Install and Set Up kubectl on Linux

Before you begin

You must use a kubectl version that is within one minor version difference of your cluster. For example, a v1.31 client can communicate with v1.30, v1.31, and v1.32 control planes. Using the latest compatible version of kubectl helps avoid unforeseen issues.

Install kubectl on Linux

The following methods exist for installing kubectl on Linux:

- Install kubectl binary with curl on Linux
- Install using native package management
- Install using other package management

Install kubectl binary with curl on Linux

1. Download the latest release with the command:

   x86-64 ARM64

   ```
   curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
   ```

   Note:
   To download a specific version, replace the `$(curl -L -s https://dl.k8s.io/release/stable.txt)` portion of the command with the specific version.

   For example, to download version 1.31.0 on Linux x86-64, type:

   ```
   curl -LO https://dl.k8s.io/release/v1.31.0/bin/linux/amd64/kubectl
   ```

   And for Linux ARM64, type:

   ```
   curl -LO https://dl.k8s.io/release/v1.31.0/bin/linux/arm64/kubectl
   ```

2. Validate the binary (optional)

   Download the kubectl checksum file:

   x86-64 ARM64

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl.sh
```

   Validate the kubectl binary against the checksum file:

```
echo "$(cat kubectl.sha256)  kubectl" | sha256sum --check
```

   If valid, the output is:

```
kubectl: OK
```

   If the check fails, `sha256` exits with nonzero status and prints output similar to:

```
kubectl: FAILED
sha256sum: WARNING: 1 computed checksum did NOT match
```

   Note:
   Download the same version of the binary and checksum.

3. Install kubectl

```
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

   Note:
   If you do not have root access on the target system, you can still install kubectl to the `~/.local/bin` directory:

```
chmod +x kubectl
mkdir -p ~/.local/bin
mv ./kubectl ~/.local/bin/kubectl
# and then append (or prepend) ~/.local/bin to $PATH
```

4. Test to ensure the version you installed is up-to-date:

```
kubectl version --client
```

   Or use this for detailed view of version:

```
kubectl version --client --output=yaml
```

Install using native package management

Debian-based distributions Red Hat-based distributions SUSE-based distributions

1. Update the `apt` package index and install packages needed to use the Kubernetes `apt` repository:

```
sudo apt-get update
# apt-transport-https may be a dummy package; if so, you can skip that package
sudo apt-get install -y apt-transport-https ca-certificates curl gnupg
```

2. Download the public signing key for the Kubernetes package repositories. The same signing key is used for all repositories so you can disregard the version in the URL:

```
# If the folder `/etc/apt/keyrings` does not exist, it should be created before the curl command, read the note belo
# sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kuberne
sudo chmod 644 /etc/apt/keyrings/kubernetes-apt-keyring.gpg # allow unprivileged APT programs to read this keyring
```

Note:
In releases older than Debian 12 and Ubuntu 22.04, folder `/etc/apt/keyrings` does not exist by default, and it should be created before the curl command.

3. Add the appropriate Kubernetes `apt` repository. If you want to use Kubernetes version different than v1.31, replace v1.31 with the desired minor version in the command below:

```
# This overwrites any existing configuration in /etc/apt/sources.list.d/kubernetes.list
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.31/deb/ /' |
sudo chmod 644 /etc/apt/sources.list.d/kubernetes.list   # helps tools such as command-not-found to work correctly
```

Note:
To upgrade kubectl to another minor release, you'll need to bump the version in `/etc/apt/sources.list.d/kubernetes.list` before running `apt-get update` and `apt-get upgrade`. This procedure is described in more detail in Changing The Kubernetes Package Repository.

4. Update `apt` package index, then install kubectl:

```
sudo apt-get update
sudo apt-get install -y kubectl
```

Install using other package management

Snap Homebrew

If you are on Ubuntu or another Linux distribution that supports the snap package manager, kubectl is available as a snap application.

```
snap install kubectl --classic
kubectl version --client
```

Verify kubectl configuration

In order for kubectl to find and access a Kubernetes cluster, it needs a kubeconfig file, which is created automatically when you create a cluster using kube-up.sh or successfully deploy a Minikube cluster. By default, kubectl configuration is located at `~/.kube/config`.

Check that kubectl is properly configured by getting the cluster state:

```
kubectl cluster-info
```

If you see a URL response, kubectl is correctly configured to access your cluster.

