

# Kubernetes Introduction

**Instructor:** Asst. Prof. Dr. Praphan Pavarangkoon

**Office:** Room no. 418-5, 4th Floor

**Email:** [praphan@it.kmitl.ac.th](mailto:praphan@it.kmitl.ac.th)

**Office hours:** Thursday at 9:00 – 11:00 or  
as an advance appointment



# Agenda

- Installation of Kubernetes on Windows Platform
- Exploring KubeCTL



# Installation of Kubernetes on Windows Platform



Installation of Kubernetes on windows is not straight forward, in its simplest form, which we are going to follow, has two steps:

1. Installation of Kubernetes Cluster (Kubernetes Instance)
2. Installation of kubectl (Command Line Interface for interaction with Kubernetes)

# Pre-requisite

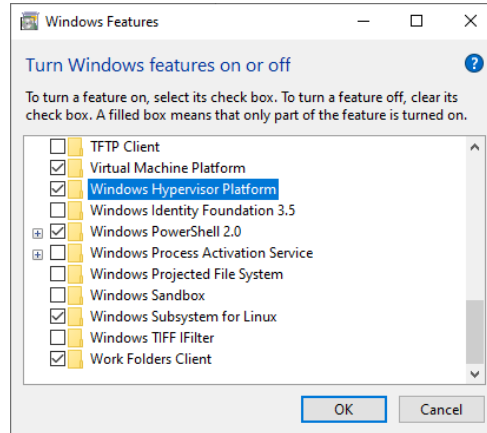


- As Kubernetes from scratch is an application natively designed for the Linux, few setting needs to be confirmed before we proceed with installation.
  1. For **Windows 11**, In the Search bar, search for "apps", and select Apps and features. Select Optional features → Add a feature → scroll down to Windows features.  
or  
For **Windows 10**, You can also press Windows Key + R to open the Run dialog, type “optionalfeatures”, and press Enter

# Pre-requisite (cont.)



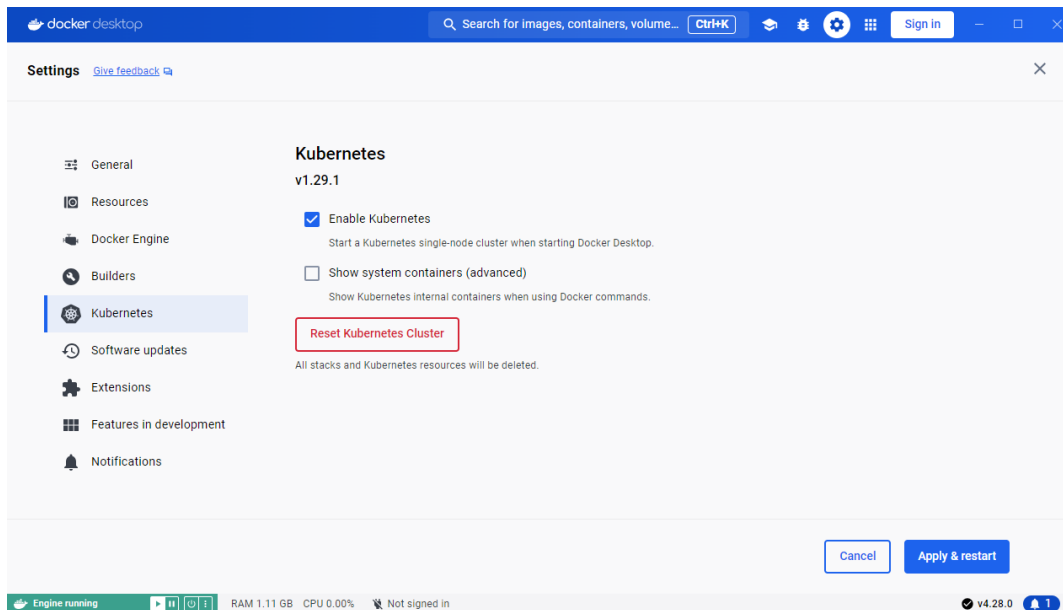
- Select Windows Hypervisor Platform



# Pre-requisite (cont.)



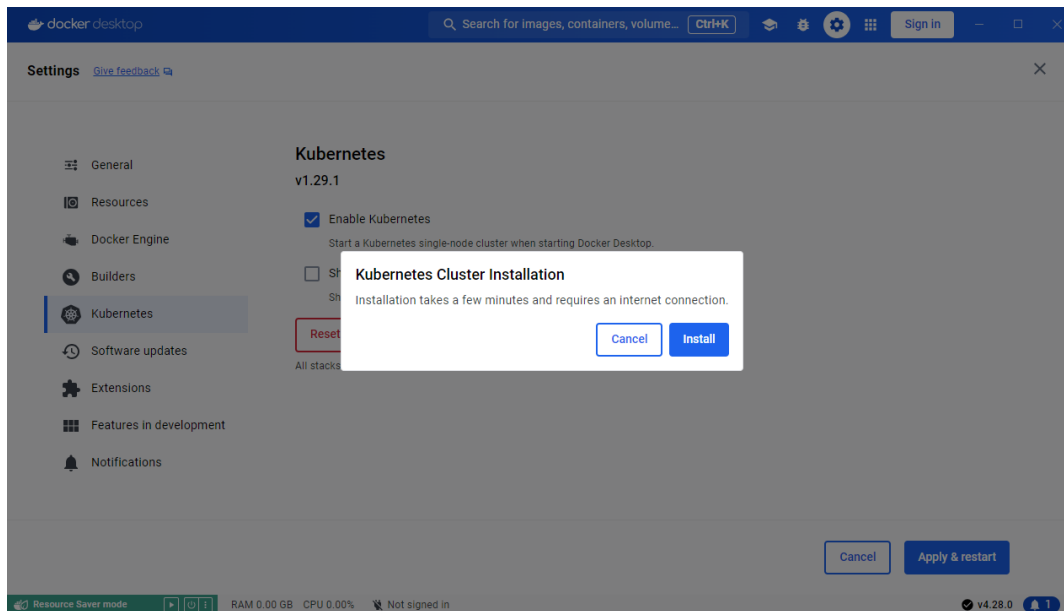
2. Now start the [Docker Desktop](#), and browse to the setting pane of the docker, click on [Kubernetes](#), then check [Enable Kubernetes](#), this will install standalone cluster of Kubernetes.



