collabnix 0

# Accessing the Kubernetes API

Kubernetes - Beginners | Intermediate | Advanced

View on GitHub Join Slack

## Accessing the Kubernetes API

### Pre-requisite

- Setup 5 Node Kubernetes Cluster

When accessing the Kubernetes API for the first time, use the Kubernetes command-line tool, `kubectl`. To access a cluster, you need to know the location of the cluster and have credentials to access it.

### Checking the location & credentials

```
[node1 kubelabs]$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
```

```
    server: https://192.168.0.18:6443
  name: kubernetes
contexts:
- context:
    cluster: kubernetes
    user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate
```

## Directly accessing the REST API

The kubectl handles locating and authenticating to the API server. If you want to directly access the REST API with an HTTP client like `curl` or `wget`, or a browser, there are multiple ways you can locate and authenticate against the API server:

- Run `kubectl` in proxy mode (recommended). This method is recommended, since it uses the stored apiserver location and verifies the identity of the API server using a self-signed cert. No man-in-the-middle (MITM) attack is possible using this method.
- Provide the location and credentials directly to the HTTP client. This works with client code that is confused by proxies. To protect against man in the middle attacks, you'll need to import a root cert into your browser. Using the Go or Python client libraries provides accessing `kubectl` in proxy mode.

## Using kubectl proxy

The following command runs `kubectl` in a mode where it acts as a reverse proxy. It handles locating the API server and authenticating.

```
kubectl proxy --port=8080 &
```

Then you can explore the API with `curl`, `wget`, or a browser, like so:

```
curl http://localhost:8080/api/
```

The output is similar to this:

```
[node1 kubelabs]$ curl http://localhost:8080/api/
{
  "kind": "APIVersions",
  "versions": [
    "v1"
  ],
  "serverAddressByClientCIDRs": [
    {
      "clientCIDR": "0.0.0.0/0",
      "serverAddress": "192.168.0.18:6443"
    }
  ]
```

## Without kubectl proxy

```
  # Check all possible clusters, as you .KUBECONFIG may have multiple contex
kubectl config view -o jsonpath='{"Cluster name\tServer\n"}{range .clusters[

# Select name of cluster you want to interact with from above output:
export CLUSTER_NAME="some_server_name"

# Point to the API server referring the cluster name
APISERVER=$(kubectl config view -o jsonpath="{.clusters[?(@.name==\"$CLUSTER

# Gets the token value
TOKEN=$(kubectl get secrets -o jsonpath="{.items[?(@.metadata.annotations['k

# Explore the API with TOKEN
curl -X GET $APISERVER/api --header "Authorization: Bearer $TOKEN" --insecur
```

Next »

## Join KubeDaily

**10** Members Online

### Support

MEMBERS ONLINE

- h3ll_boy
- Heisenberg
- Klu
- MEE6
- Nitinkashyap — Apex Legends
- ojaswa
- Parmeshwar
- prasad
- trimankaur
- vikas027

Free voice chat from Discord

Connect

Tweets by collabnix

**kubelabs is maintained by collabnix.**

This page was generated by GitHub Pages.