SE489 DevOps Engineering

Lab 9

Lab 9: Exploring KubeCTL

Objectives:

Student will learn about Minikube use case cluster, installation, configuration and maneuvering of it.

Introduction of Minikube:

Minikube is library which lets one use Kubernetes on one's computer without need of installing other things for using clusters and associated maneuvering.

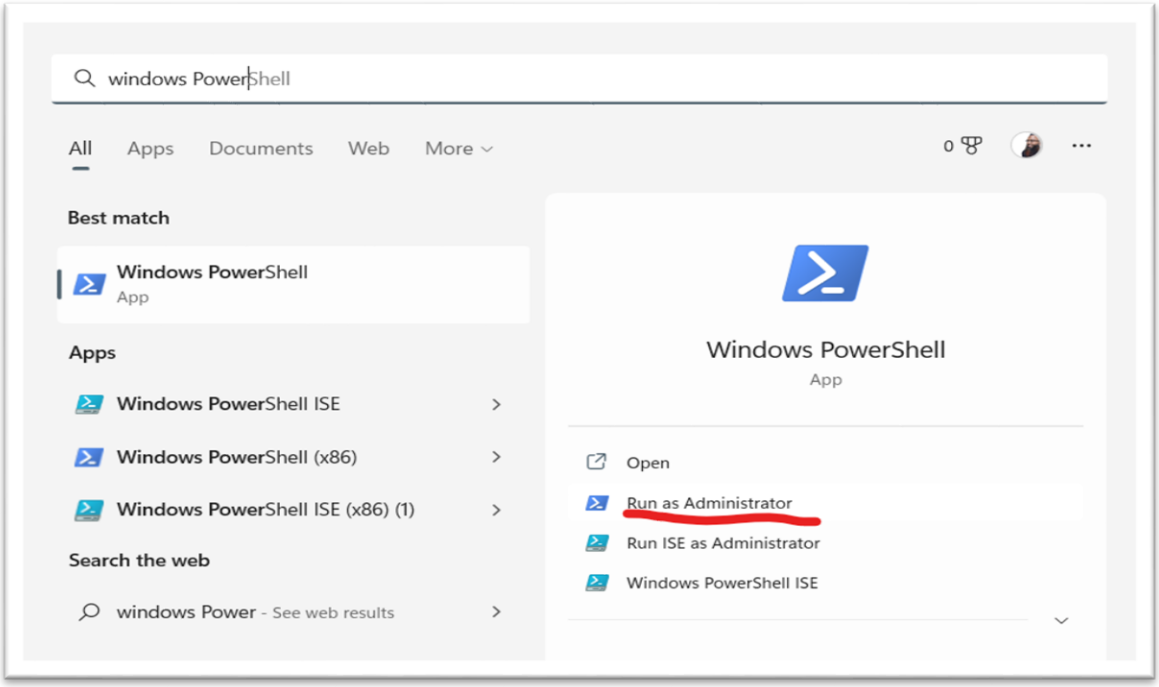
In a nutshell, Minikube is a one node Kubernetes cluster, which runs on your laptop.

In its simplest form, we will first install Chocolatey Installer, and then with the help of this we will install minikube.

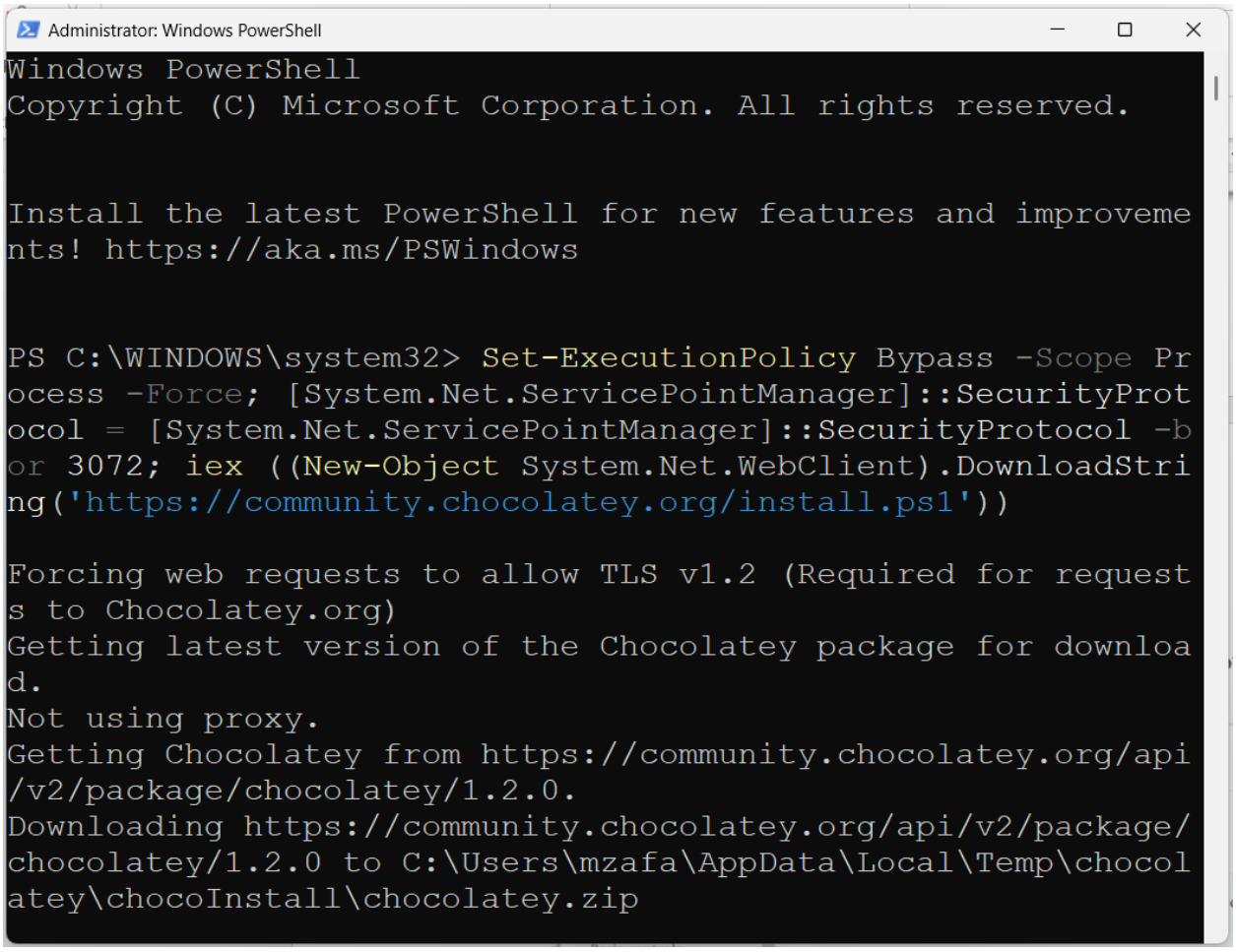
1. Installation of Chocolatey

Open Windows Power Shell in Administrator mode, and then run this script on the powershell, wait a few minutes for windows to complete the installation of the script.

