

# SE489 DevOps Engineering

## Lab 8



## Lab 8: Installation of Kubernetes on Windows Platform

### Objectives:

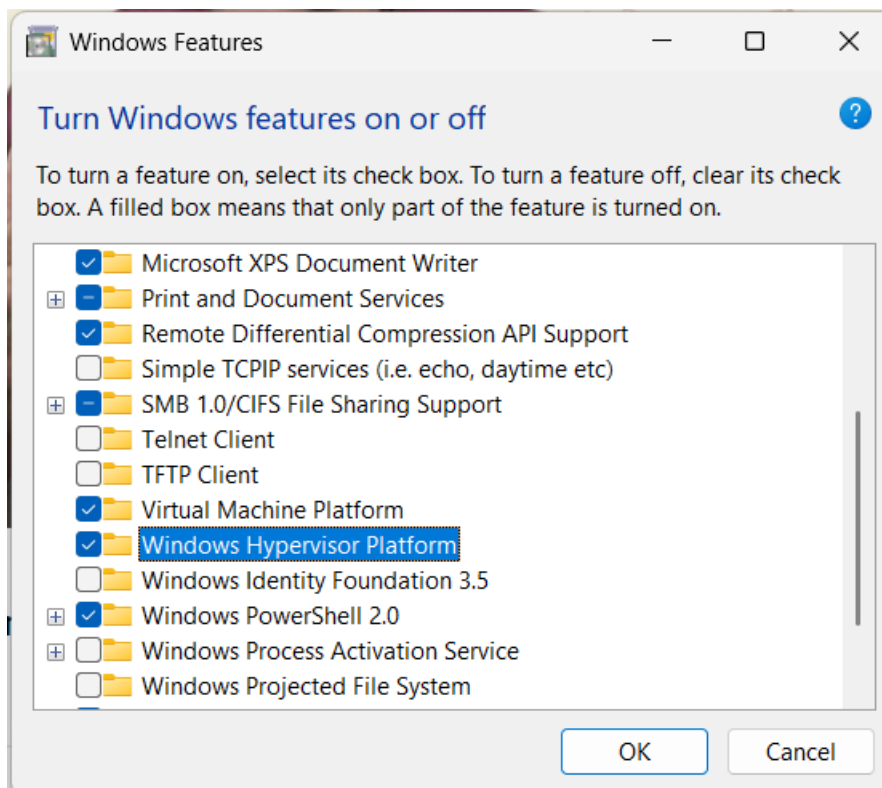
On successful completion of this lab, session students will learn how to download, install, configure and run Kubernetes on windows platform.

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

- a. Installation of Kubernetes Cluster (Kubernetes Instance)
- b. Installation of **kubectrl** (Command Line Interface for interaction with Kubernetes)

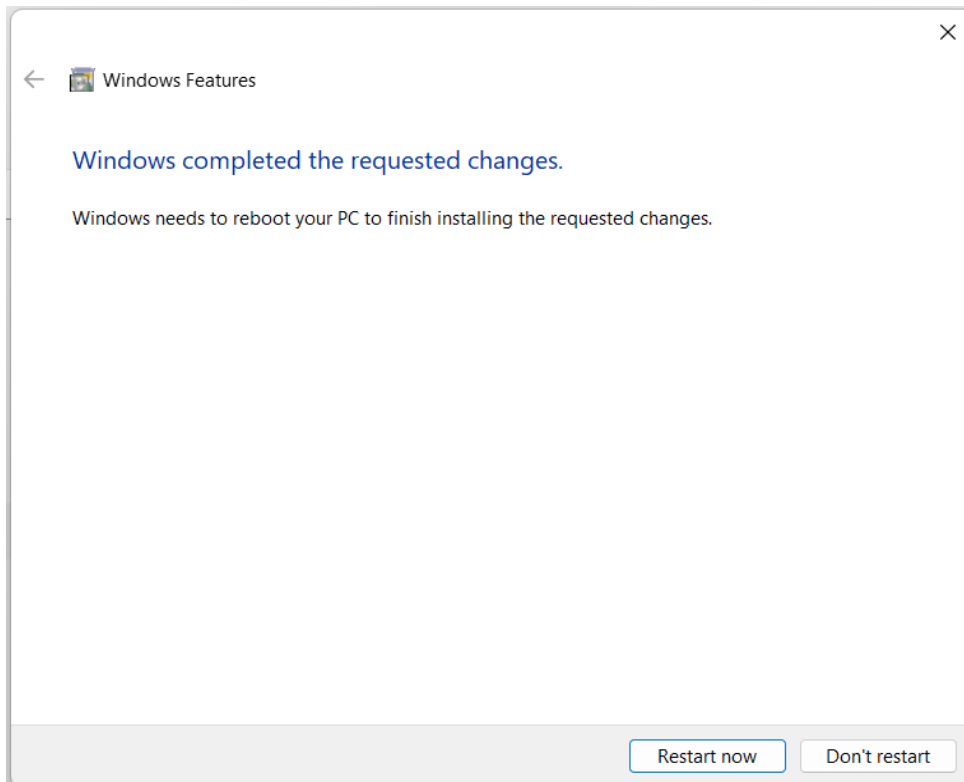
**Pre-requisite:** As Kubernetes from scratch is an application natively designed for the Linux, few setting needs to be confirmed before we proceed with installation.