# Pre-requisite (cont.)



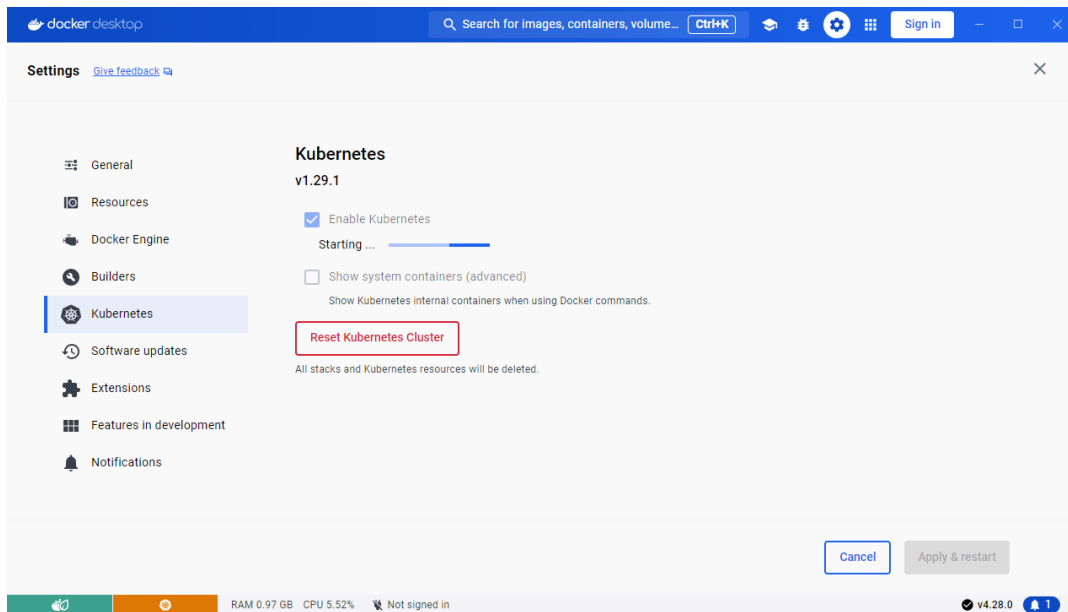
3. Click on **Apply & restart**, this will install additional components required for running Kubernetes.



# Pre-requisite (cont.)



An internet connection is required for downloading and installing required components.

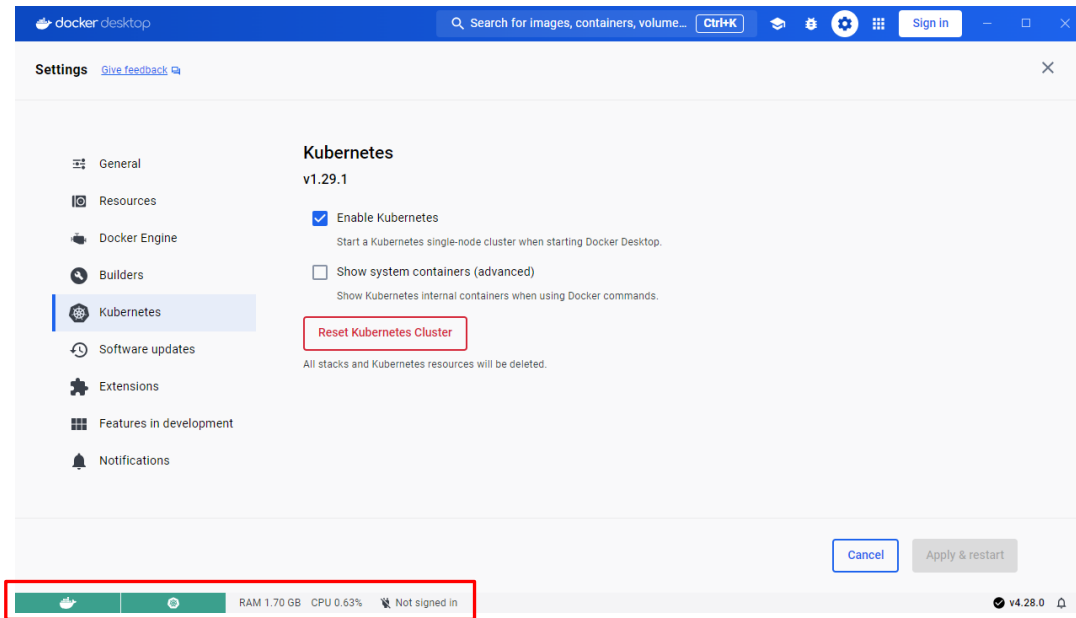




# Pre-requisite (cont.)



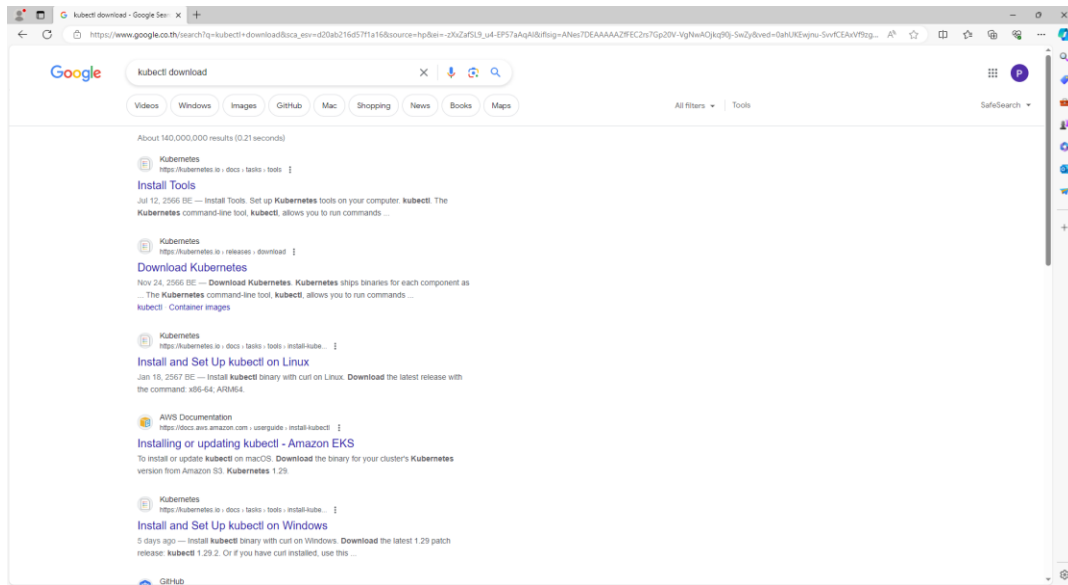
- Once finished, you can see bottom left corner of the Docker desktop interface, Kubernetes icon appears in **green** along with whale icon of the docker, indicating Kubernetes has been installed and running successfully.