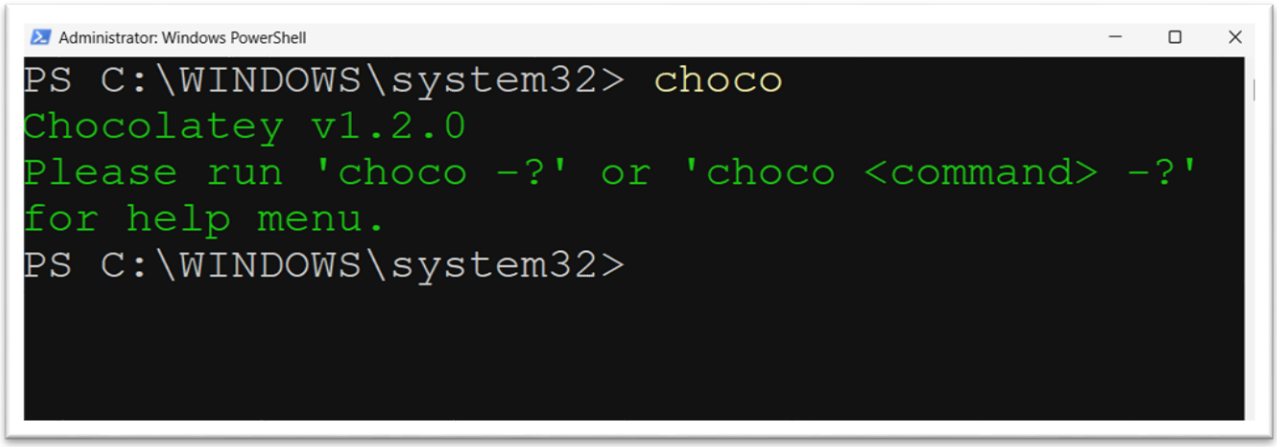
```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-
Object
System.Net.WebClient).DownloadString('https://community.chocolatey.org/ins
tall.ps1'))
```



Now paste the above script on the PowerShell prompt



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

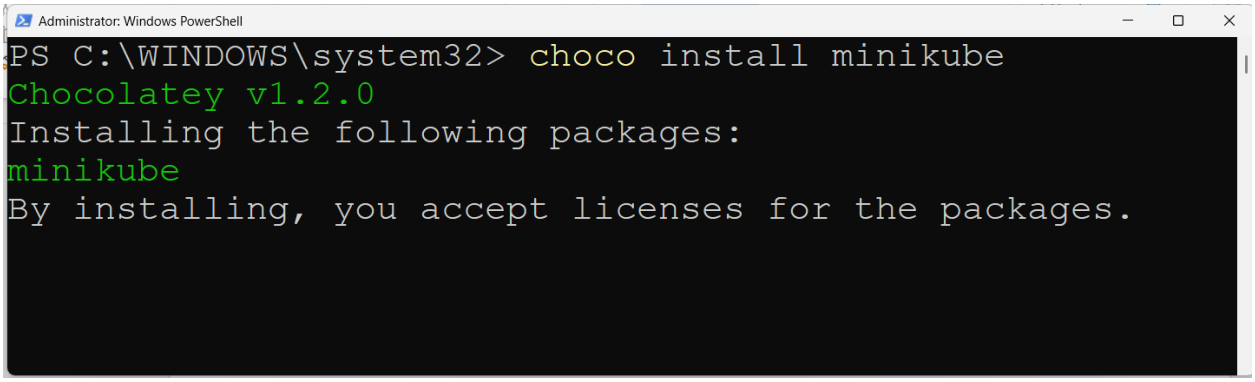


Obviously Chocolatey has been installed successfully on the system

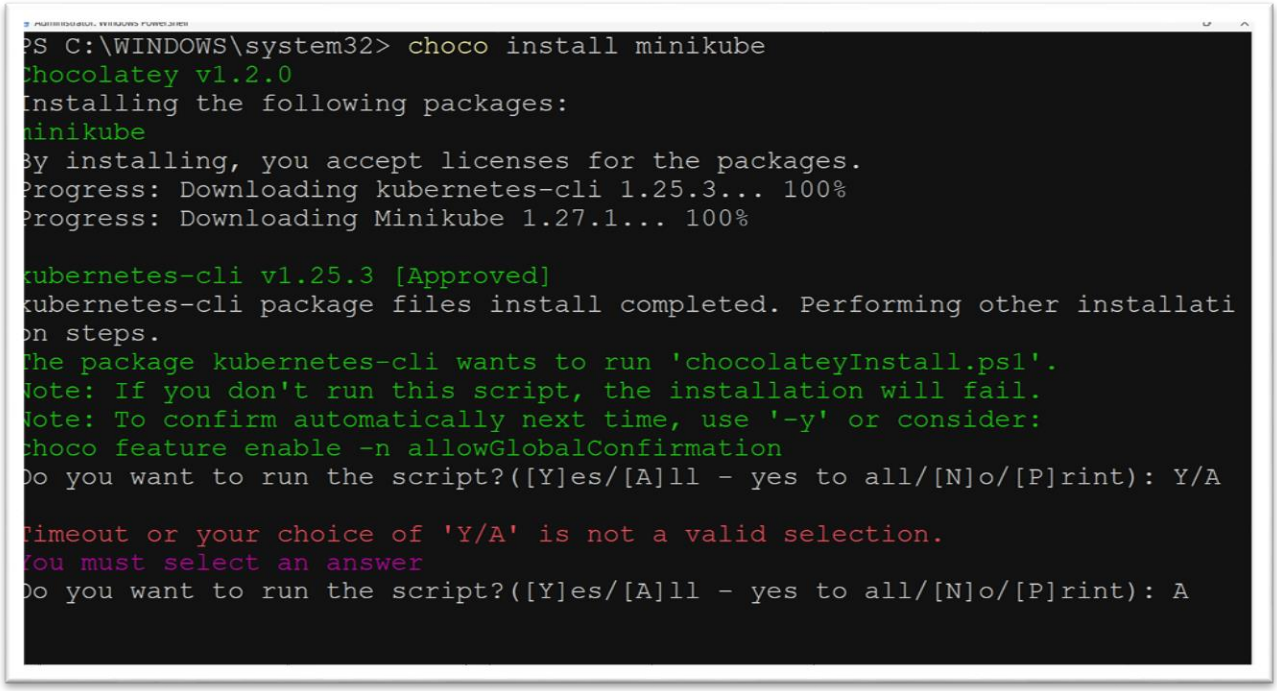
2. Installation of Minikube

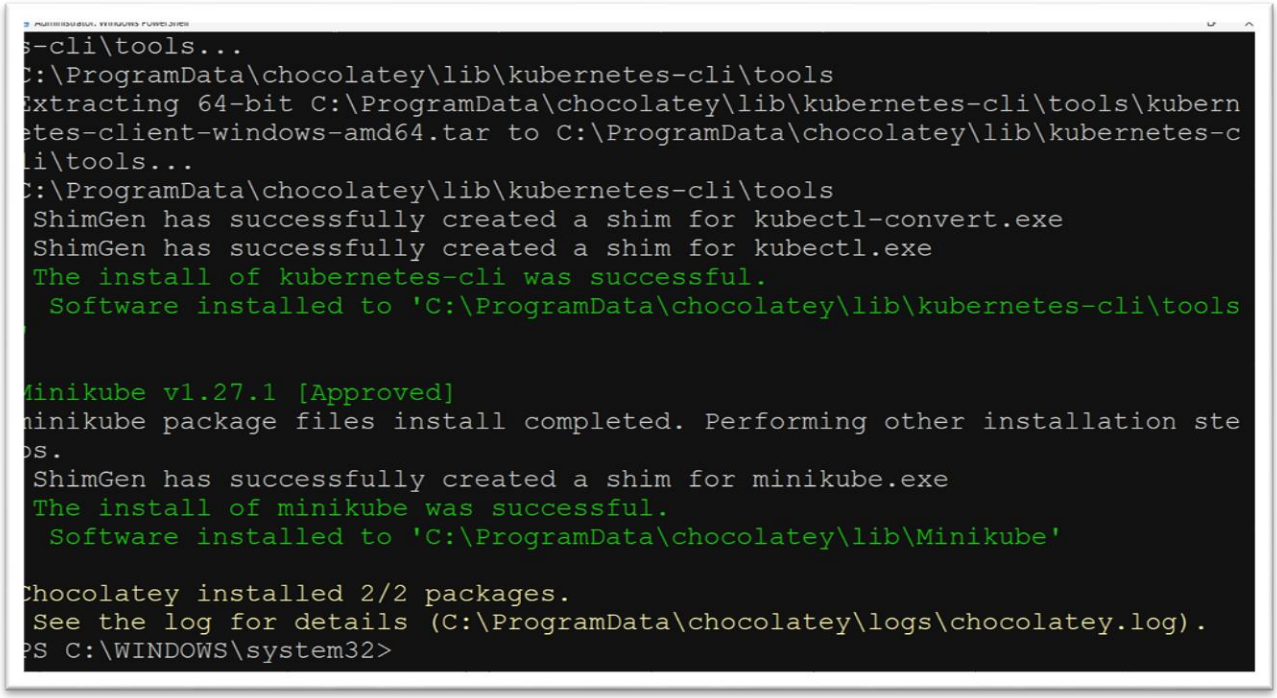
On the PowerShell terminal (**opened as administrator**) run following command to install Minikube on the system