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

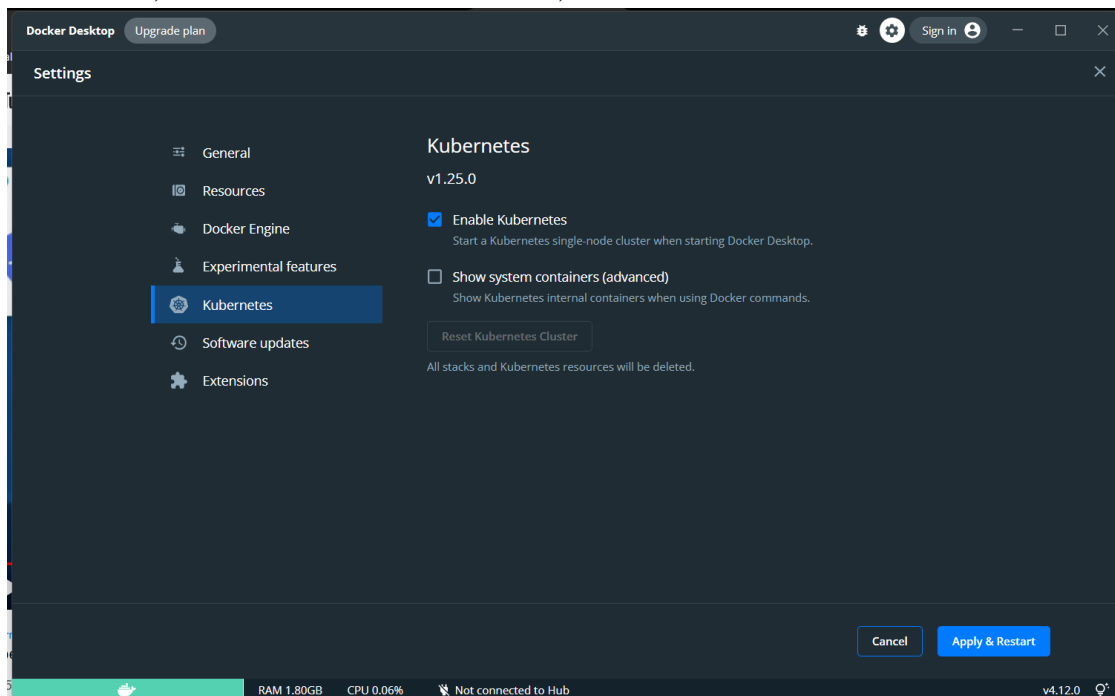


Select Windows Hypervisor Platform

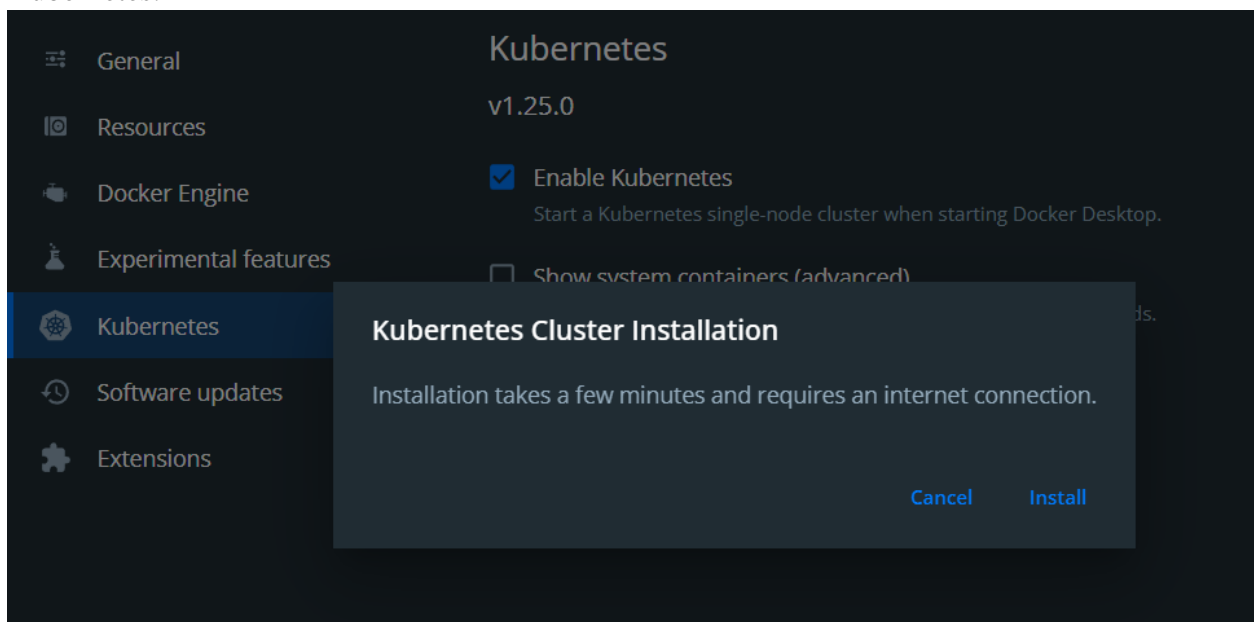
After completion of installation, following screen will appear



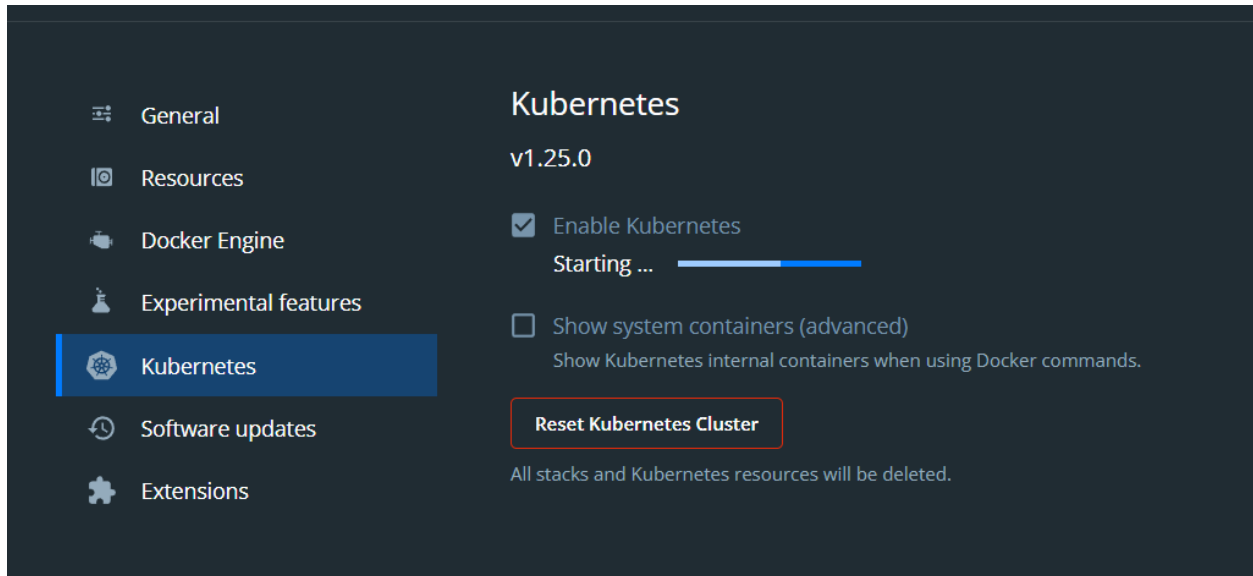
2. Now start the Docker desktop, and browse to the setting pane of the docker, click on Kubernetes, then check Enable Kubernetes, this will install standalone cluster of Kubernetes.