# Pre-requisite (cont.)



- Now for installation of **kubectl**, a CLI interface for the Kubernetes, search for “**kubectl**” on the Google. (You do not need to do this step for the latest version of Docker Desktop.)



# Pre-requisite (cont.)



On the page opened, click on “**Install kubectl on Windows**”.

The screenshot shows the Kubernetes documentation website. The top navigation bar includes links for Documentation, Kubernetes Blog, Training, Partners, Community, Case Studies, Versions, and English. A banner for KubeCon + CloudNativeCon Europe 2024 is visible. The left sidebar contains a search bar and a list of navigation items: Documentation, Getting started, Concepts, Tasks, and a sub-menu for Install Tools. The main content area is titled 'Install Tools' and features a section for 'kubectl' with a description and a list of links for installing kubectl on Linux, macOS, and Windows. Below this is a section for 'kind' with a description and a link to the 'kind Quick Start' page. The right sidebar includes links to edit the page, create a child page, create an issue, and print the entire section.

Kubernetes Documentation / Tasks / Install Tools

## Install Tools

### kubectl

The Kubernetes command-line tool, `kubectl`, allows you to run commands against Kubernetes clusters. You can use `kubectl` to deploy applications, inspect and manage cluster resources, and view logs. For more information including a complete list of `kubectl` operations, see the [kubectl reference documentation](#).

`kubectl` is installable on a variety of Linux platforms, macOS and Windows. Find your preferred operating system below.

- [Install kubectl on Linux](#)
- [Install kubectl on macOS](#)
- [Install kubectl on Windows](#)

### kind

[kind](#) lets you run Kubernetes on your local computer. This tool requires that you have either `Docker` or `Podman` installed.

The `kind` [Quick Start](#) page shows you what you need to do to get up and running with `kind`.

[View tool Quick Start Guide](#)

# Pre-requisite (cont.)



On the page opened, click on “**latest ... patch release**”.

Kubernetes Documentation / Tasks / Install Tools / Install and Set Up kubectl on Windows

## 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 v1.29 client can communicate with v1.28, v1.29, and v1.30 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 with curl on Windows
- Install on Windows using Chocolatey, Scoop, or winget

#### Install kubectl binary with curl on Windows

1. Download the latest 1.29 patch release: **kubectl 1.29.2**.

Or if you have curl installed, use this command:

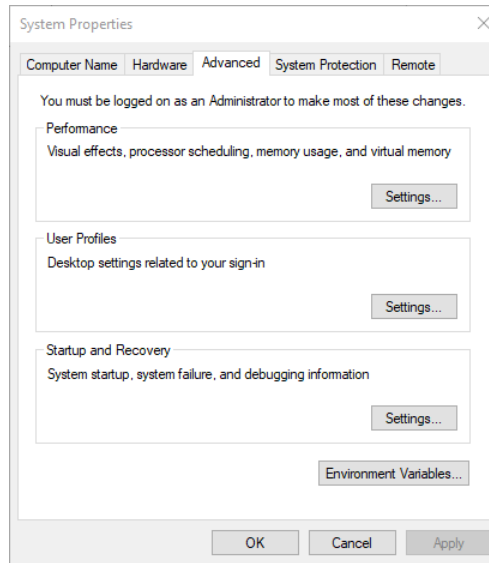
```
curl.exe -LO "https://dl.k8s.io/release/v1.29.2/bin/windows/amd64/kubectl.exe"
```

Save kubectl binary (**kubectl.exe**) somewhere in your system, a preferred location is “**C:\kubectl**”.

# Pre-requisite (cont.)



Now, on search, look for “**Environment**”, Environment Variable link will appear, click and open “**Edit the system environment variables**” setting of the system.

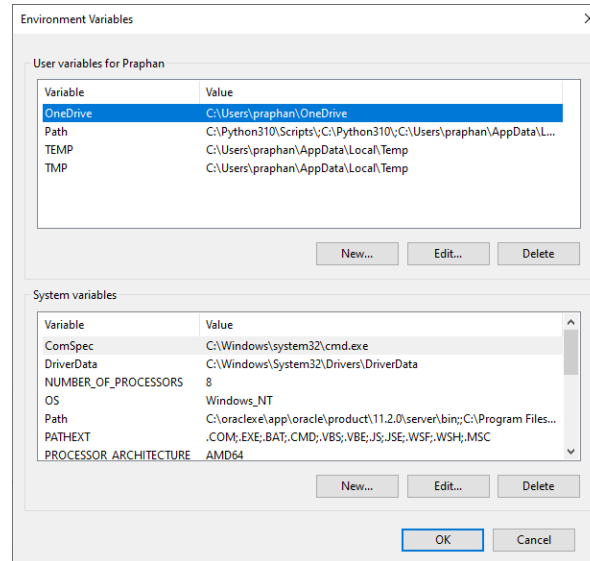


On the setting window opened thus, click on “**Environment Variables...**” at the bottom.