choco install minikube

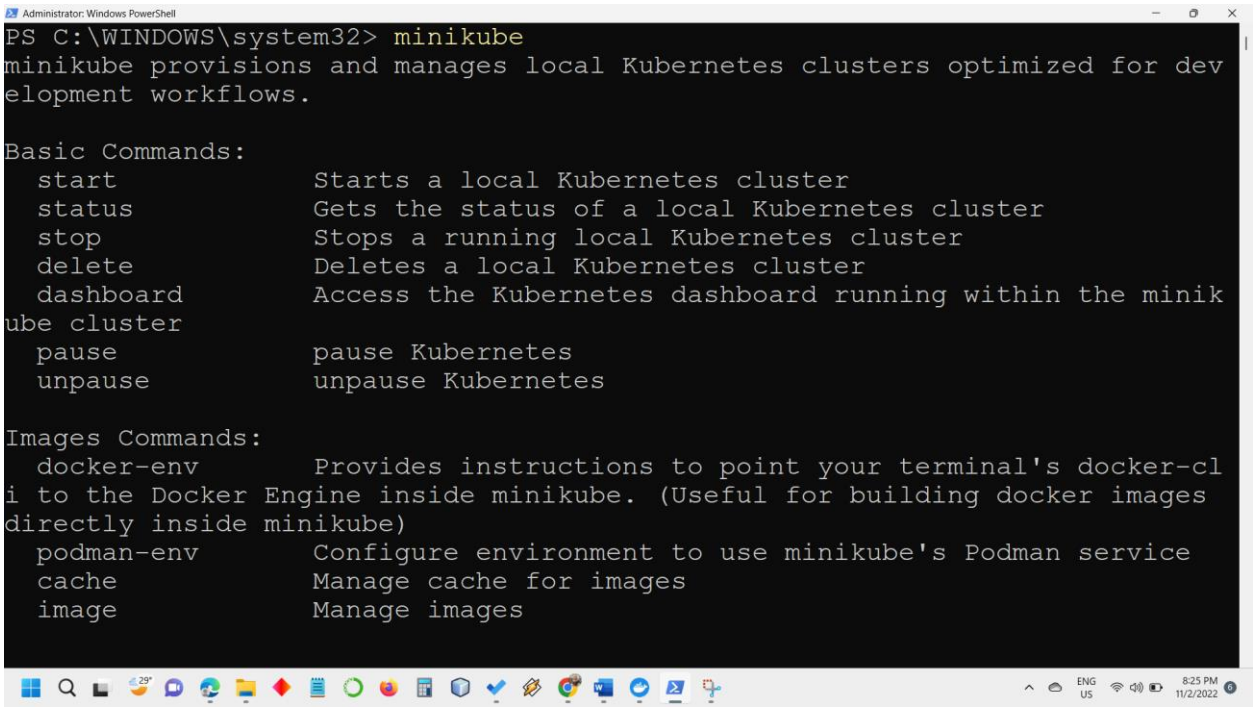


When asked give permission as A





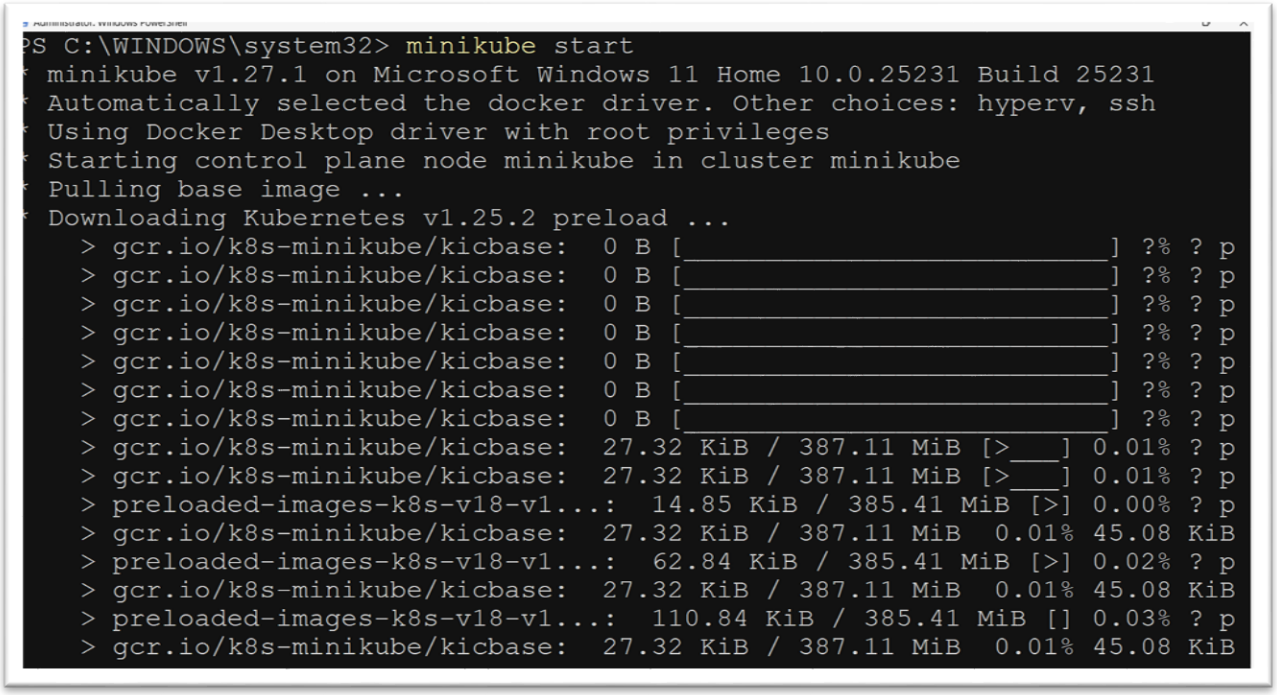
When installation has been completed successfully, above screen will appear.



A verbose screen is evidence of successful installation of minikube.

You have installed a usecase cluster on Kubernetes of your system.