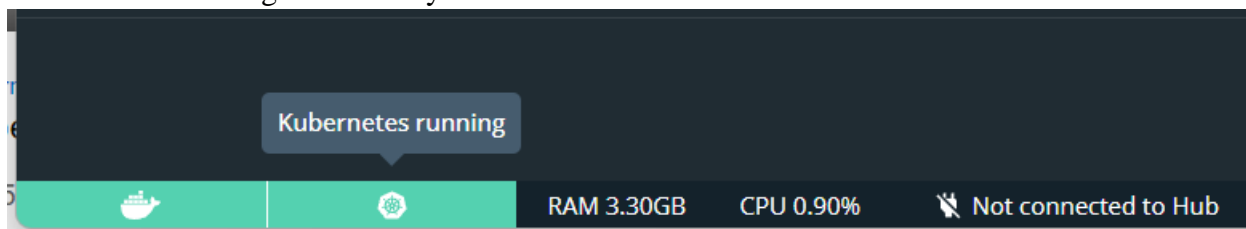
3. Click on apply and restart, this will install additional components required for running Kubernetes.



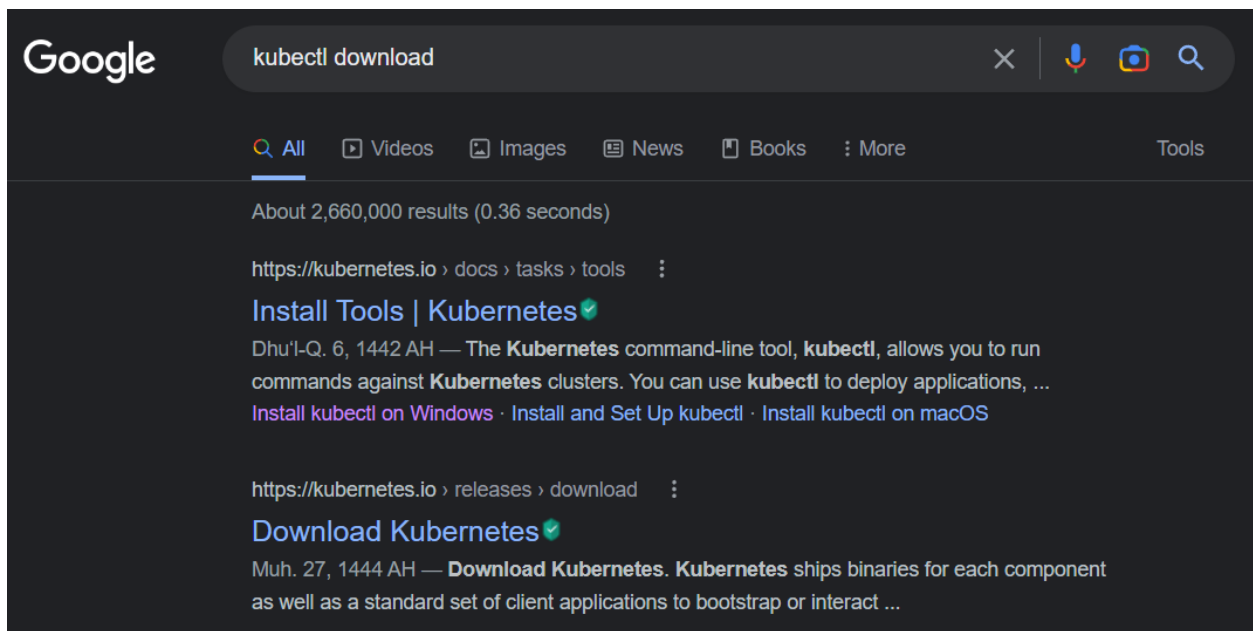
An internet connection is required for downloading and installing required components.



- 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.