# Pre-requisite (cont.)



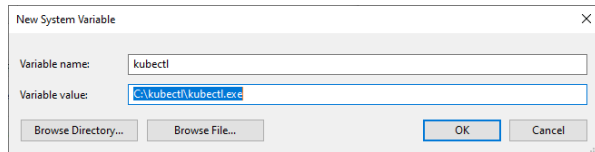
On the subsequent window opened, either select “New...” on **User variables** section (if you want to make changes for current user only) or select “New...” on **System variables** section (this will make changes system wise for all user of the system)



# Pre-requisite (cont.)

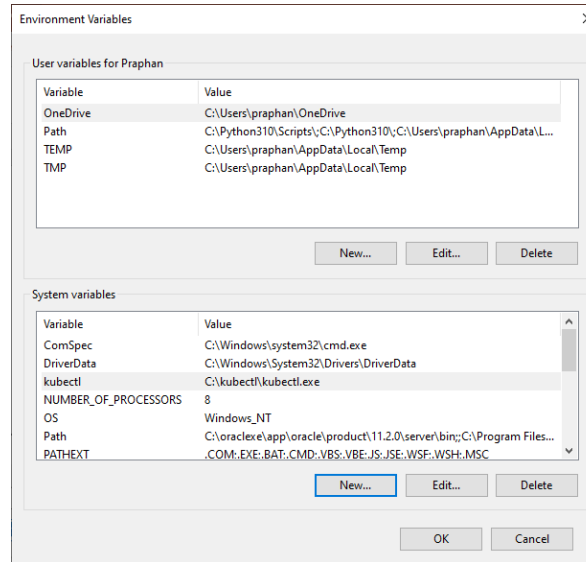


Add variable, write `kubectl` for the variable name and either provide value for this or browse to the location where “`kubectl.exe`” file was saved.



Once done, click on “OK”, a new Environment Variable entry has been made, and it should be visible as well.

# Pre-requisite (cont.)



Click on “**OK**” to finalize everything.



# Pre-requisite (cont.)



6. Open “**Command Prompt**”, and type “**kubectl**”, if it responds with a verbose screen like below, means you have set everything perfectly.

```
Command Prompt
Microsoft Windows [Version 10.0.19045.4046]
(c) Microsoft Corporation. All rights reserved.

C:\Users\praphan>kubectl
kubectl controls the Kubernetes cluster manager.

Find more information at: https://kubernetes.io/docs/reference/kubectl/

Basic Commands (Beginner):
  create      Create a resource from a file or from stdin
  expose      Take a replication controller, service, deployment or pod and expose it as a new
Kubernetes service
  run         Run a particular image on the cluster
  set         Set specific features on objects

Basic Commands (Intermediate):
  explain     Get documentation for a resource
  get         Display one or many resources
  edit        Edit a resource on the server
  delete      Delete resources by file names, stdin, resources and names, or by resources and
label selector

Deploy Commands:
  rollout     Manage the rollout of a resource
  scale       Set a new size for a deployment, replica set, or replication controller
  autoscale   Auto-scale a deployment, replica set, stateful set, or replication controller

Cluster Management Commands:
  certificate Modify certificate resources
  cluster-info Display cluster information
```

# Some Basic Commands (cont.)



7. On Command Prompt window, write “**kubectl config current-context**”.

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The command prompt shows the user's current directory as "C:\Users\praphan". The command entered is "kubectl config current-context", and the output displayed is "docker-desktop". The prompt then returns to "C:\Users\praphan>".

```
Command Prompt
C:\Users\praphan>kubectl config current-context
docker-desktop
C:\Users\praphan>
```

It shows current context of the Kubernetes instance, which in turn reveals that it is going to run on “**docker-desktop**” cluster.

# Some Basic Commands (cont.)



8. To know the information about the nodes available on current cluster, run this command “**kubectl get nodes**”.

A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The command prompt shows the user at the C:\Users\praphan directory. The command "kubectl get nodes" has been entered and executed. The output is a table with five columns: NAME, STATUS, ROLES, AGE, and VERSION. There is one row of data: "docker-desktop" is the name, "Ready" is the status, "control-plane" is the role, "78m" is the age, and "v1.29.1" is the version. The prompt "C:\Users\praphan>" is shown below the output.

```
Command Prompt
C:\Users\praphan>kubectl get nodes
NAME          STATUS    ROLES    AGE   VERSION
docker-desktop Ready    control-plane  78m   v1.29.1
C:\Users\praphan>
```

It shows that only one machine is running.

# Some Basic Commands (cont.)



9. To know about pods available on the current cluster, use this command “**kubectl get pods**”.

A screenshot of a Windows Command Prompt window titled "Administrator: Command Prompt". The window shows the command `C:\Windows\system32>kubectl get pods` being entered, followed by the output `No resources found in default namespace.` and the prompt `C:\Windows\system32>` on the next line. The window has a black background and white text.

It shows that currently no pods are available.

# Some Basic Commands (cont.)



10. To know about all the details of the Kubernetes cluster, use this command  
“**kubectl version --output=yaml**”.

```
Command Prompt
C:\Users\praphan>kubectl version --output=yaml
clientVersion:
  buildDate: "2024-01-17T15:51:03Z"
  compiler: gc
  gitCommit: bc401b91f2782410b3fb3f9acf43a995c4de90d2
  gitTreeState: clean
  gitVersion: v1.29.1
  goVersion: go1.21.6
  major: "1"
  minor: "29"
  platform: windows/amd64
kustomizeVersion: v5.0.4-0.20230601165947-6ce0bf390ce3
serverVersion:
  buildDate: "2024-01-17T15:41:12Z"
  compiler: gc
  gitCommit: bc401b91f2782410b3fb3f9acf43a995c4de90d2
  gitTreeState: clean
  gitVersion: v1.29.1
  goVersion: go1.21.6
  major: "1"
  minor: "29"
  platform: linux/amd64

C:\Users\praphan>
```

# Exploring KubeCTL



- [Minikube](#) is library which lets one use Kubernetes on one's computer without need of installing other things for using clusters and associated maneuvering.
- In a nutshell, [Minikube](#) is a one node Kubernetes cluster, which runs on your laptop.
- In its simplest form, we will first install “[Chocolatey](#)” Installer, and then with the help of this we will install [Minikube](#).

