Install Kubectl to Control a Kubernetes Cluster

Project Astra

Ben Cammett
September 22, 2020

NetApp®

Table of Contents

Install Kubectl to Control a Kubernetes Cluster

Kubectl is the command line interface for interacting with a Kubernetes cluster.
Overview

Kubectl gets its cluster connection and login information from the cluster's `kubeconfig` file. In order to use kubectl you will need to:

1. Install kubectl on your local machine.
2. Download the cluster's kubeconfig file.
3. Move the kubeconfig file to your user directory, then update your shell's environment.

More kubectl documentation can be found on the official Kubernetes site.

Windows

The easiest way to install kubectl on Windows is to use the Chocolatey package manager. You can install Chocolatey with a single command from a PowerShell or cmd window. The installation instructions for both Administrator and non-Administrator rights are here on the Chocolatey website.

After you have installed Chocolatey, open PowerShell or cmd, then install kubectl with the command:

```
choco install kubernetes-cli
```

Move to your home directory:

```
cd C:\users\[your username]
```

For example:

```
cd C:\users\jdoe
```

Create the `.kube` directory:

```
mkdir .kube
```

Move to the `.kube` directory:

```
cd .kube
```

Download your cluster's `kubeconfig` file. Move this file to the `C:\users\yourusername\.kube` directory. Then update your shell environment to call the file:

```
$env:KUBECONFIG="C:\users\[your username]\.kube\[kubeconfig file name]"
```

For example:

```
$env:KUBECONFIG="C:\users\jdoe\.kube\kubeconfig"
```

Mac OS

You can install kubectl on MacOS using either the Homebrew or Macports package managers.

To install with Homebrew, open a Terminal window and use the command:

```
brew install kubernetes-cli
```

To install with Macports, open a Terminal window and use the command:

```
port install kubectl
```

Download your cluster's `kubeconfig` file. Move this file to your home directory:

```
mv /Users/username/Downloads/kubeconfig /Users/username/
```

Then update your shell environment to call the file:

```
export KUBECONFIG=/Users/username/kubeconfig
```

External Linux Server

Another option is to SSH to a server, install kubectl, then use it to control your Kubernetes cluster. If you have access to a Linux server, there are several advantages to this approach:

- If you have root on the server, you will be able to install the Kubernetes software without having to deal with any desktop computer permissions problems.

- Speaking of computer permissions issues, it's often the case that in order to run Kubernetes on a Windows computer, you need to run PowerShell on Windows as Administrator. If you are not able to run applications with administrator rights, you can avoid the issue by connecting to a Linux server with SSH and running kubectl from there.
- If you frequently find yourself using several different computers, you can SSH to the server from anywhere. This saves you from having to install Kubernetes on every machine you use.
- If you have customized your SSH program to your liking (fonts, colors, etc.) it can be nicer to use it than the stock terminal or PowerShell window.

To install kubectl on a Linux server, follow the instructions for your distribution.

Ubuntu, Debian, and HypriotOS

Install with the commands:

```
sudo apt-get update && sudo apt-get install -y apt-transport-https

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a
/etc/apt/sources.list.d/kubernetes.list

sudo apt-get update

sudo apt-get install -y kubectl
```

CentOS, RHEL, and Fedora

Install with the commands:

```
cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF

yum install -y kubectl
```

Download your cluster's `kubeconfig` file. Upload the file to the server. Then update your shell

environment to call the file:

```
export KUBECONFIG=/path/to/kubeconfig
```

For example:

```
export KUBECONFIG=/usr/home/jdoe/kubeconfig
```