3. To start cluster, run **`minikube start`**



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

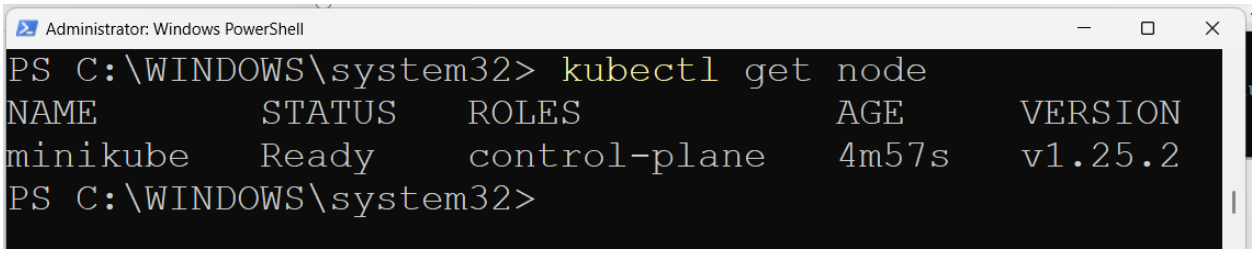
Once download is complete, cluster preparation will begin

```
Administrator: Windows PowerShell
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_________________________] ?% ? p/s 1
m10s
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.25.2 on Docker 20.10.18 ...\
```

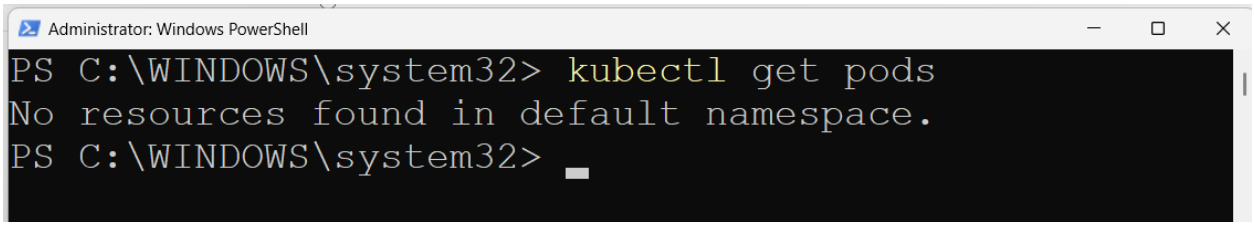
Once done, following message will appear

```
Administrator: Windows PowerShell
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_____________________________] ?% ? p
    > gcr.io/k8s-minikube/kicbase:  0 B [_________________________] ?% ? p/s 1
m10s
* Creating docker container (CPUs=2, Memory=2200MB) ...
* Preparing Kubernetes v1.25.2 on Docker 20.10.18 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" n
amespace by default
PS C:\WINDOWS\system32>
```

4. Now check the number of nodes in the Kubernetes, with kubectl

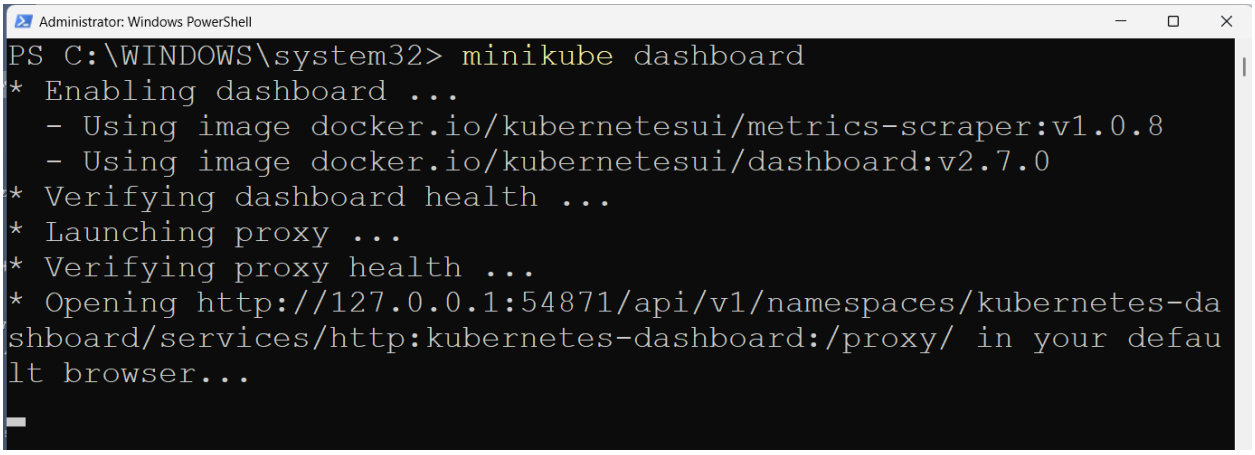


5. Check for pods

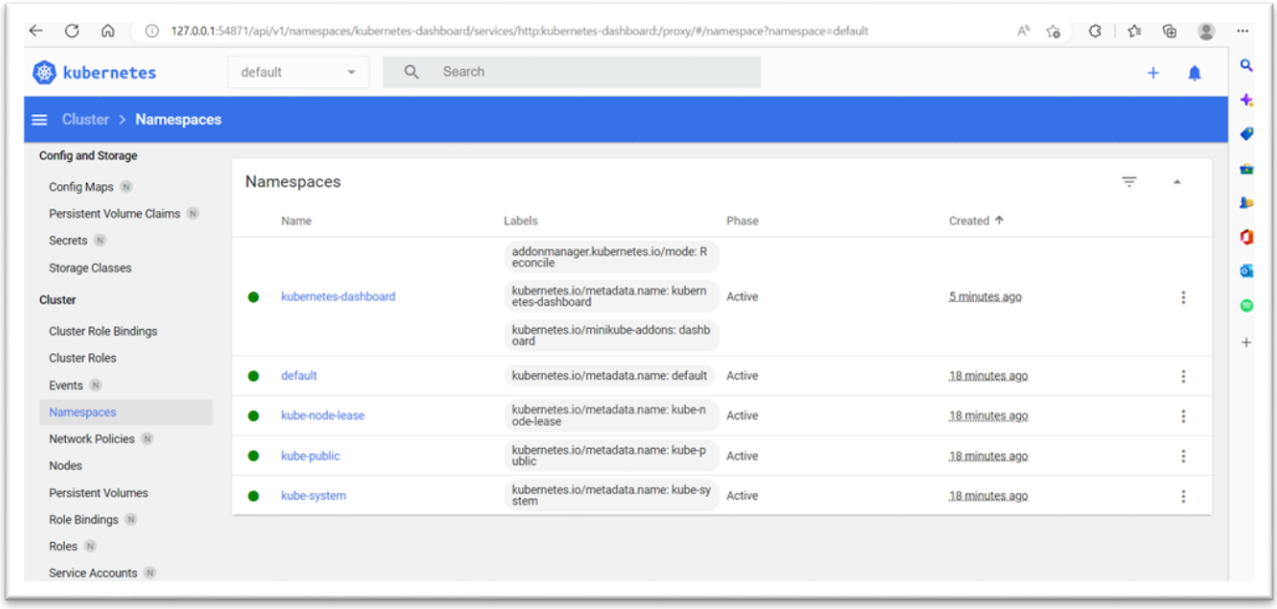


which means there are no active pods in the deployment

6. Minikube is bundled with a dashboard for further investigation and accurate information about cluster, let's launch this dashboard

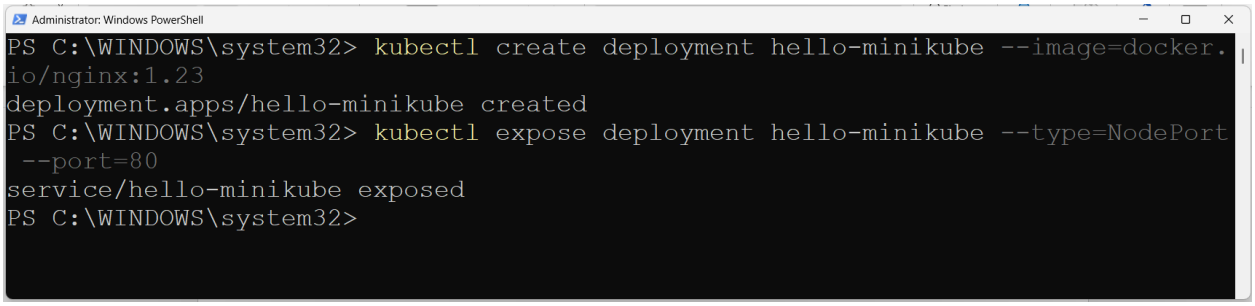


7. A default web browser window will open, showing Kubernetes dashboard with various information and insight into the cluster, click on Namespaces on the left pane, all the information about namespaces will be displayed