# Installation of Chocolatey



- Open “Windows PowerShell” in Administrator mode, and then run this script on the powershell, wait a few minutes for windows to complete the installation of the script.  
“Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))”

# Installation of Chocolatey (cont.)



- Now paste the above script on the PowerShell prompt

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol
= [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('http
s://community.chocolatey.org/install.ps1'))
Forcing web requests to allow TLS v1.2 (Required for requests to Chocolatey.org)
Getting latest version of the Chocolatey package for download.
Not using proxy.
Getting Chocolatey from https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2.
Downloading https://community.chocolatey.org/api/v2/package/chocolatey/2.2.2 to C:\Users\praphan\AppData\Local\Temp\chocolat
ey\chocoInstall\chocolatey.zip
Not using proxy.
```



# Installation of Chocolatey (cont.)



When installation finishes, run “choco” to check if everything worked properly

A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The window has a dark blue background. The command prompt shows the following text:

```
PS C:\Windows\system32> choco
Chocolatey v2.2.2
Please run 'choco -?' or 'choco <command> -?' for help menu.
PS C:\Windows\system32>
```

Obviously Chocolatey has been installed successfully on the system.

# Installation of Minikube



- On the PowerShell terminal (opened as Administrator) run following command to install **Minikube** on the system.

“**choco install minikube**”

A screenshot of a Windows PowerShell terminal window titled "Administrator: Windows PowerShell". The terminal has a dark blue background with white text. The command "choco install minikube" has been entered and executed. The output shows that Chocolatey v2.2.2 is being used to install minikube. It lists the packages to be installed and shows the progress of downloading the Kubernetes CLI, which is at 61% completion.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> choco install minikube
Chocolatey v2.2.2
Installing the following packages:
minikube
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-cli 1.29.1... 61%
```

# Installation of Minikube (cont.)



When asked give permission as “A”

```
Administrator: Windows PowerShell
PS C:\Windows\system32> choco install minikube
Chocolatey v2.2.2
Installing the following packages:
minikube
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-cli 1.29.1... 100%

kubernetes-cli v1.29.1 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.
The package kubernetes-cli wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): A
```

# Installation of Minikube (cont.)



When installation has been completed successfully, above screen will appear.

```
Administrator: Windows PowerShell
Installing the following packages:
minikube
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-cli 1.29.1... 100%

kubernetes-cli v1.29.1 [Approved]
kubernetes-cli package files install completed. Performing other installation steps.
The package kubernetes-cli wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): A

Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar.gz to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
Extracting 64-bit C:\ProgramData\chocolatey\lib\kubernetes-cli\tools\kubernetes-client-windows-amd64.tar to C:\ProgramData\chocolatey\lib\kubernetes-cli\tools...
C:\ProgramData\chocolatey\lib\kubernetes-cli\tools
ShimGen has successfully created a shim for kubectl-convert.exe
ShimGen has successfully created a shim for kubectl.exe
The install of kubernetes-cli was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\kubernetes-cli\tools'
Progress: Downloading Minikube 1.32.0... 100%

Minikube v1.32.0 [Approved]
Minikube package files install completed. Performing other installation steps.
ShimGen has successfully created a shim for minikube.exe
The install of Minikube was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\Minikube'

Chocolatey installed 2/2 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Windows\system32>
```

# Installation of Minikube (cont.)



A verbose screen is evidence of successful installation of [Minikube](#).  
You have installed a usecase cluster on Kubernetes of your system.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube
WD0313 14:08:35.732533 15780 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
minikube provisions and manages local Kubernetes clusters optimized for development workflows.

Basic Commands:
  start           Starts a local Kubernetes cluster
  status          Gets the status of a local Kubernetes cluster
  stop            Stops a running local Kubernetes cluster
  delete          Deletes a local Kubernetes cluster
  dashboard       Access the Kubernetes dashboard running within the minikube cluster
  pause          pause Kubernetes
  unpause         unpause Kubernetes

Images Commands:
  docker-env      Provides instructions to point your terminal's docker-cli to the Docker Engine inside minikube.
                  (Useful for building docker images directly inside minikube)
  podman-env       Configure environment to use minikube's Podman service
  cache           Manage cache for images
  image           Manage images

Configuration and Management Commands:
  addons          Enable or disable a minikube addon
  config          Modify persistent configuration values
  profile         Get or list the current profiles (clusters)
  update-context  Update kubeconfig in case of an IP or port change

Networking and Connectivity Commands:
  service         Returns a URL to connect to a service
  tunnel          Connect to LoadBalancer services

Advanced Commands:
  mount           Mounts the specified directory into minikube
```

# Exploring KubeCTL (cont.)



- To start cluster, run “**minikube start**”

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube start
W0313 14:11:58.081964 11668 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\pragphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
* minikube v1.32.0 on Microsoft Windows 10 Pro 10.0.19045.4046 Build 19045.4046
* Automatically selected the docker driver. Other choices: hyperv, ssh
* Using Docker Desktop driver with root privileges
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.28.3 preload ...
  > gcr.io/k8s-minikube/kicbase...: 1.91 MiB / 453.90 MiB  0.42% 211.47 KiB
```

Minikube will start downloading required packages and libraries, depending upon network speed, it may take a few to several minutes.

# Exploring KubeCTL (cont.)



Once download is complete, cluster preparation will begin.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube start
W0313 14:11:58.081964 11668 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eece19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
* minikube v1.32.0 on Microsoft Windows 10 Pro 10.0.19045.4046 Build 19045.4046
* Automatically selected the docker driver. Other choices: hyperv, ssh
* Using Docker Desktop driver with root privileges
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.28.3 preload ...
  > preloaded-images-k8s-v18-v1...: 403.35 MiB / 403.35 MiB 100.00% 18.54 M
  > gcr.io/k8s-minikube/kicbase...: 453.90 MiB / 453.90 MiB 100.00% 14.69 M
* Creating docker container (CPUs=2, Memory=1967MB) ...
* Preparing Kubernetes v1.28.3 on Docker 24.0.7 .../■
```

# Exploring KubeCTL (cont.)



Once done, following message will appear

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube start
W0313 14:11:58.081964 11668 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
* minikube v1.32.0 on Microsoft Windows 10 Pro 10.0.19045.4046 Build 19045.4046
* Automatically selected the docker driver. Other choices: hyperv, ssh
* Using Docker Desktop driver with root privileges
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.28.3 preload ...
  > preloaded-images-k8s-v18-v1...: 403.35 MiB / 403.35 MiB 100.00% 18.54 M
  > gcr.io/k8s-minikube/kicbase...: 453.90 MiB / 453.90 MiB 100.00% 14.69 M
* Creating docker container (CPUs=2, Memory=1967MB) ...
* Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Configuring bridge CNI (Container Networking Interface) ...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Verifying Kubernetes components...
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\Windows\system32>
```



