SE489 DevOps Engineering

Lab 8

# Lab 8: Installation of Kubernetes on Windows Platform

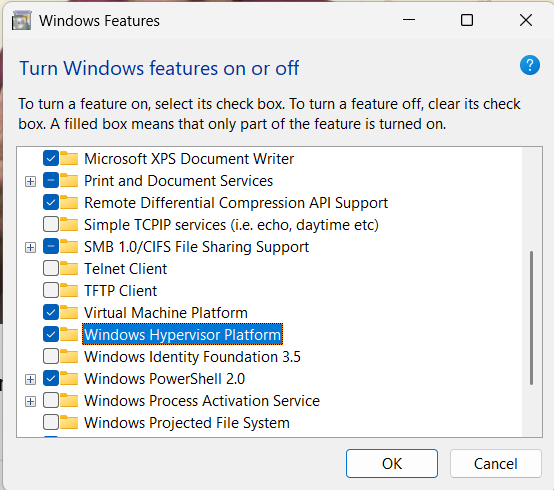
## Objectives:

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

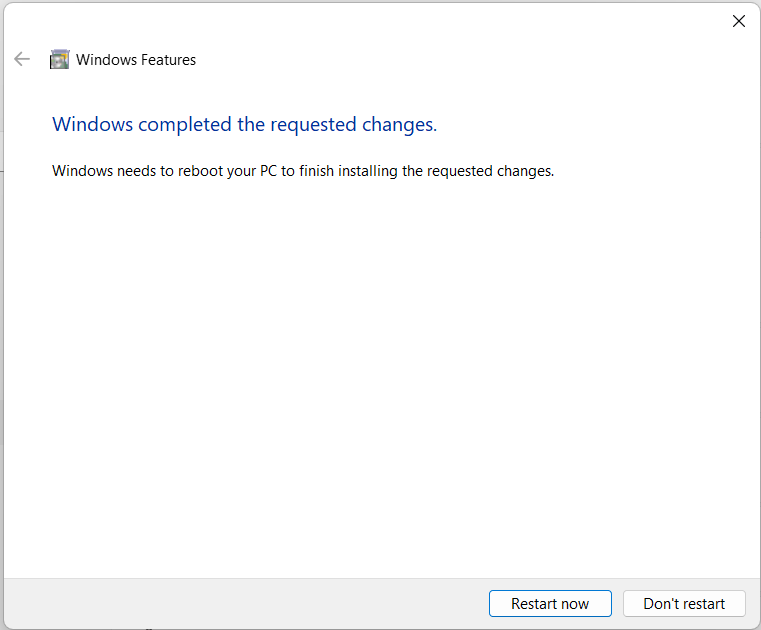
Installation of Kubernetes on windows is not straight forward, in its simplest form, which we are going to follow, has two steps-

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  
     
   Select Windows Hypervisor Platform

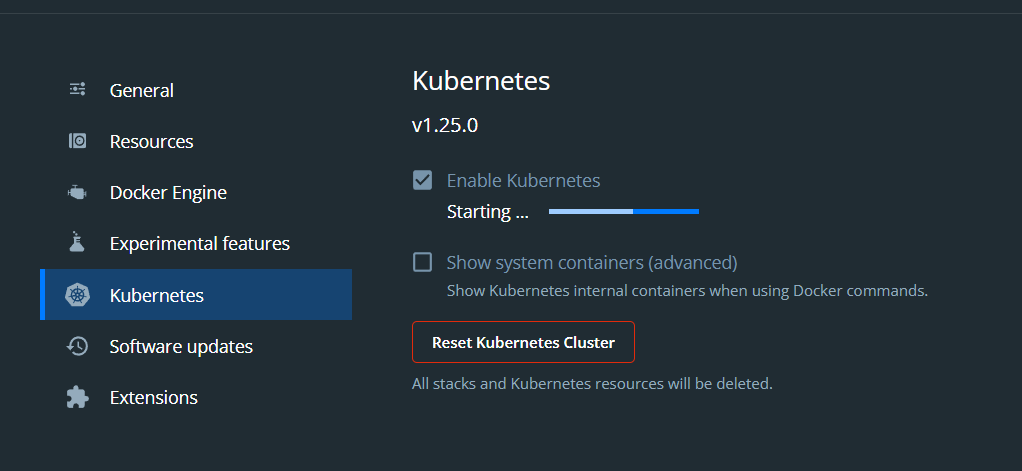
After completion of installation, following screen will appear