If you see a message similar to the following, kubectl is not configured correctly or is not able to connect to a Kubernetes cluster.

```
The connection to the server <server-name:port> was refused - did you specify the right host or port?
```

For example, if you are intending to run a Kubernetes cluster on your laptop (locally), you will need a tool like Minikube to be installed first and then re-run the commands stated above.

If `kubectl cluster-info` returns the url response but you can't access your cluster, to check whether it is configured properly, use:

```
kubectl cluster-info dump
```

Troubleshooting the 'No Auth Provider Found' error message

In Kubernetes 1.26, kubectl removed the built-in authentication for the following cloud providers' managed Kubernetes offerings. These providers have released kubectl plugins to provide the cloud-specific authentication. For instructions, refer to the following provider documentation:

- Azure AKS: kubelogin plugin
- Google Kubernetes Engine: gke-gcloud-auth-plugin

(There could also be other reasons to see the same error message, unrelated to that change.)

Optional kubectl configurations and plugins

Enable shell autocompletion

kubectl provides autocompletion support for Bash, Zsh, Fish, and PowerShell, which can save you a lot of typing.

Below are the procedures to set up autocompletion for Bash, Fish, and Zsh.

Bash Fish Zsh

Introduction

The kubectl completion script for Bash can be generated with the command `kubectl completion bash`. Sourcing the completion script in your shell enables kubectl autocompletion.

However, the completion script depends on **bash-completion**, which means that you have to install this software first (you can test if you have bash-completion already installed by running `type _init_completion`).

Install bash-completion

bash-completion is provided by many package managers (see here). You can install it with `apt-get install bash-completion` or `yum install bash-completion`, etc.

The above commands create `/usr/share/bash-completion/bash_completion`, which is the main script of bash-completion. Depending on your package manager, you have to manually source this file in your `~/.bashrc` file.

To find out, reload your shell and run `type _init_completion` . If the command succeeds, you're already set, otherwise add the following to your `~/.bashrc` file:

```
source /usr/share/bash-completion/bash_completion
```

Reload your shell and verify that bash-completion is correctly installed by typing `type _init_completion` .

Enable kubectl autocompletion

Bash

You now need to ensure that the kubectl completion script gets sourced in all your shell sessions. There are two ways in which you can do this:

User System

```
echo 'source <(kubectl completion bash)' >>~/.bashrc
```

If you have an alias for kubectl, you can extend shell completion to work with that alias:

```
echo 'alias k=kubectl' >>~/.bashrc
echo 'complete -o default -F __start_kubectl k' >>~/.bashrc
```

Note:
bash-completion sources all completion scripts in `/etc/bash_completion.d`.

Both approaches are equivalent. After reloading your shell, kubectl autocompletion should be working. To enable bash autocompletion in current session of shell, source the ~/.bashrc file:

```
source ~/.bashrc
```

Install `kubectl convert` plugin

A plugin for Kubernetes command-line tool `kubectl` , which allows you to convert manifests between different API versions. This can be particularly helpful to migrate manifests to a non-deprecated api version with newer Kubernetes release. For more info, visit migrate to non deprecated apis

1. Download the latest release with the command:

   x86-64 ARM64

   ```
   curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl-co
   ```

2. Validate the binary (optional)

   Download the kubectl-convert checksum file:

   x86-64 ARM64

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl-co
```

Validate the kubectl-convert binary against the checksum file:

```
echo "$(cat kubectl-convert.sha256) kubectl-convert" | sha256sum --check
```

If valid, the output is:

```
kubectl-convert: OK
```

If the check fails, `sha256` exits with nonzero status and prints output similar to:

```
kubectl-convert: FAILED
sha256sum: WARNING: 1 computed checksum did NOT match
```

Note:
Download the same version of the binary and checksum.

3. Install kubectl-convert

```
sudo install -o root -g root -m 0755 kubectl-convert /usr/local/bin/kubectl-convert
```

4. Verify plugin is successfully installed

```
kubectl convert --help
```

If you do not see an error, it means the plugin is successfully installed.

5. After installing the plugin, clean up the installation files:

```
rm kubectl-convert kubectl-convert.sha256
```

What's next

- Install Minikube
- See the getting started guides for more about creating clusters.
- Learn how to launch and expose your application.
- If you need access to a cluster you didn't create, see the Sharing Cluster Access document.
- Read the kubectl reference docs

Feedback

Was this page helpful?

Yes No

Last modified August 08, 2024 at 4:34 PM PST: Update SUSE install-kubectl-linux.md (0299ca8f34)