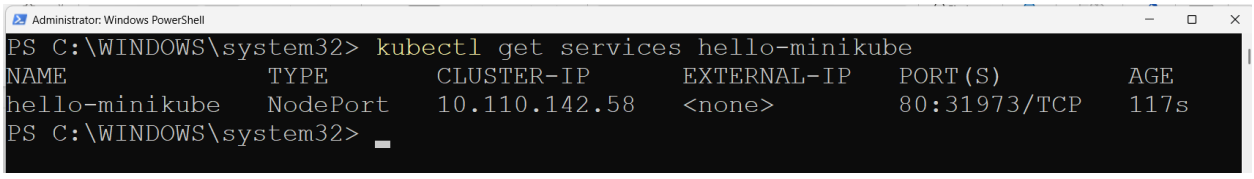


8. Let's create a sample application, and deploy it on this cluster, and expose this to port 80
kubectl create deployment hello-minikube --image=docker.io/nginx:1.23

kubectl expose deployment hello-minikube --type=NodePort --port=80

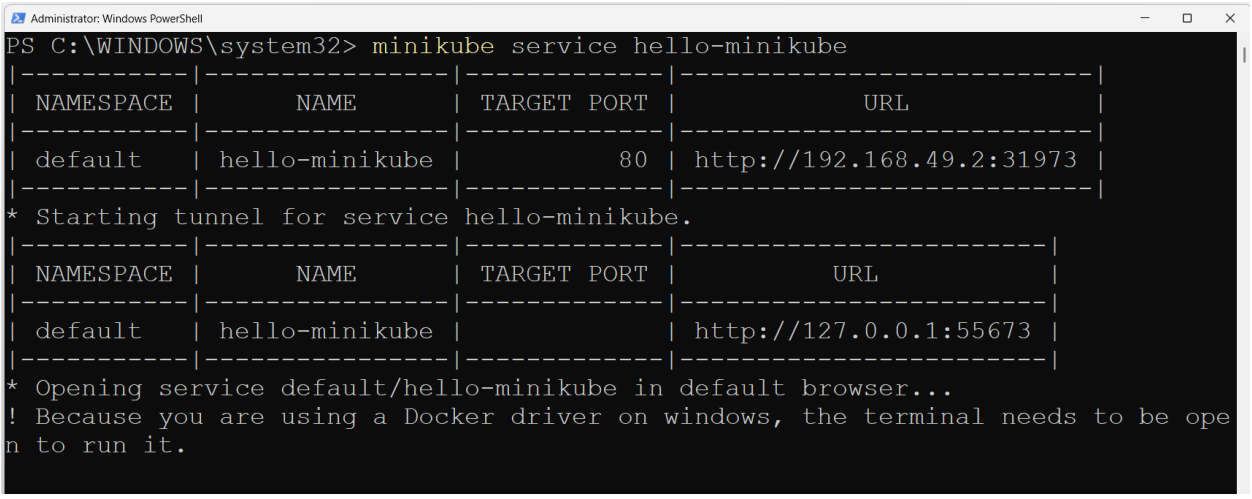


9. Let's check the cluster again for the deployment, we have just created with following command,
kubectl get services hello-minikube



we can see that Kubernetes has assigned an internal ip to our cluster and port mapping is also there

9. Let’s launch this service and see the output of this service, run this command on the PowerShell, **`minikube service hello-minikube`**

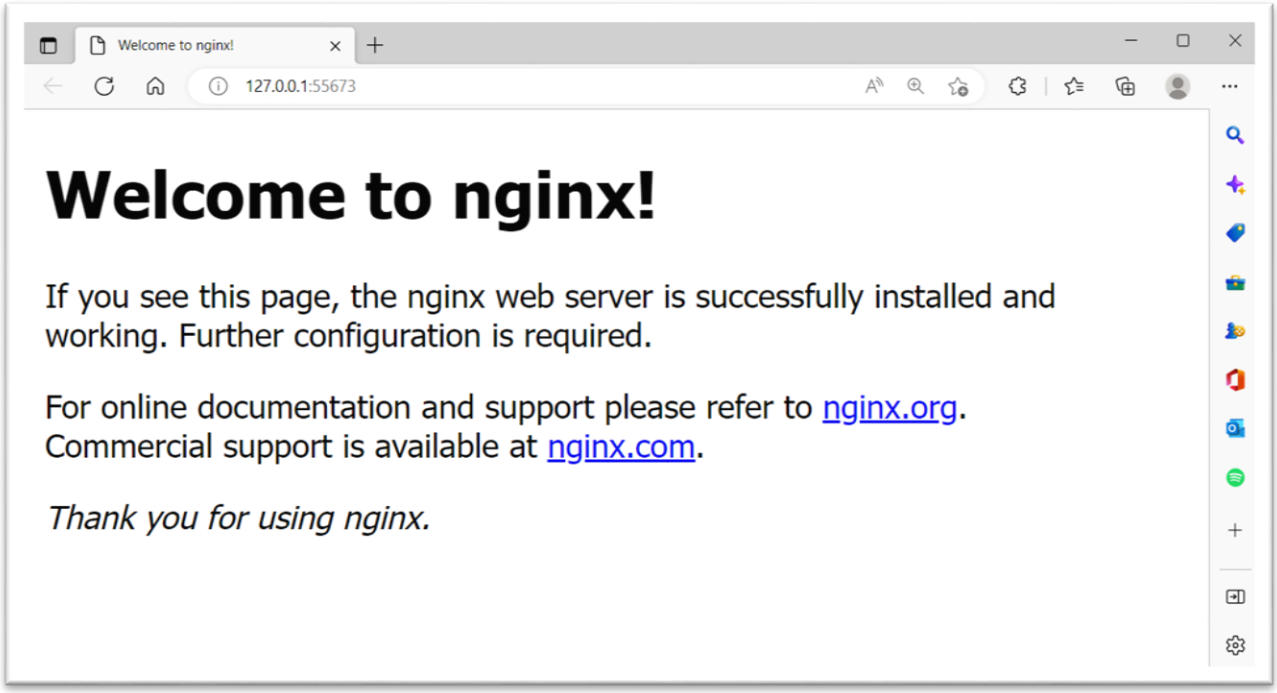


A mapping table is displayed on the console screen showing namespace, name target port and url of the service.