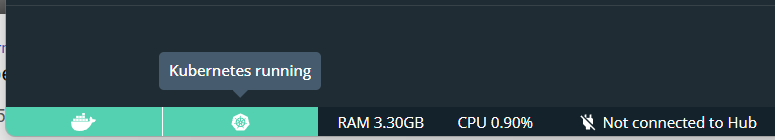
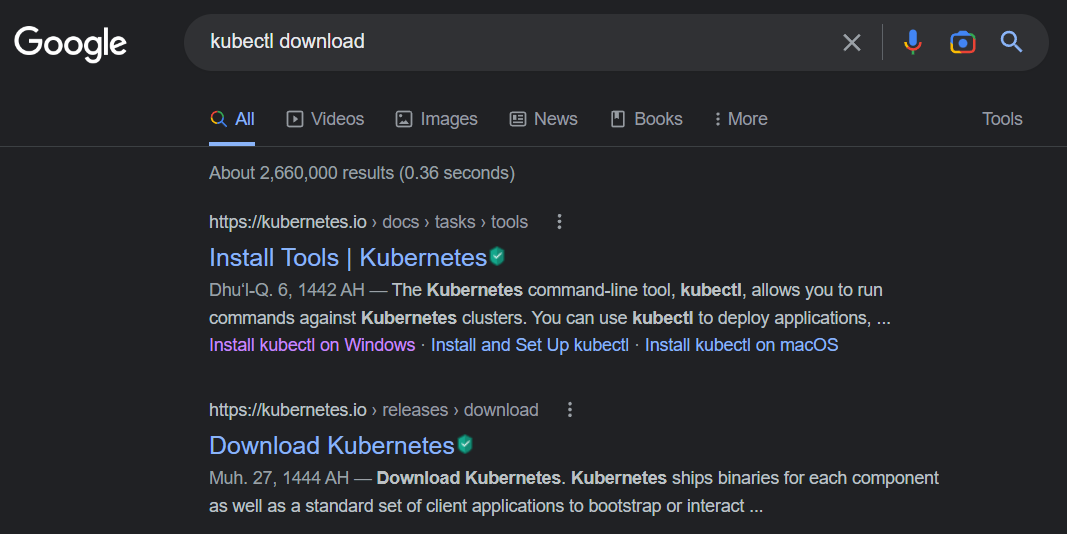
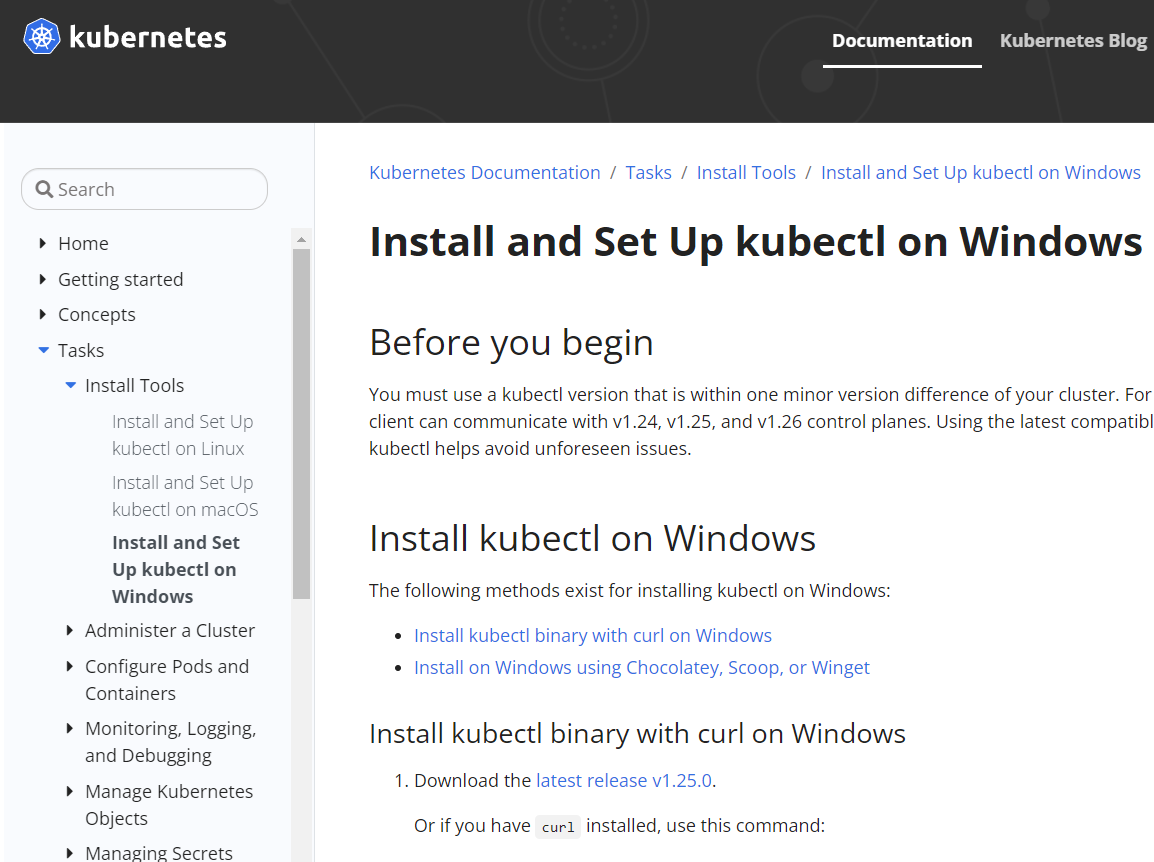