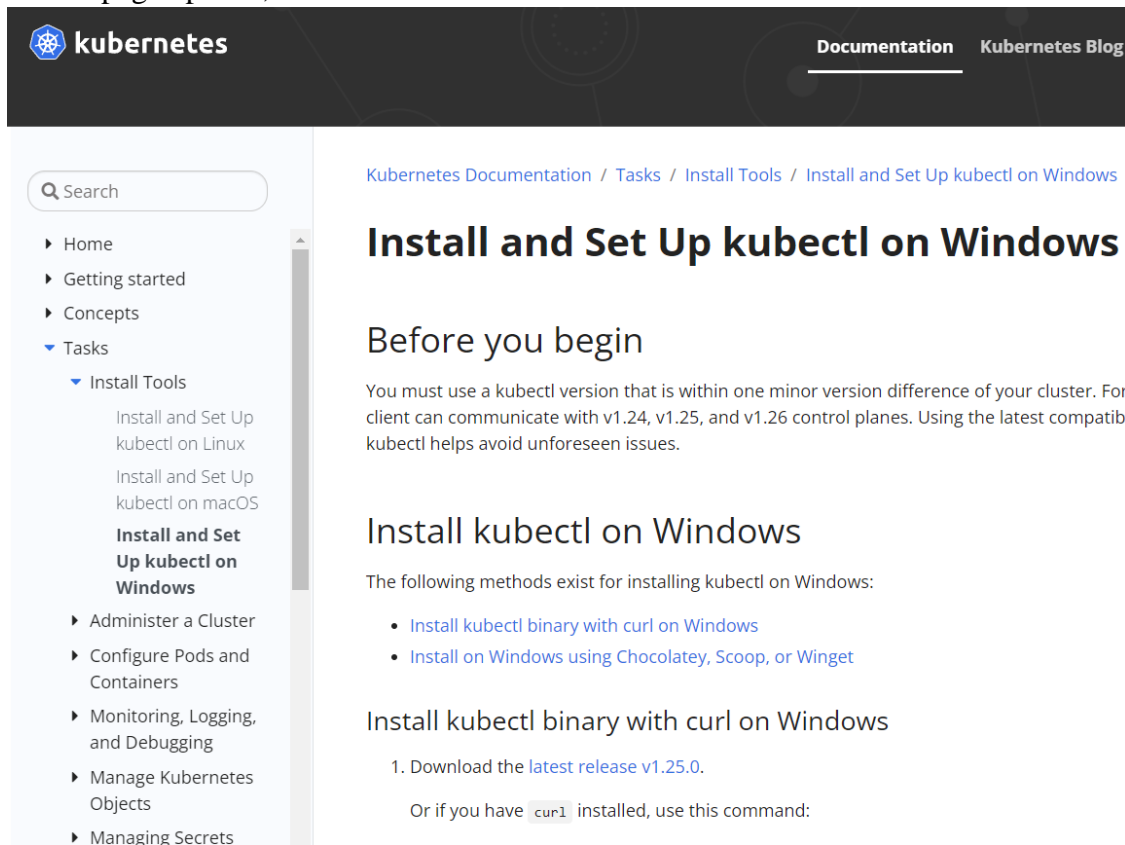


- Now for installation of **kubect1**, a CLI interface for the Kubernetes, search for **kubect1** on the google

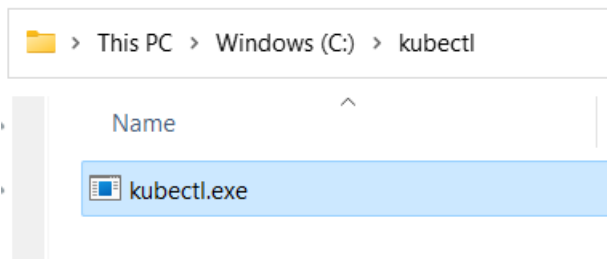


click install on windows.

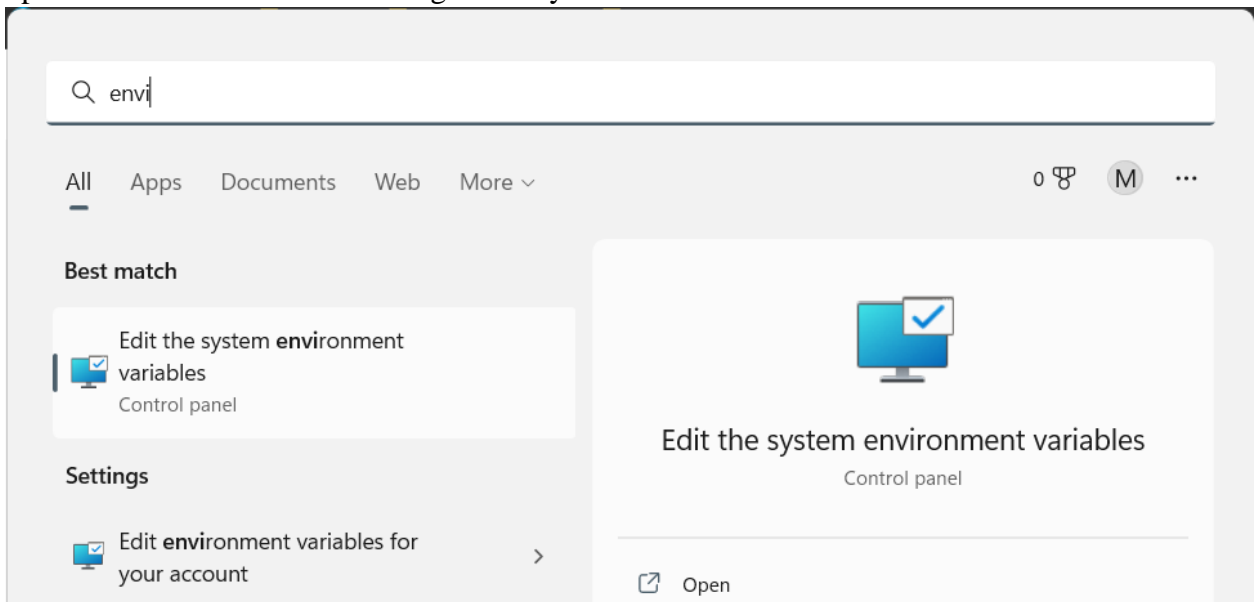
6. On the page opened, click on **latest release**



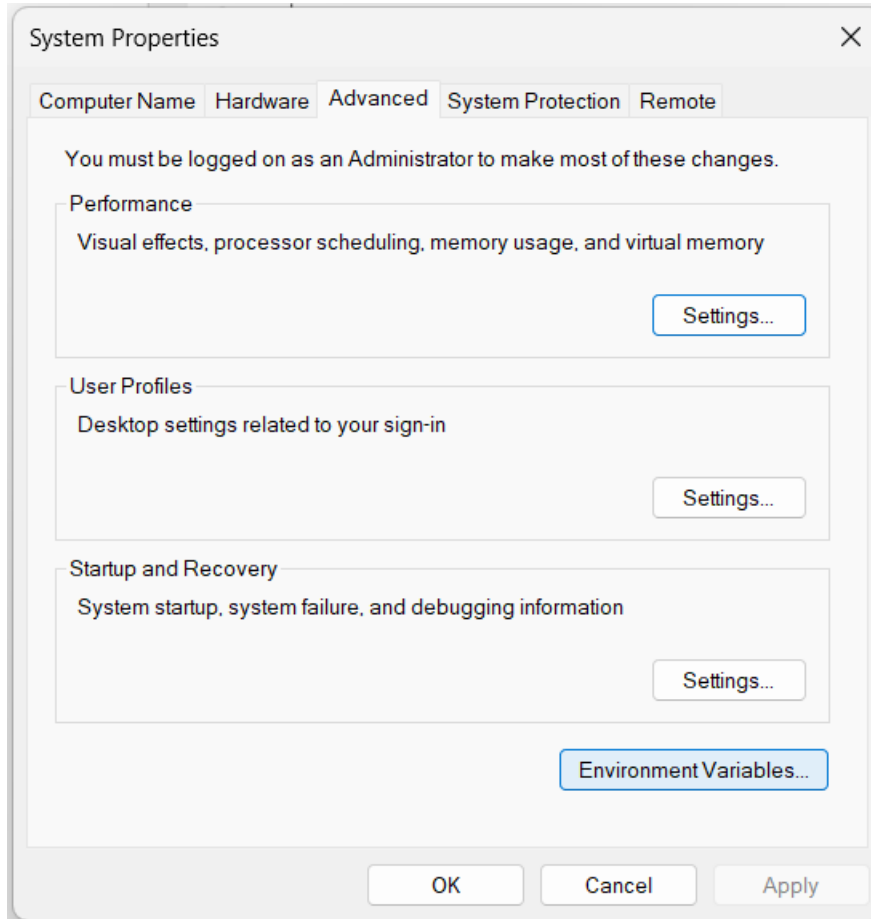
save kubectl binary somewhere in your system, a preferred location is **C:\kubectl**