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**

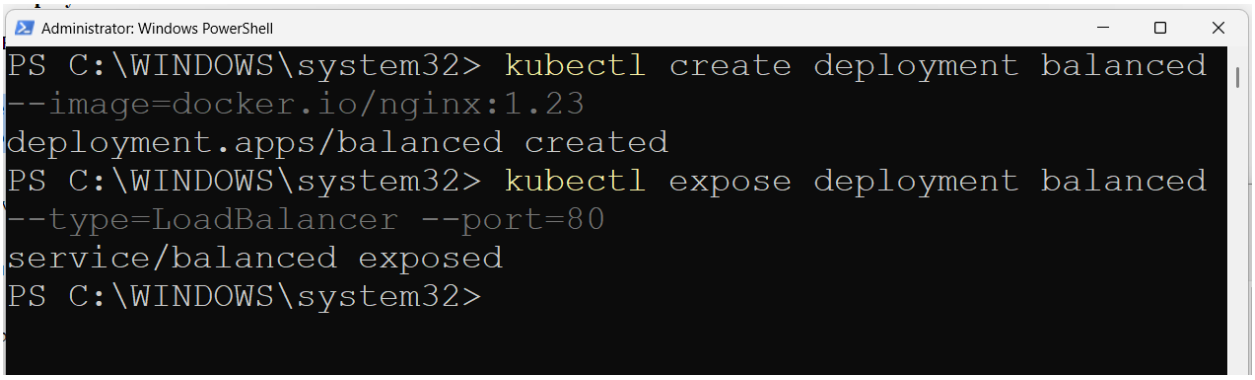
10. Because of previous command a new default web browser window will open, showing home page for nginx server