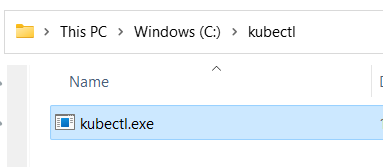
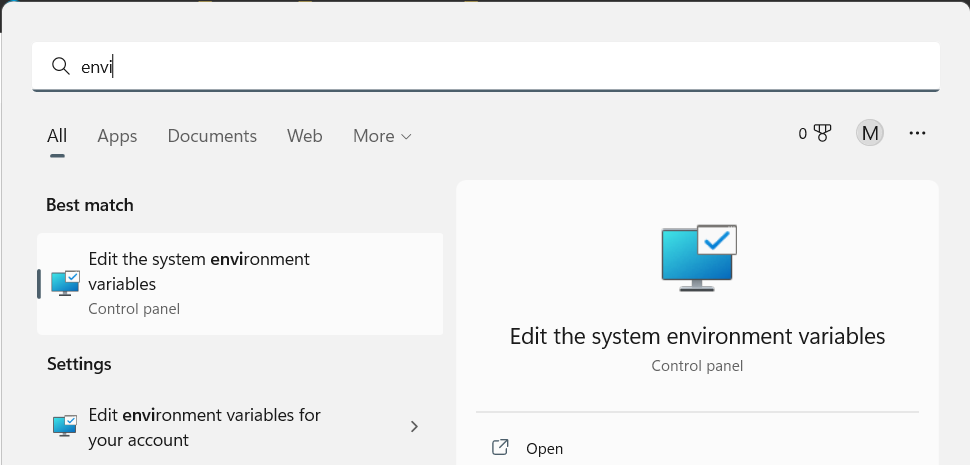
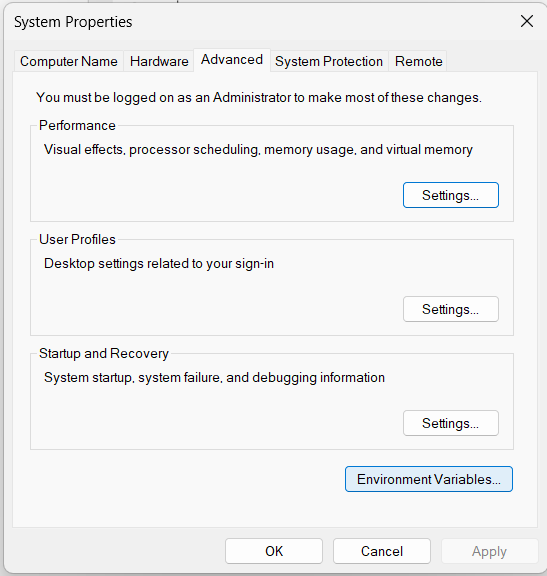