# Exploring KubeCTL (cont.)



- Now check the number of nodes in the Kubernetes, with “**kubectl get node**”

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl get node
NAME      STATUS    ROLES    AGE   VERSION
minikube   Ready     control-plane  2m42s  v1.28.3
PS C:\Windows\system32>
```

# Exploring KubeCTL (cont.)



- Check for pods, with “**kubectl get pods**”

A screenshot of a Windows PowerShell terminal window titled "Administrator: Windows PowerShell". The terminal has a dark blue background. The command prompt shows the user at "PS C:\Windows\system32>". The user enters the command `kubectl get pods`. The output displayed is "No resources found in default namespace." followed by another prompt "PS C:\Windows\system32>".

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl get pods
No resources found in default namespace.
PS C:\Windows\system32>
```

which means there are no active pods in the deployment.

# Exploring KubeCTL (cont.)



- **Minikube** is bundled with a dashboard for further investigation and accurate information about cluster, let's launch this dashboard.  
“minikube dashboard”

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube dashboard
WD313 14:20:53.997844 8064 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8ee1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
* Enabling dashboard ...
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
* Some dashboard features require the metrics-server addon. To enable all features please run:

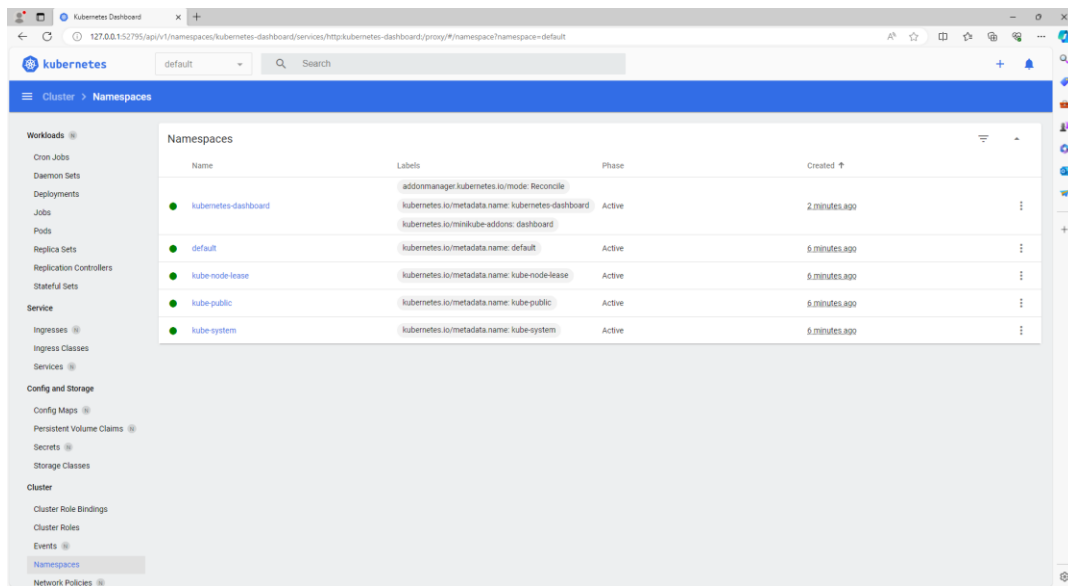
    minikube addons enable metrics-server

* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:52795/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your d
efault browser...
```

# Exploring KubeCTL (cont.)



- A default web browser window will open, showing Kubernetes dashboard with various information and insight into the cluster, click on “Namespaces” on the left pane, all the information about namespaces will be displayed.



# Exploring KubeCTL (cont.)



- Let's create a sample application, and deploy it on this cluster, and expose this to port 80
  - `kubectl create deployment hello-minikube --image=docker.io/nginx:1.23`
  - `kubectl expose deployment hello-minikube --type=NodePort --port=80`

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl create deployment hello-minikube --image=docker.io/nginx:1.23
deployment.apps/hello-minikube created
PS C:\Windows\system32> kubectl expose deployment hello-minikube --type=NodePort --port=80
service/hello-minikube exposed
PS C:\Windows\system32>
```

# Exploring KubeCTL (cont.)



- Let's check the cluster again for the deployment, we have just created with following command, “**kubectl get services hello-minikube**”

