Set up a local Kubernetes cluster Lab Guide using Minikube

Table of Contents

Introduction

This guide outlines the steps required to set up a local Kubernetes cluster on a Windows system using either Minikube or Kind. By following the instructions, you will be able to create a Kubernetes cluster locally to practice deploying, managing, and testing applications.

Problem Statement

Setting up Kubernetes on Windows can be a challenging process due to differences in operating system architecture and compatibility issues with certain tools. This guide simplifies the setup process by offering clear, step-by-step instructions for creating a local Kubernetes cluster using Minikube or Kind, two widely-used solutions for running Kubernetes on a local machine.

Prerequisites

Software Requirements

- **Windows 10 or later**
- **Docker Desktop for Windows** (for Kind)
- **kubectl** (Kubernetes command-line tool)
- **Windows Subsystem for Linux 2 (WSL2)**

Hardware Requirements

- **CPU**: Minimum 2 CPUs
- **Memory**: Minimum 4GB RAM (recommended 8GB or more)
- **Disk Space**: At least 10GB of free space

Setup Instructions

Set Up a Local Kubernetes Cluster Using Minikube.

Step 1: Install Minikube

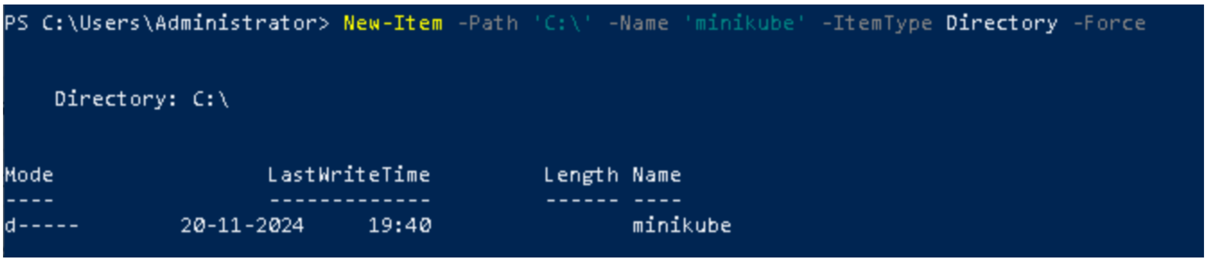
- **Download and Install the Latest Minikube Release**:

  To install the latest stable release of Minikube on x86-64 Windows, follow these steps:

- **Manual Installation**:
  - Visit the Minikube releases page and download the latest installer for Windows.
  - **PowerShell Installation**:

    If you prefer to use PowerShell, run the following commands:

    ```
    New-Item -Path 'C:\' -Name 'minikube' -ItemType Directory -Force
    ```

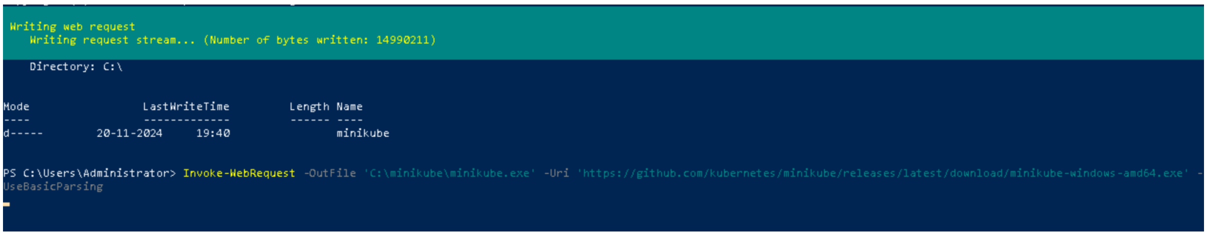
    

    ```
    PS C:\Users\Administrator> New-Item -Path 'C:\' -Name 'minikube' -ItemType Directory -Force