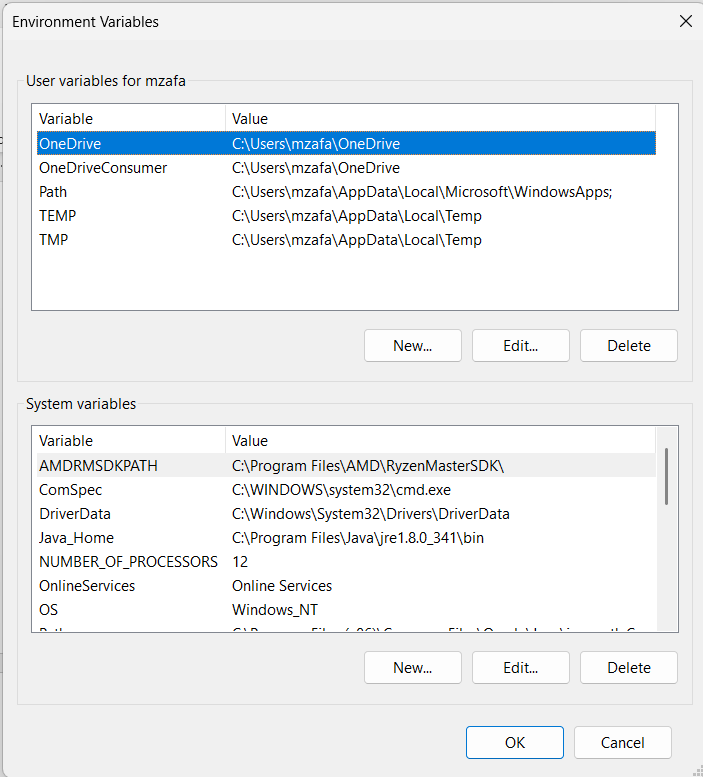
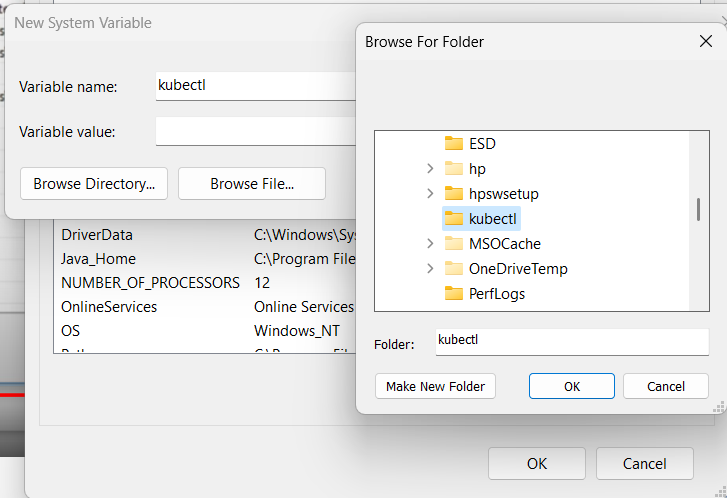
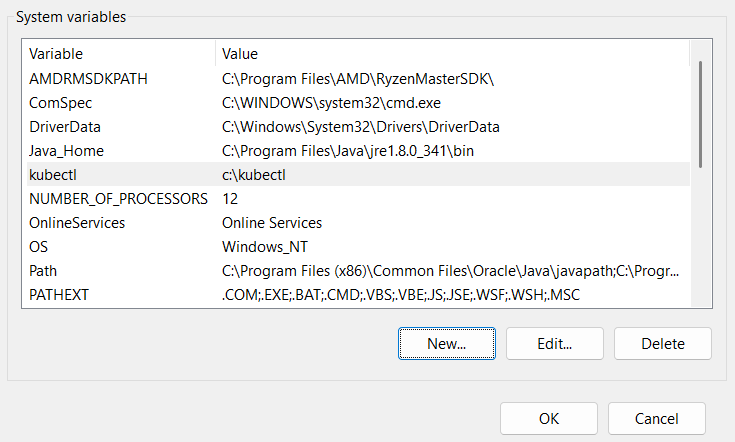