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl get services hello-minikube
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
hello-minikube NodePort    10.97.137.82  <none>        80:30823/TCP     43s
PS C:\Windows\system32>
```

We can see that Kubernetes has assigned an internal IP to our cluster and port mapping is also there.

# Exploring KubeCTL (cont.)



- Let's launch this service and see the output of this service, run this command on the PowerShell, “**minikube service hello-minikube**”

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube service hello-minikube
W0313 14:27:46.518255 2636 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.

NAMESPACE   NAME           TARGET PORT   URL
-----
default     hello-minikube 80            http://192.168.49.2:30823

* Starting tunnel for service hello-minikube.

NAMESPACE   NAME           TARGET PORT   URL
-----
default     hello-minikube          http://127.0.0.1:53066

* Opening service default/hello-minikube in default browser...
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

A mapping table is displayed on the console screen showing namespace, name target port and URL of the service.

# Exploring KubeCTL (cont.)



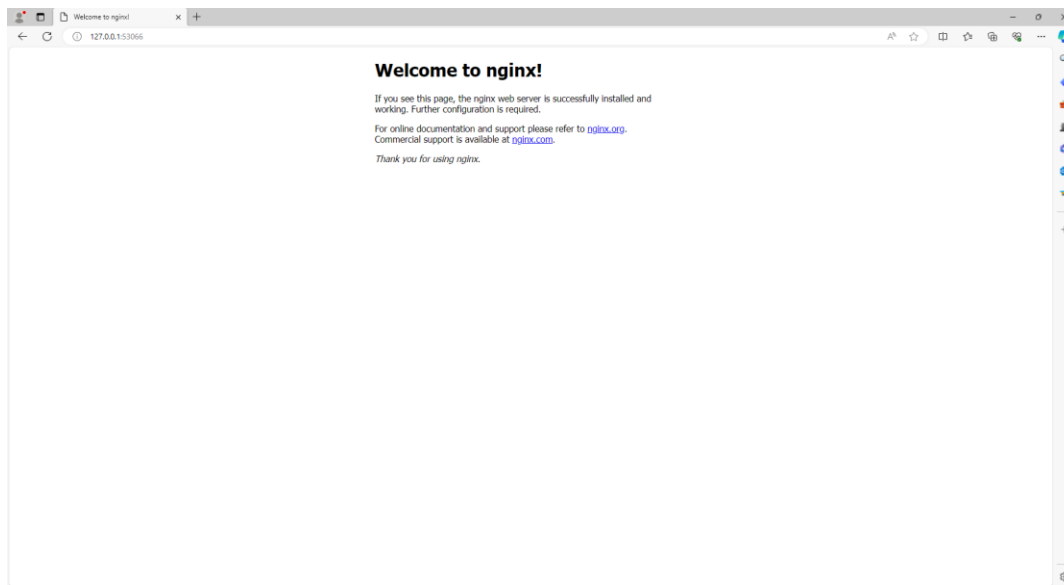
- It is to note that, once run this command, we can't use the PowerShell terminal further, we need to open another window or terminate this command.
- To terminate the current ongoing job, press “**Ctrl + C**”



# Exploring KubeCTL (cont.)



- Because of previous command a new default web browser window will open, showing home page for nginx server



# LoadBalancer Deployment Commands



- To start and use LoadBalancer deployment, use the “**minikube tunnel**” command.  
“**kubectl create deployment balanced --image=docker.io/nginx:1.23**”  
“**kubectl expose deployment balanced --type=LoadBalancer --port=80**”

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl create deployment balanced --image=docker.io/nginx:1.23
deployment.apps/balanced created
PS C:\Windows\system32> kubectl expose deployment balanced --type=LoadBalancer --port=80
service/balanced exposed
PS C:\Windows\system32>
```

# LoadBalancer Deployment Commands (cont.)



- Now to create a routable IP for balanced deployment, start tunnel command in **another window**.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube tunnel
W0313 14:32:09.533485 9496 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8ee1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
* Tunnel successfully started

* NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://mi
nikube.sigs.k8s.io/docs/handbook/accessing/#access-to-ports-1024-on-windows-requires-root-permission
* Starting tunnel for service balanced.
```

This window should remain open, in order to tunnel be available.

# LoadBalancer Deployment Commands (cont.)



- To know the external IP, run “**kubectl get services balanced**”

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl get services balanced
NAME      TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
balanced  LoadBalancer 10.105.125.47  127.0.0.1      80:32201/TCP     3m
PS C:\Windows\system32>
```

Deployment can be accessed with `http://<External IP>:80`

# Some Administrative Commands



- To know about the pods in the cluster, run this command “**kubectl get pods**”.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
balanced-9b9fb74d-d78jb             1/1     Running   0           4m37s
hello-minikube-54b5567fd9-4dcww     1/1     Running   0           9m4s
PS C:\Windows\system32>
```

# Some Administrative Commands (cont.)



- To know about the nodes in the cluster, run this command  
“**kubectl get nodes**”.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl get nodes
NAME      STATUS    ROLES    AGE   VERSION
minikube   Ready     control-plane  18m   v1.28.3
PS C:\Windows\system32>
```

# Some Administrative Commands (cont.)



- To know about the namespaces, run this command  
“**kubectl get namespaces**”.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl get namespaces
NAME                STATUS AGE
default             Active 20m
kube-node-lease      Active 20m
kube-public          Active 20m
kube-system          Active 20m
kubernetes-dashboard Active 15m
PS C:\Windows\system32>
```

# Some Administrative Commands (cont.)



- To know about the deployments in the cluster, run this command “**kubectl get deployments**”.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
balanced      1/1     1            1           7m23s
hello-minikube 1/1     1            1           11m
PS C:\Windows\system32>
```



# Some Administrative Commands (cont.)



- To know about the services in the cluster, run this command “**kubectl get services**”.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl get services
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
balanced      LoadBalancer 10.105.125.47  127.0.0.1      80:32201/TCP     7m45s
hello-minikube NodePort       10.97.137.82   <none>         80:30823/TCP     12m
kubernetes     ClusterIP      10.96.0.1     <none>         443/TCP          21m
PS C:\Windows\system32>
```

# Some Administrative Commands (cont.)



- To know about the cluster, run this command “**kubectl cluster-info**”.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:52686
CoreDNS is running at https://127.0.0.1:52686/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
PS C:\Windows\system32>
```

# Some Administrative Commands (cont.)



- To know about the status of the cluster, run this command “**minikube status**”.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube status
WD313 14:39:31.316916 4272 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
PS C:\Windows\system32>
```

# Some Administrative Commands (cont.)



- To learn about configuration detail, run this command  
“**kubectl config**”

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl config
Modify kubeconfig files using subcommands like "kubectl config set current-context my-context".

The loading order follows these rules:

1. If the --kubeconfig flag is set, then only that file is loaded. The flag may only be set once and no merging takes place.
2. If $KUBECONFIG environment variable is set, then it is used as a list of paths (normal path delimiting rules for your system). These paths are merged. When a value is modified, it is modified in the file that defines the stanza. When a value is created, it is created in the first file that exists. If no files in the chain exist, then it creates the last file in the list.
3. Otherwise, $HOME/.kube/config is used and no merging takes place.

Available Commands:
  current-context  Display the current-context
  delete-cluster   Delete the specified cluster from the kubeconfig
  delete-context   Delete the specified context from the kubeconfig
  delete-user      Delete the specified user from the kubeconfig
  get-clusters     Display clusters defined in the kubeconfig
  get-contexts     Describe one or many contexts
  get-users        Display users defined in the kubeconfig
  rename-context   Rename a context from the kubeconfig file
  set              Set an individual value in a kubeconfig file
  set-cluster      Set a cluster entry in kubeconfig
  set-context      Set a context entry in kubeconfig
  set-credentials  Set a user entry in kubeconfig
  unset            Unset an individual value in a kubeconfig file
  use-context      Set the current-context in a kubeconfig file
  view            Display merged kubeconfig settings or a specified kubeconfig file

Usage:
  kubectl config SUBCOMMAND [options]

Use "kubectl config <command> --help" for more information about a given command.
```