        Directory: C:\


    Mode                 LastWriteTime         Length Name
    ----                 -------------         ------ ----
    d-----        20-11-2024     19:40                minikube
    ```

    ```
    Invoke-WebRequest -OutFile 'C:\minikube\minikube.exe' -Uri
    'https://github.com/kubernetes/minikube/releases/latest/download/miniku
    be-windows-amd64.exe' -UseBasicParsing
    ```

    ```
    Writing web request
       Writing request stream... (Number of bytes written: 14990211)

       Directory: C:\


    Mode                 LastWriteTime         Length Name
    ----                 -------------         ------ ----
    d-----        20-11-2024     19:40                minikube


    PS C:\Users\Administrator> Invoke-WebRequest -OutFile 'C:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/minikube/releases/latest/download/minikube-windows-amd64.exe' -
    UseBasicParsing
    ```

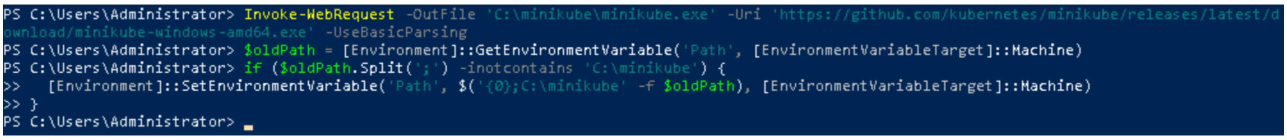
  Make sure to run PowerShell as Administrator to execute these commands.

- **Add Minikube to Your PATH**: To add the Minikube path (`C:/minikube`) to the system and user environment variables, open the Environment Variables settings, edit the "Path" variable in both the System and User sections, and add `C:/minikube`. Save the changes by clicking "OK" in each dialog box.

  After downloading, you need to add the Minikube binary to your system's PATH. Run the following command in PowerShell:

  ```
  $oldPath = [Environment]::GetEnvironmentVariable('Path',
  [EnvironmentVariableTarget]::Machine)
  if ($oldPath.Split(';') -inotcontains 'C:\minikube') {
  ```

```
  [Environment]::SetEnvironmentVariable('Path', $('{0};C:\minikube' -f
$oldPath), [EnvironmentVariableTarget]::Machine)
}
```



```
PS C:\Users\Administrator> Invoke-WebRequest -OutFile 'C:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/minikube/releases/latest/d
ownload/minikube-windows-amd64.exe' -UseBasicParsing
PS C:\Users\Administrator> $oldPath = [Environment]::GetEnvironmentVariable('Path', [EnvironmentVariableTarget]::Machine)
PS C:\Users\Administrator> if ($oldPath.Split(';') -inotcontains 'C:\minikube') {
>>   [Environment]::SetEnvironmentVariable('Path', $('{0};C:\minikube' -f $oldPath), [EnvironmentVariableTarget]::Machine)
>> }
PS C:\Users\Administrator> _
```

- **Restart Your Terminal**:

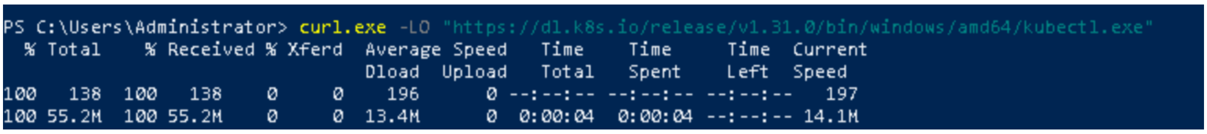
  If you used PowerShell for installation, close the terminal and reopen it before running Minikube commands

Step 2: Install `kubectl`

- **Download kubectl with curl**

  - If you have `curl` installed, you can download `kubectl` directly using the following command:

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

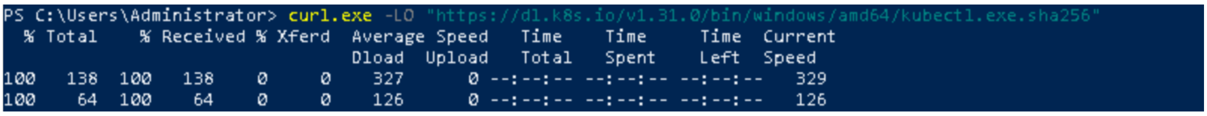
    ```
    PS C:\Users\Administrator> curl.exe -LO "https://dl.k8s.io/release/v1.31.0/bin/windows/amd64/kubectl.exe"
      % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                     Dload  Upload   Total   Spent    Left  Speed
    100   138  100   138    0     0    196      0 --:--:-- --:--:-- --:--:--   197
    100 55.2M  100 55.2M    0     0  13.4M      0  0:00:04  0:00:04 --:--:-- 14.1M
    ```

- **(Optional) Validate the Binary**

  - To ensure the downloaded binary is not corrupted, you can validate it against the checksum file.

  - Download the checksum file:

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

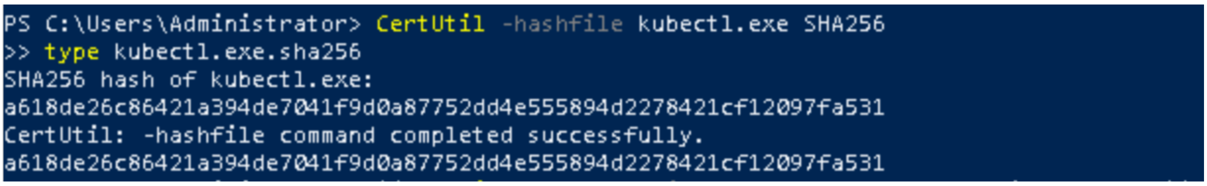
    

    ```
    PS C:\Users\Administrator> curl.exe -LO "https://dl.k8s.io/v1.31.0/bin/windows/amd64/kubectl.exe.sha256"
      % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                     Dload  Upload   Total   Spent    Left  Speed
    100   138  100   138    0     0    327      0 --:--:-- --:--:-- --:--:--   329
    100    64  100    64    0     0    126      0 --:--:-- --:--:-- --:--:--   126
    ```

- **Validate the kubectl Binary**

  - Using Command Prompt, manually compare the SHA256 hash:

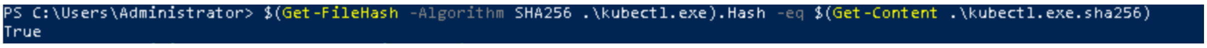
    ```
    CertUtil -hashfile kubectl.exe SHA256
    type kubectl.exe.sha256
    ```



```
PS C:\Users\Administrator> CertUtil -hashfile kubectl.exe SHA256
>> type kubectl.exe.sha256
SHA256 hash of kubectl.exe:
a618de26c86421a394de7041f9d0a87752dd4e555894d2278421cf12097fa531
CertUtil: -hashfile command completed successfully.
a618de26c86421a394de7041f9d0a87752dd4e555894d2278421cf12097fa531
```

- Alternatively, use PowerShell to automate the verification:

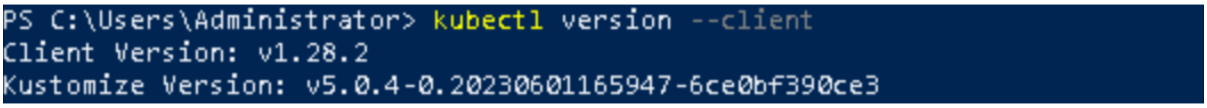
```
$(Get-FileHash -Algorithm SHA256 .\kubectl.exe).Hash -eq $(Get-Content .\kubectl.exe.sha256)
```



```
PS C:\Users\Administrator> $(Get-FileHash -Algorithm SHA256 .\kubectl.exe).Hash -eq $(Get-Content .\kubectl.exe.sha256)
True
```