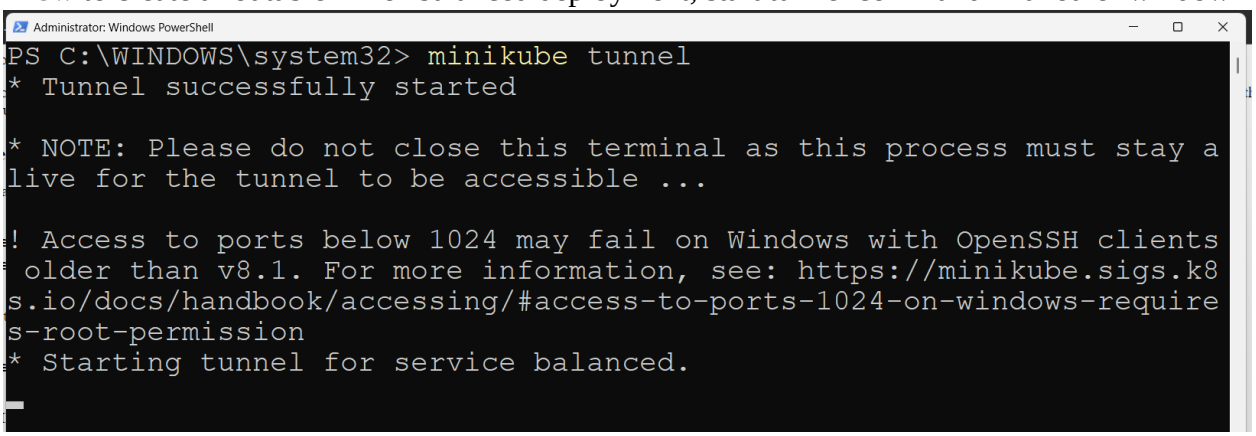
LoadBalancer Deployment commands

11. 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
```

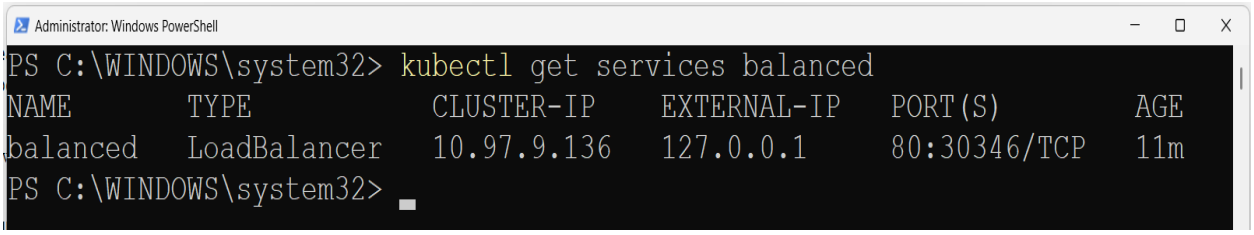


12. Now to create a routable IP for balanced deployment, start tunnel command in another window



this window should remain open, in order to tunnel be available

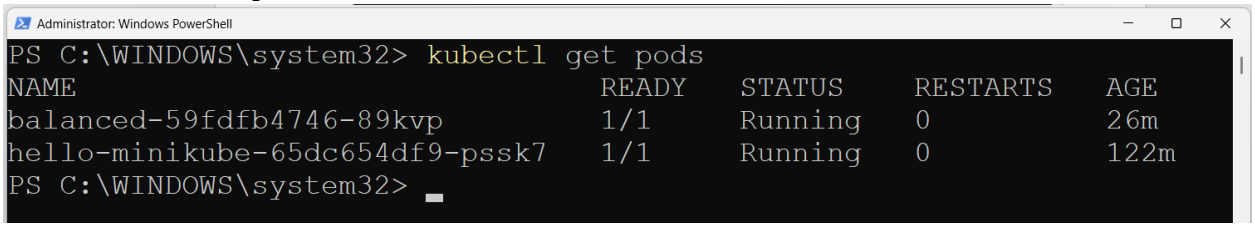
13. To know the external IP, run minikube get services balanced



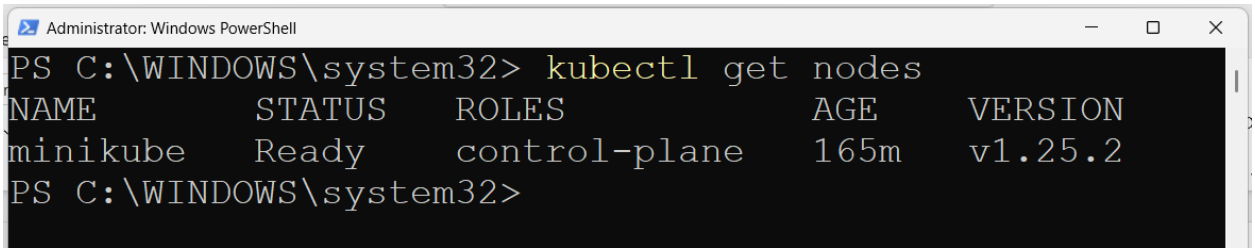
deployment can be accessed with ***http:\\<External IP>:80***

Some Administrative commands

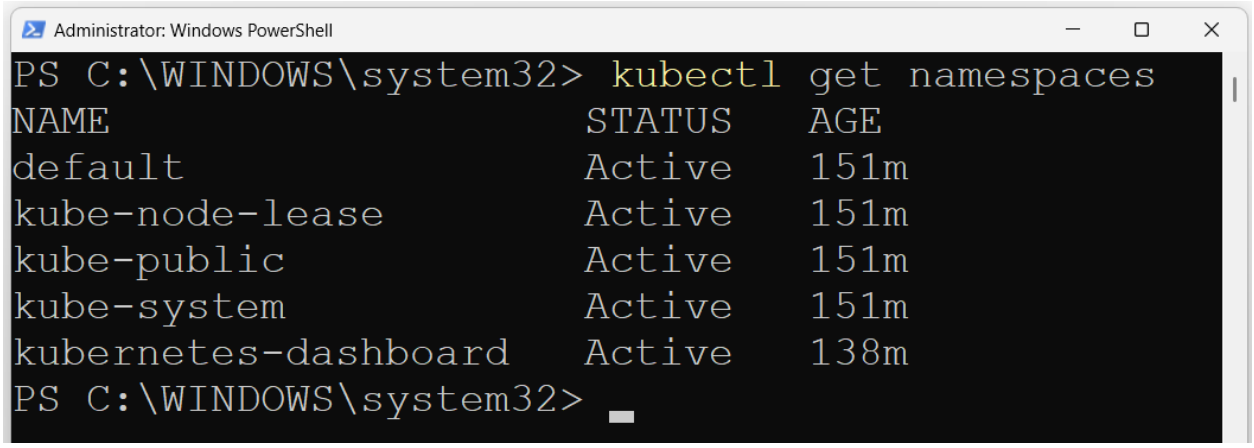
14. To know about the pods in the cluster, run this



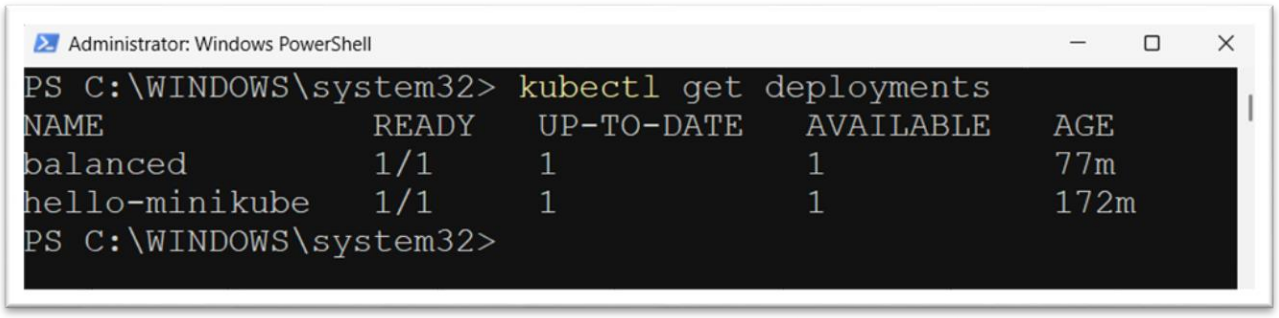
15. To know about the nodes in the cluster



