Install and Set Up kubectl on Windows

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 Windows

The following methods exist for installing kubectl on Windows:

- Install kubectl binary on Windows (via direct download or curl)
- Install on Windows using Chocolatey, Scoop, or winget

Install kubectl binary on Windows (via direct download or curl)

1. You have two options for installing kubectl on your Windows device

   - Direct download:

     Download the latest 1.31 patch release binary directly for your specific architecture by visiting the Kubernetes release page. Be sure to select the correct binary for your architecture (e.g., amd64, arm64, etc.).

   - Using curl:

     If you have `curl` installed, use this command:

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

     Note:
     To find out the latest stable version (for example, for scripting), take a look at https://dl.k8s.io/release/stable.txt.

2. Validate the binary (optional)

   Download the `kubectl` checksum file:

   ```
   curl.exe -LO "https://dl.k8s.io/v1.31.0/bin/windows/amd64/kubectl.exe.sha256"
   ```

   Validate the `kubectl` binary against the checksum file:

   - Using Command Prompt to manually compare `CertUtil` 's output to the checksum file downloaded:

```
CertUtil -hashfile kubectl.exe SHA256
type kubectl.exe.sha256
```

- Using PowerShell to automate the verification using the `-eq` operator to get a `True` or `False` result:

```
$(Get-FileHash -Algorithm SHA256 .\kubectl.exe).Hash -eq $(Get-Content .\kubectl.exe.sha256)
```

3. Append or prepend the `kubectl` binary folder to your `PATH` environment variable.

4. Test to ensure the version of `kubectl` is the same as downloaded:

```
kubectl version --client
```

Or use this for detailed view of version:

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

Note:
Docker Desktop for Windows adds its own version of `kubectl` to `PATH`. If you have installed Docker Desktop before, you may need to place your `PATH` entry before the one added by the Docker Desktop installer or remove the Docker Desktop's `kubectl`.

Install on Windows using Chocolatey, Scoop, or winget

1. To install kubectl on Windows you can use either Chocolatey package manager, Scoop command-line installer, or winget package manager.

choco | scoop | winget

```
choco install kubernetes-cli
```

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

```
kubectl version --client
```

3. Navigate to your home directory:

```
# If you're using cmd.exe, run: cd %USERPROFILE%
cd ~
```

4. Create the `.kube` directory:

```
mkdir .kube
```

5. Change to the `.kube` directory you just created:

```
cd .kube
```

6. Configure kubectl to use a remote Kubernetes cluster:

```
New-Item config -type file
```

Note:
Edit the config file with a text editor of your choice, such as Notepad.

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 PowerShell.

The kubectl completion script for PowerShell can be generated with the command `kubectl completion powershell`.

To do so in all your shell sessions, add the following line to your `$PROFILE` file:

```
kubectl completion powershell | Out-String | Invoke-Expression
```

This command will regenerate the auto-completion script on every PowerShell start up. You can also add the generated script directly to your `$PROFILE` file.

To add the generated script to your `$PROFILE` file, run the following line in your powershell prompt:

```
kubectl completion powershell >> $PROFILE
```

After reloading your shell, kubectl autocompletion should be working.

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:

```
curl.exe -LO "https://dl.k8s.io/release/v1.31.0/bin/windows/amd64/kubectl-convert.exe"
```

2. Validate the binary (optional).

   Download the `kubectl-convert` checksum file:

```
curl.exe -LO "https://dl.k8s.io/v1.31.0/bin/windows/amd64/kubectl-convert.exe.sha256"
```

   Validate the `kubectl-convert` binary against the checksum file:

   - Using Command Prompt to manually compare `CertUtil` 's output to the checksum file downloaded:

```
CertUtil -hashfile kubectl-convert.exe SHA256
type kubectl-convert.exe.sha256
```

   - Using PowerShell to automate the verification using the `-eq` operator to get a `True` or `False` result:

```
$($(CertUtil -hashfile .\kubectl-convert.exe SHA256)[1] -replace " ", "") -eq $(type .\kubectl-convert.exe.sha256)
```

3. Append or prepend the `kubectl-convert` binary folder to your `PATH` environment variable.

4. Verify the 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:

```
del kubectl-convert.exe
del kubectl-convert.exe.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 25, 2024 at 8:53 AM PST: update kubectl download options (bf96dbe898)