- **Test kubectl Installation**
  - Again, open a terminal and verify the installation:

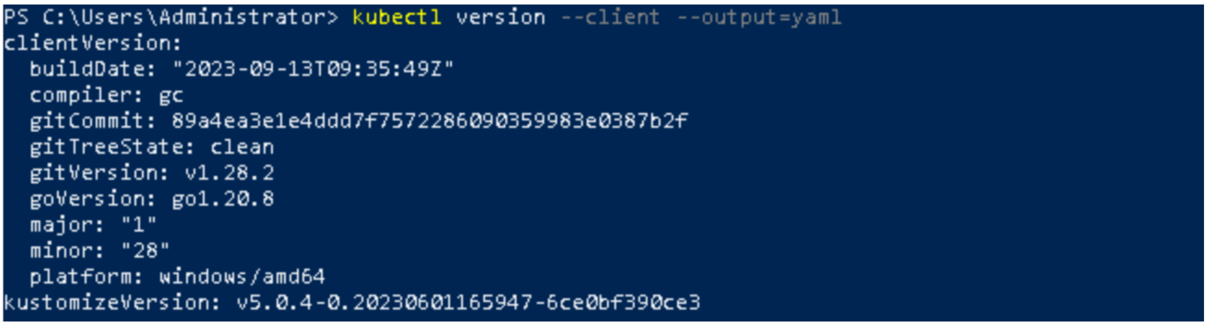
```
kubectl version --client
```



```
PS C:\Users\Administrator> kubectl version --client
Client Version: v1.28.2
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
```

  - For detailed version information, use:

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

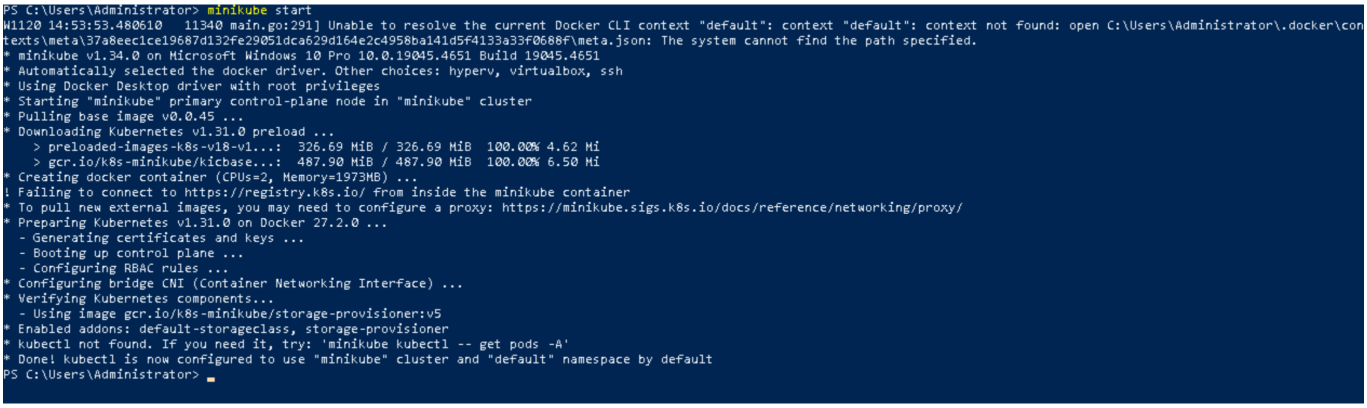


```
PS C:\Users\Administrator> kubectl version --client --output=yaml
clientVersion:
  buildDate: "2023-09-13T09:35:49Z"
  compiler: gc
  gitCommit: 89a4ea3e1e4ddd7f7572286090359983e0387b2f
  gitTreeState: clean
  gitVersion: v1.28.2
  goVersion: go1.20.8
  major: "1"
  minor: "28"
  platform: windows/amd64
kustomizeVersion: v5.0.4-0.20230601165947-6ce0bf390ce3
```