1. Now start the Docker desktop, and browse to the setting pane of the docker, click on Kubernetes, then check Enable Kubernetes, this will install standalone cluster of Kubernetes.  
   A screenshot of a computer

   Description automatically generated
2. Click on apply and restart, this will install additional components required for running Kubernetes.  
   Graphical user interface, text

   Description automatically generated

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


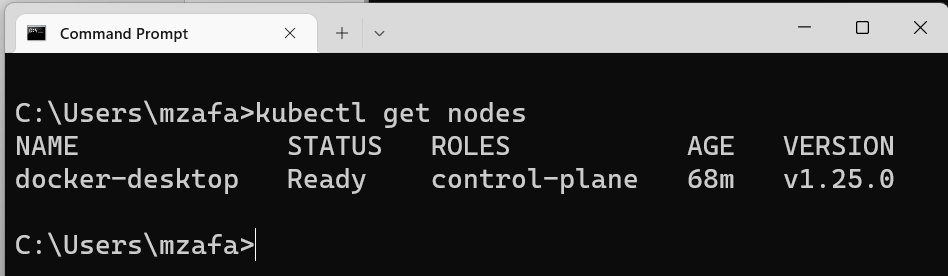
1. Once finished, you can see bottom left corner of the Docker desktop interface, Kubernetes icon appears in green along with whale icon of the docker, indicating Kubernetes has been installed and running successfully.  
   
2. Now for installation of **kubectl**, a CLI interface for the Kubernetes, search for **kubectl** on the google   
     
   click install on windows.
3. On the page opened, click on **latest release**  
   save kubectl binary somewhere in your system, a preferred location is **C:\kubectl**
4. Now, on search, look for “Environment”, Environment Variable link will appear, click and open Environment variable setting of the system.  
   
5. On the setting window opened thus, click on “Environment Variables” at the bottom  
   
6. 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)   
   
7. 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.
8. Open command prompt, and type **kubectl,** if it responds with a verbose screen like below, means you have set everything perfectly.

Text

Description automatically generated

## Some basic commands

## On command window, write kubectl config current-context Graphical user interface, text, application Description automatically generated it shows current context of the Kubernetes instance, which in turn reveals that it is going to run on docker-desktop cluster

1. To know the information about the nodes available on current cluster, run these commands  
     
   it shows that only one machine is running
2. To know about pods available on the current cluster, use this command

**kubectl get pods**Graphical user interface, text, application

Description automatically generatedit shows that currently no pods are available

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