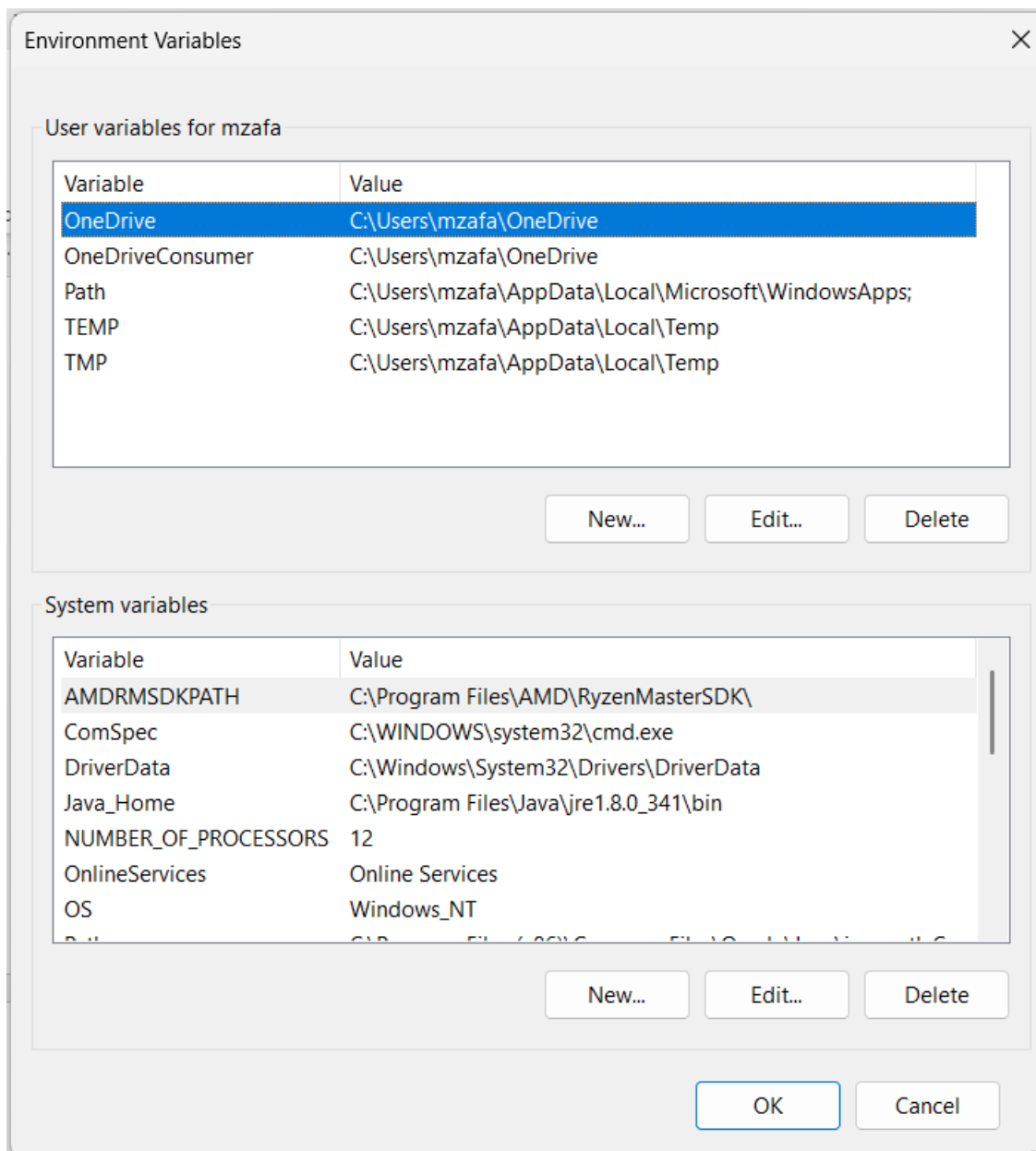
7. Now, on search, look for “Environment”, Environment Variable link will appear, click and open Environment variable setting of the system.