Step 3: Start your cluster:

From a PowerShell with administrator access run(Make sure your `Docker Desktop` is running):

```
minikube start
```



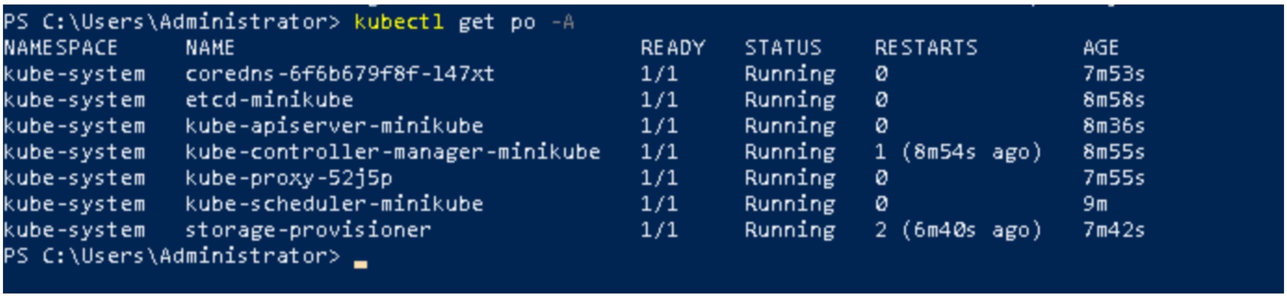
```
PS C:\Users\Administrator> minikube start
W1120 14:53:53.480610   11340 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": context not found: open C:\Users\Administrator\.docker\con
texts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: The system cannot find the path specified.
* minikube v1.34.0 on Microsoft Windows 10 Pro 10.0.19045.4651 Build 19045.4651
* Automatically selected the docker driver. Other choices: hyperv, virtualbox, ssh
* Using Docker Desktop driver with root privileges
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.45 ...
* Downloading Kubernetes v1.31.0 preload ...
    > preloaded-images-k8s-v18-v1...:  326.69 MiB / 326.69 MiB  100.00% 4.62 Mi
    > gcr.io/k8s-minikube/kicbase...:  487.90 MiB / 487.90 MiB  100.00% 6.50 Mi
* Creating docker container (CPUs=2, Memory=1973MB) ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.31.0 on Docker 27.2.0 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: default-storageclass, storage-provisioner
* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Users\Administrator>
```

Step 4: Interact with your cluster

Once you have `kubectl` installed, you can use it to interact with your Kubernetes cluster. To verify that your setup is working and to see the resources in your cluster, use the following command:

```
kubectl get po -A
```

- **Explanation:**
  - `kubectl`: The command-line tool for interacting with Kubernetes.
  - `get`: This command retrieves information about resources in the cluster.
  - `po`: This stands for "pods." Pods are the smallest deployable units in Kubernetes, representing a single instance of a running process in your cluster.
  - `-A`: This flag stands for "all namespaces." It allows you to view pods from all namespaces in your cluster.



```
PS C:\Users\Administrator> kubectl get po -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS        AGE
kube-system   coredns-6f6b679f8f-147xt           1/1     Running   0               7m53s
kube-system   etcd-minikube                      1/1     Running   0               8m58s
kube-system   kube-apiserver-minikube            1/1     Running   0               8m36s
kube-system   kube-controller-manager-minikube   1/1     Running   1 (8m54s ago)   8m55s
kube-system   kube-proxy-52j5p                   1/1     Running   0               7m55s
kube-system   kube-scheduler-minikube            1/1     Running   0               9m
kube-system   storage-provisioner                1/1     Running   2 (6m40s ago)   7m42s
PS C:\Users\Administrator>
```

References

- Minikube Documentation
- Docker Desktop for Windows