# Some Administrative Commands (cont.)



- To pause the cluster, run “**minikube pause**”, followed by “**minikube status**”.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube pause
WD0313 14:41:06.937733 3392 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
* Pausing node minikube ...
* Paused 18 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
PS C:\Windows\system32> minikube status
WD0313 14:41:57.311856 5680 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
minikube
type: Control Plane
host: Running
kubelet: Stopped
apiserver: Paused
kubeconfig: Configured
PS C:\Windows\system32>
```

# Some Administrative Commands (cont.)



- To resume the cluster, run “**minikube unpause**”, followed by “**minikube status**”

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube unpause
W0313 14:42:52.840921 16840 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
* Unpausing node minikube ...
* Unpaused 18 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
PS C:\Windows\system32> minikube status
W0313 14:43:13.364974 16328 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
PS C:\Windows\system32>
```

# Some Administrative Commands (cont.)



- To stop the cluster, run “**minikube stop**”, followed by “**minikube status**”.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> minikube stop
W0313 14:43:59.415587 17768 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
* Stopping node "minikube" ...
* Powering off "minikube" via SSH ...
* 1 node stopped.
PS C:\Windows\system32> minikube status
W0313 14:44:21.107714 3096 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": co
ntext not found: open C:\Users\praphan\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688
f\meta.json: The system cannot find the path specified.
minikube
type: Control Plane
host: Stopped
kubelet: Stopped
apiserver: Stopped
kubeconfig: Stopped
PS C:\Windows\system32>
```

# Some Administrative Commands (cont.)



- To list the event log, run this command “**kubectl get events**”.

```
Administrator: Windows PowerShell
PS C:\Windows\system32> kubectl get events
```

LAST SEEN	TYPE	REASON	OBJECT	MESSAGE
17m	Normal	Scheduled	pod/balanced-9b9fb74d-d78jb	Successfully assigned default/balanced-9b9fb74d-d78jb to minikube
16m	Normal	Pulled	pod/balanced-9b9fb74d-d78jb	Container image "docker.io/nginx:1.23" already present on machine
16m	Normal	Created	pod/balanced-9b9fb74d-d78jb	Created container nginx
16m	Normal	Started	pod/balanced-9b9fb74d-d78jb	Started container nginx
4m13s	Warning	NodeNotReady	pod/balanced-9b9fb74d-d78jb	Node is not ready
25s	Normal	SandboxChanged	pod/balanced-9b9fb74d-d78jb	Pod sandbox changed, it will be killed and re-created.
19s	Normal	Pulled	pod/balanced-9b9fb74d-d78jb	Container image "docker.io/nginx:1.23" already present on machine
18s	Normal	Created	pod/balanced-9b9fb74d-d78jb	Created container nginx
18s	Normal	Started	pod/balanced-9b9fb74d-d78jb	Started container nginx
17m	Normal	SuccessfulCreate	replicaset/balanced-9b9fb74d-d78jb	Created pod: balanced-9b9fb74d-d78jb
17m	Normal	ScalingReplicaSet	deployment/balanced	Scaled up replica set balanced-9b9fb74d-d78jb to 1
21m	Normal	Scheduled	pod/hello-minikube-54b5567fd9-4dcww	Successfully assigned default/hello-minikube-54b5567fd9-4dcww to minikube
21m	Normal	Pulling	pod/hello-minikube-54b5567fd9-4dcww	Pulling image "docker.io/nginx:1.23"
21m	Normal	Pulled	pod/hello-minikube-54b5567fd9-4dcww	Successfully pulled image "docker.io/nginx:1.23" in 17.659s (17.659s including waiting)
21m	Normal	Created	pod/hello-minikube-54b5567fd9-4dcww	Created container nginx
21m	Normal	Started	pod/hello-minikube-54b5567fd9-4dcww	Started container nginx
4m13s	Warning	NodeNotReady	pod/hello-minikube-54b5567fd9-4dcww	Node is not ready
25s	Normal	SandboxChanged	pod/hello-minikube-54b5567fd9-4dcww	Pod sandbox changed, it will be killed and re-created.
19s	Normal	Pulled	pod/hello-minikube-54b5567fd9-4dcww	Container image "docker.io/nginx:1.23" already present on machine
18s	Normal	Created	pod/hello-minikube-54b5567fd9-4dcww	Created container nginx
18s	Normal	Started	pod/hello-minikube-54b5567fd9-4dcww	Started container nginx
21m	Normal	SuccessfulCreate	replicaset/hello-minikube-54b5567fd9-4dcww	Created pod: hello-minikube-54b5567fd9-4dcww



# Some Administrative Commands (cont.)



- To update the active context, run this command  
“**docker context use default**”.

A screenshot of an Administrator Windows PowerShell window. The window has a title bar that says "Administrator: Windows PowerShell". The command prompt shows the following text:

```
PS C:\Windows\system32> docker context use default
default
Current context is now "default"
PS C:\Windows\system32>
```

# Uninstallation of Minikube



# PowerShell elevated privileges

```
minikube stop
```

```
minikube delete
```

# Delete the .minikube and .kube directories under user

```
C:\users\{user}\.minikube (rm $HOME/.minikube)
```

```
C:\users\{user}\.kube (rm $HOME/.kube)
```

# If one has installed it via Chocolatey

```
minikube stop
```

```
minikube delete
```

```
choco uninstall minikube
```

```
choco uninstall kubectl
```

# Q & A