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

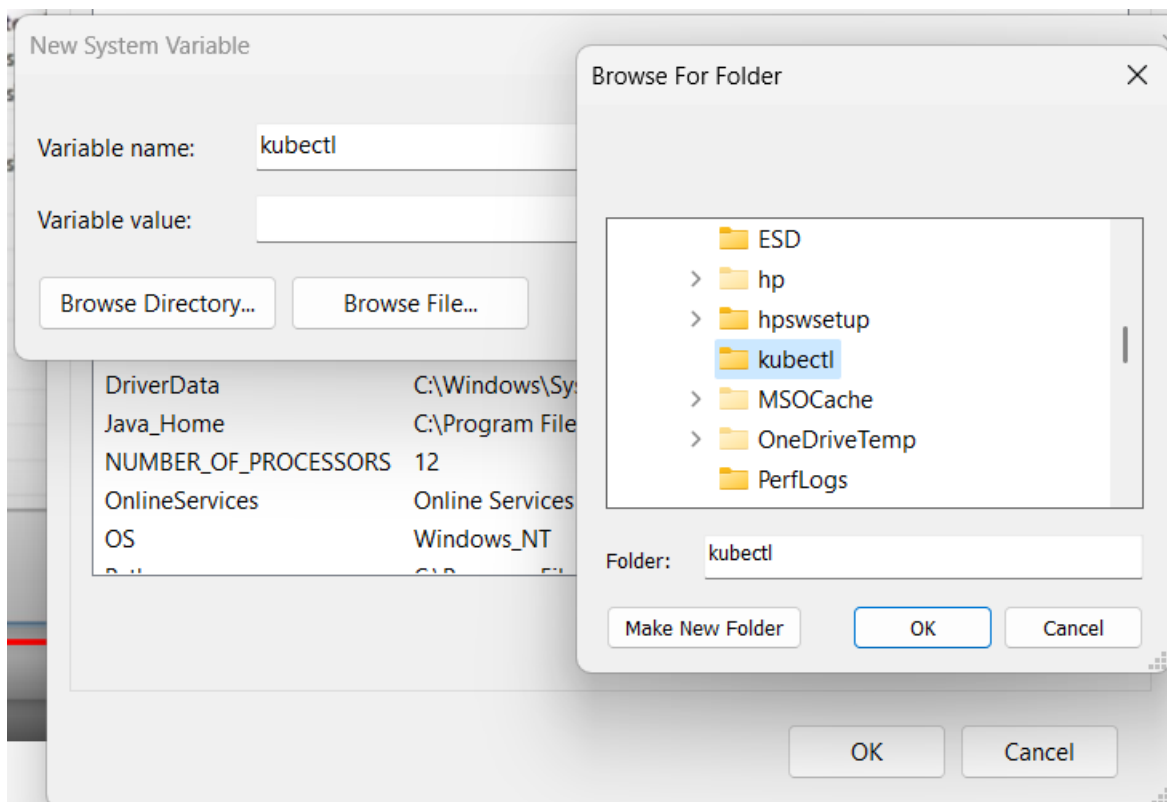


9. On the subsequent window opened, either select **New** on User variables section (if you want to make changes for current user only) or select **New** on System variables section (this will make changes system wise for all user of the system)

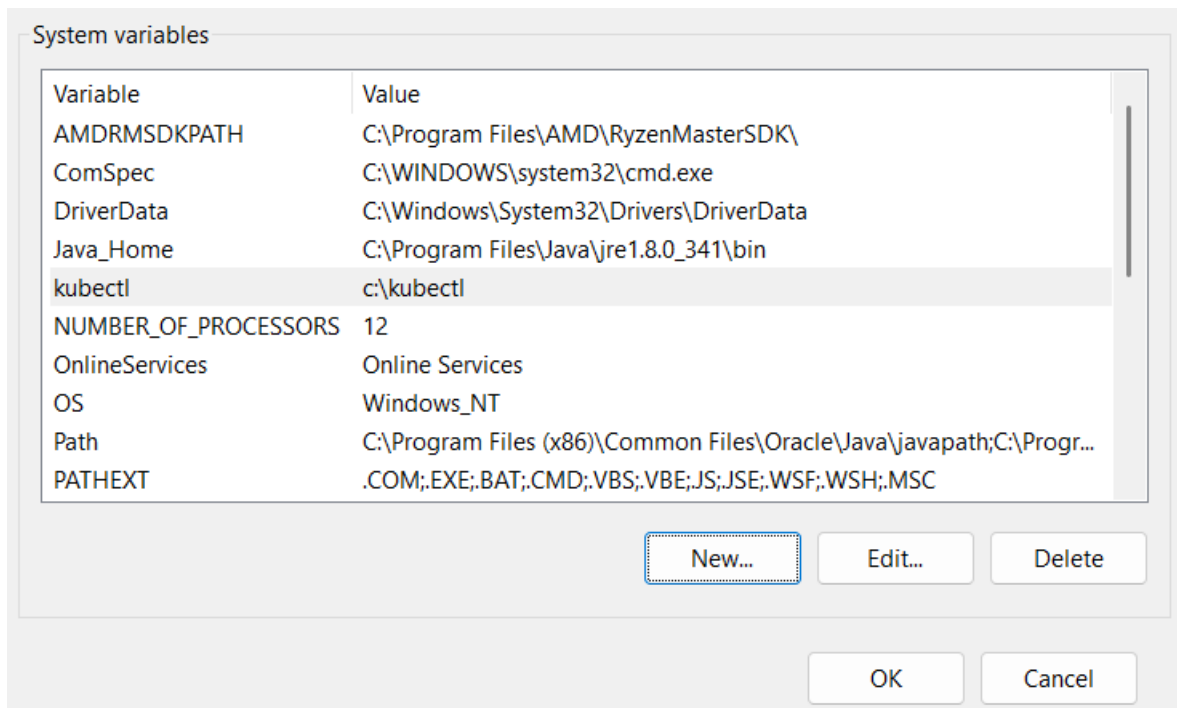


10. Add variable, write kubectl for the variable name and either provide value for this or browse to the location where **kubectl.exe** file was saved.



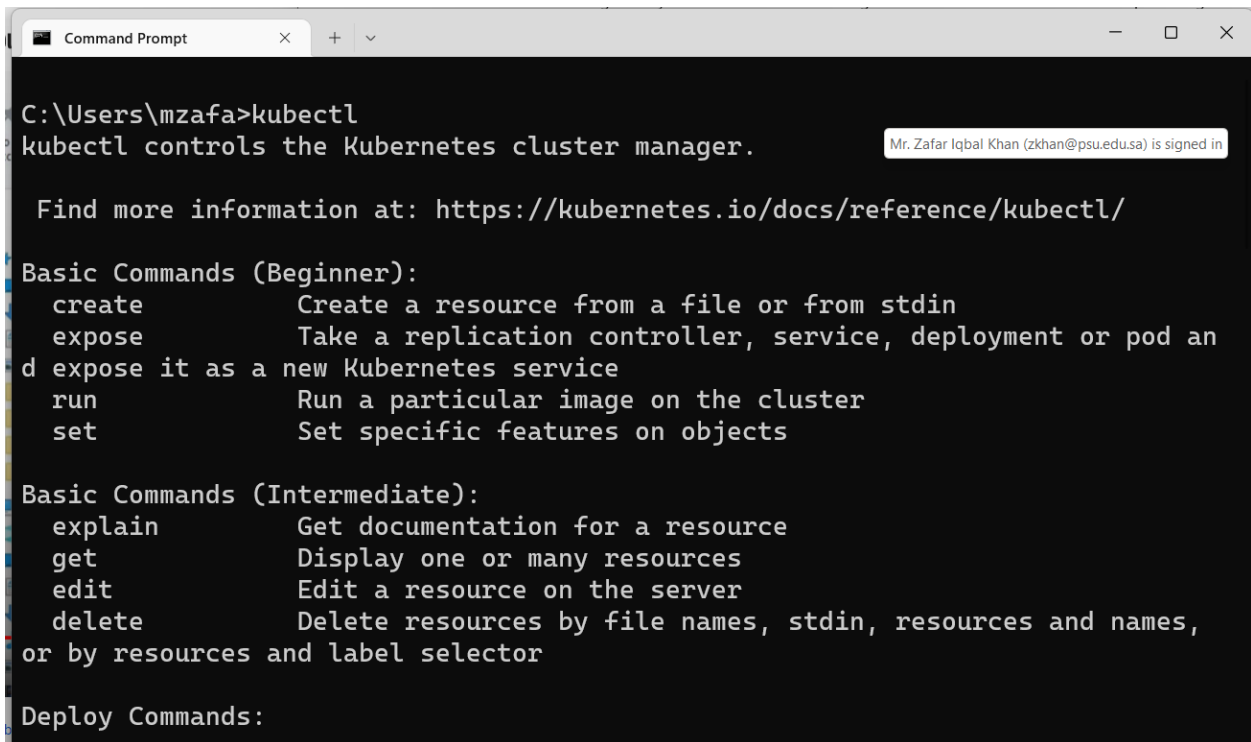


Once done, click on Ok, a new Environment Variable entry has been made, and it should be visible as well.



Click on OK to finalize everything.

11. Open command prompt, and type **kubectl**, if it responds with a verbose screen like below, means you have set everything perfectly.



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

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

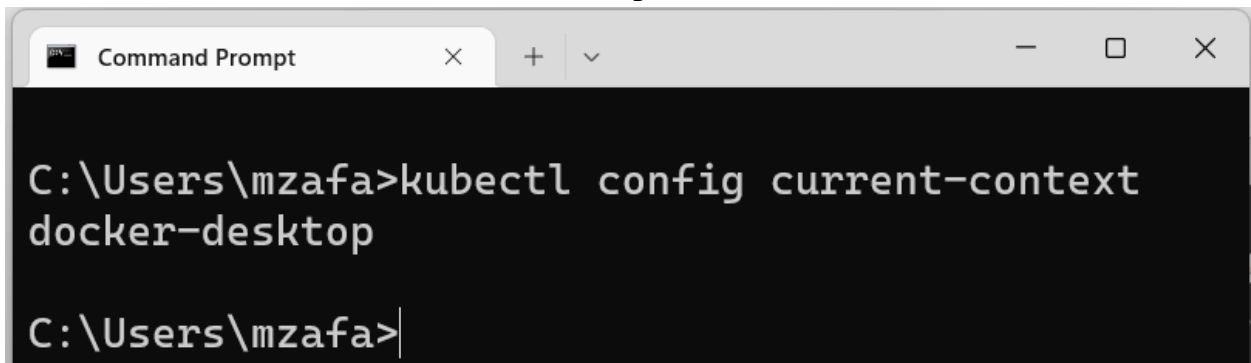
Basic Commands (Beginner):
  create          Create a resource from a file or from stdin
  expose          Take a replication controller, service, deployment or pod and
  expose it as a new Kubernetes service
  run             Run a particular image on the cluster
  set             Set specific features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

### Some basic commands

12. On command window, write **kubectl config current-context**

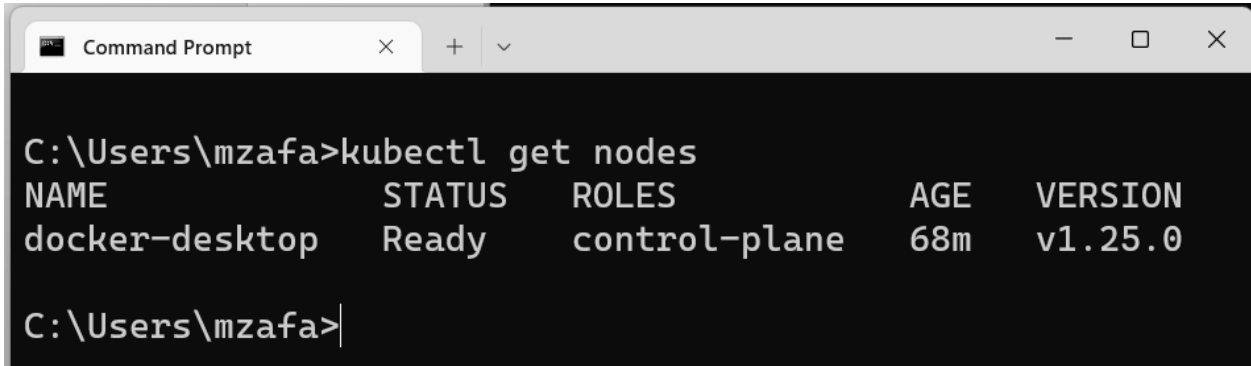


```
C:\Users\mzafa>kubectl config current-context
docker-desktop

C:\Users\mzafa>
```

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

13. To know the information about the nodes available on current cluster, run these commands

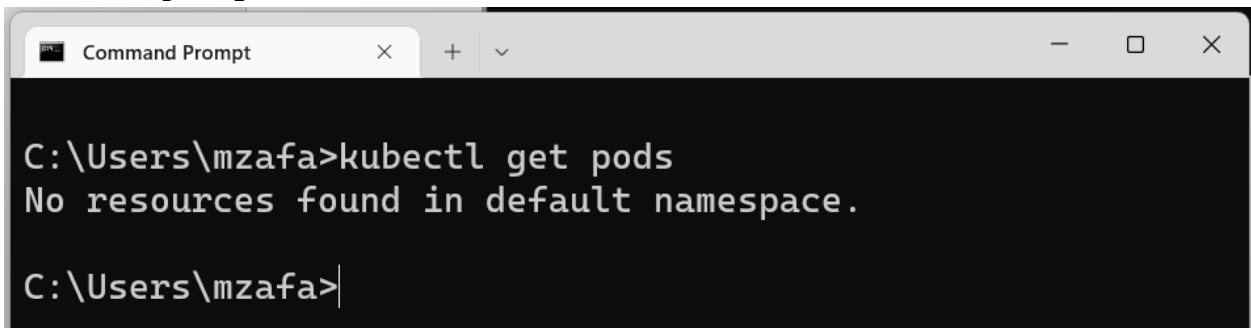


```
C:\Users\mzaafa>kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
docker-desktop      Ready    control-plane   68m   v1.25.0

C:\Users\mzaafa>
```

it shows that only one machine is running

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

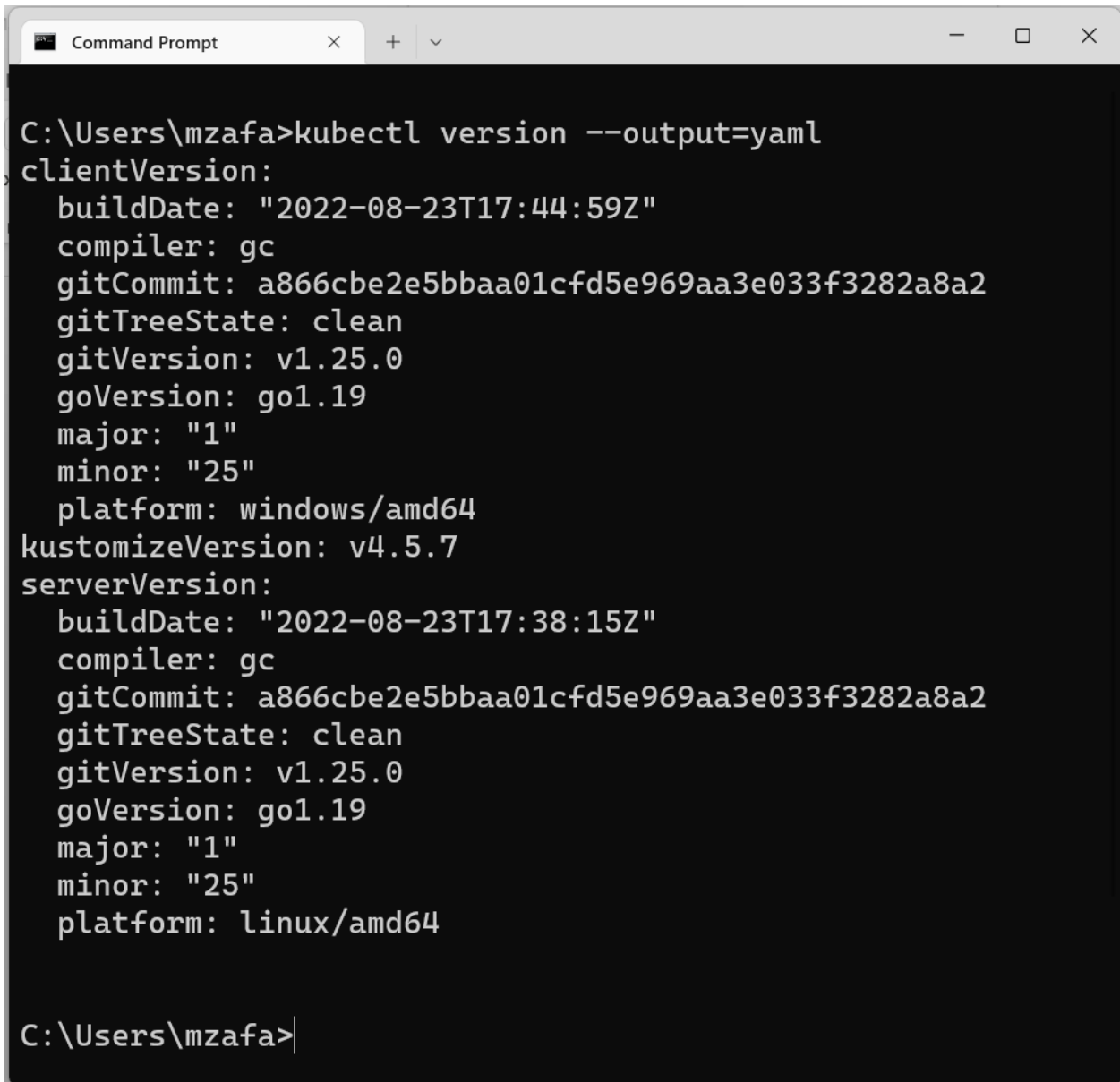


```
C:\Users\mzaafa>kubectl get pods
No resources found in default namespace.

C:\Users\mzaafa>
```

it shows that currently no pods are available

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



```
Command Prompt
C:\Users\mzafa>kubectl version --output=yaml
clientVersion:
  buildDate: "2022-08-23T17:44:59Z"
  compiler: gc
  gitCommit: a866cbe2e5bbaa01cfd5e969aa3e033f3282a8a2
  gitTreeState: clean
  gitVersion: v1.25.0
  goVersion: go1.19
  major: "1"
  minor: "25"
  platform: windows/amd64
kustomizeVersion: v4.5.7
serverVersion:
  buildDate: "2022-08-23T17:38:15Z"
  compiler: gc
  gitCommit: a866cbe2e5bbaa01cfd5e969aa3e033f3282a8a2
  gitTreeState: clean
  gitVersion: v1.25.0
  goVersion: go1.19
  major: "1"
  minor: "25"
  platform: linux/amd64

C:\Users\mzafa>
```