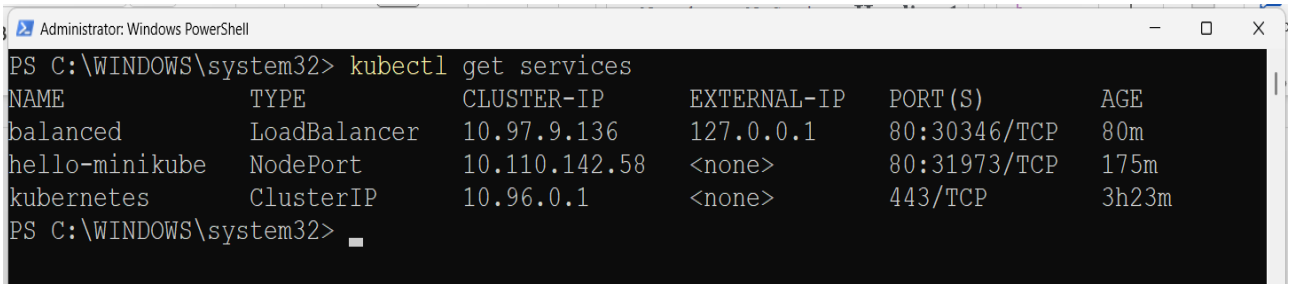
16. To know about the namespaces



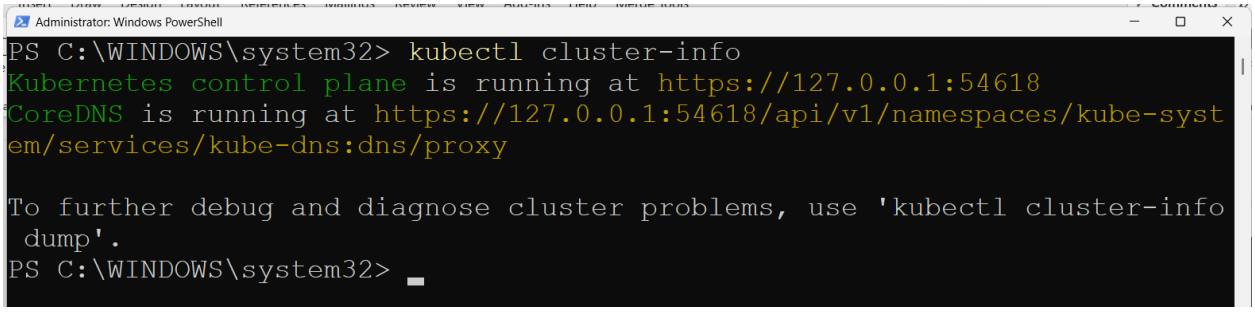
17. To know about the deployments in the cluster



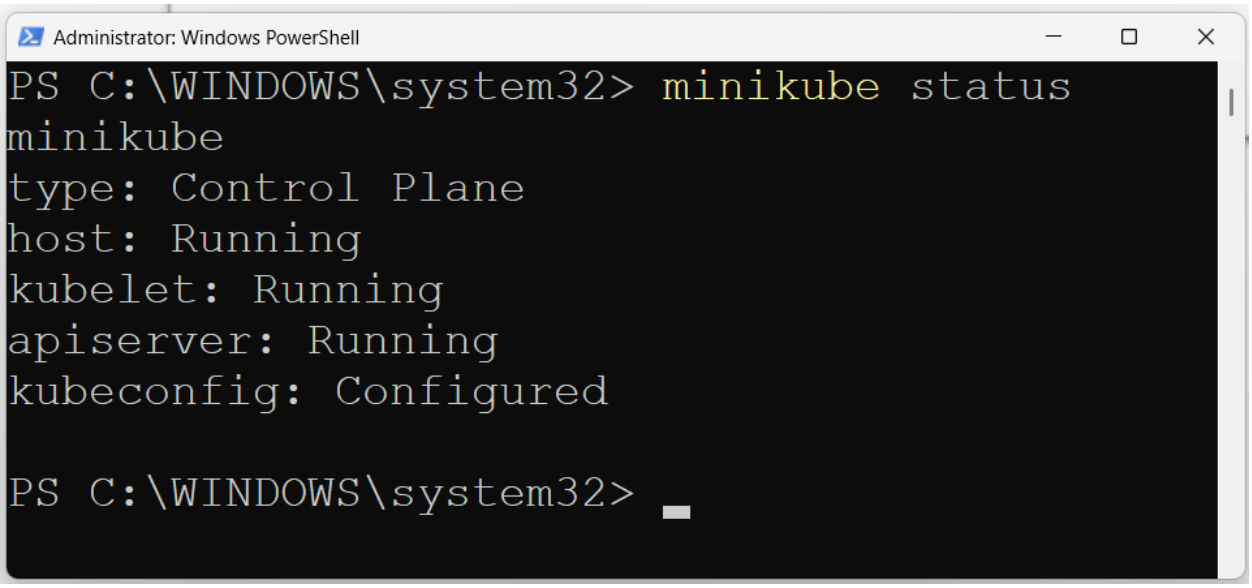
18. To know about the services in the cluster



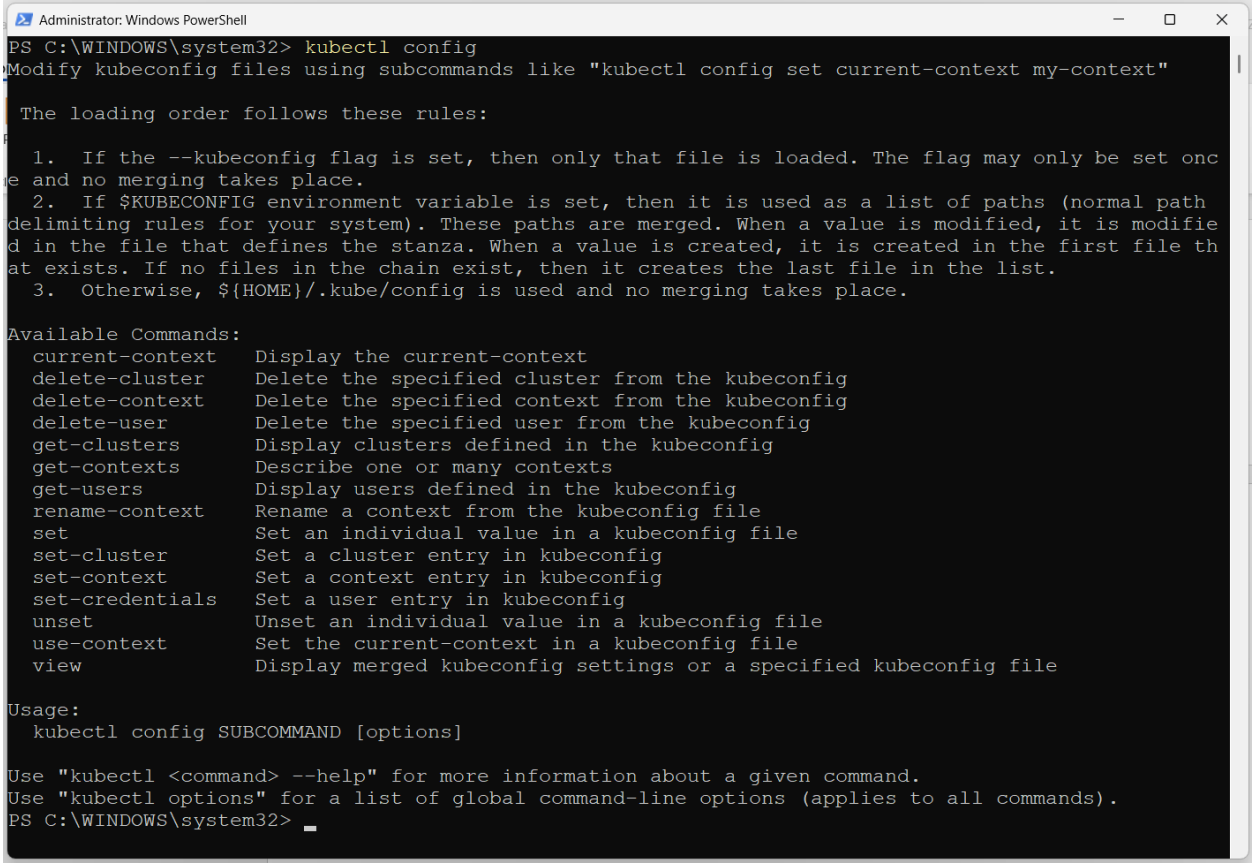
19. To know about the cluster



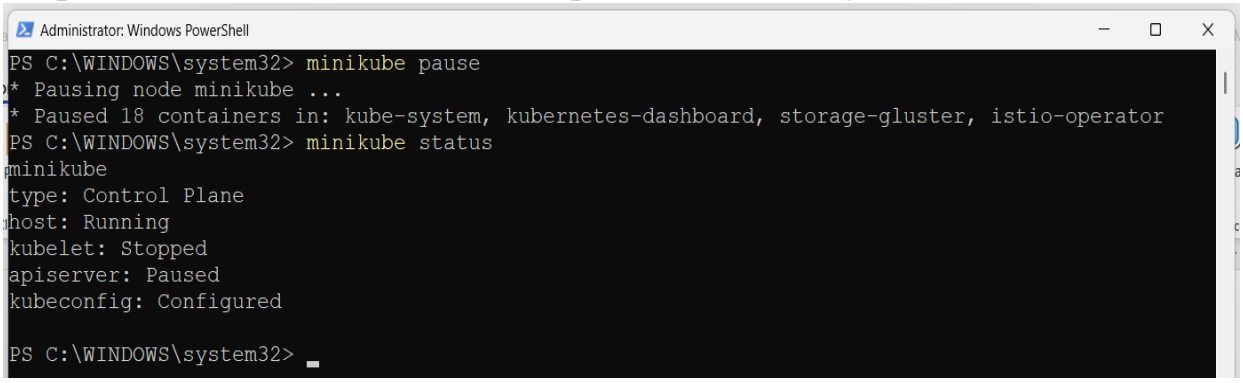
20. To know about the status of the cluster



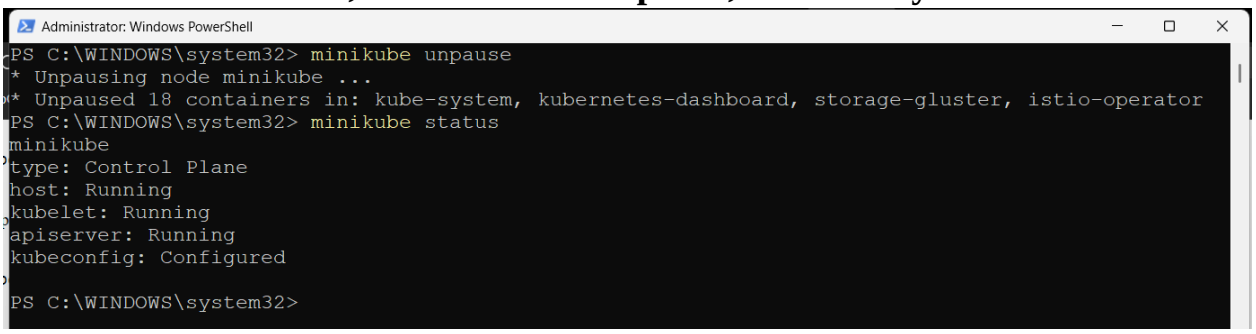
21. **To learn about configuration detail, run command**



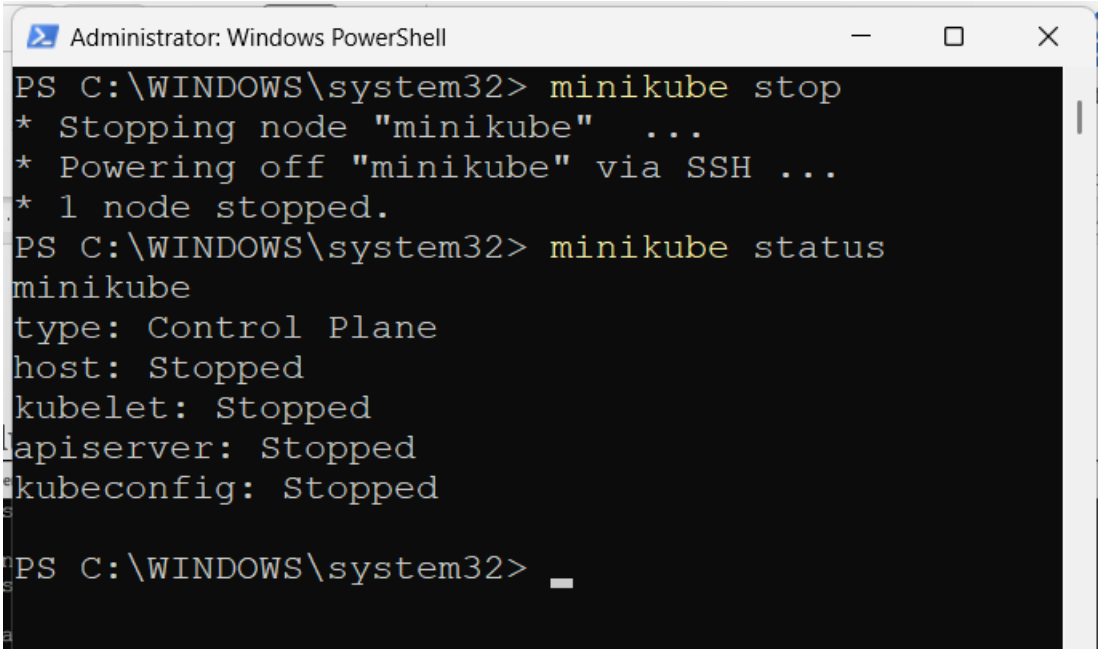
22. **To pause the cluster, run minikube pause, followed by minikube status**



23. **To resume the cluster, run minikube unpause, followed by minikube status**



24. **To stop the cluster, run minikube stop, followed by minikube status**



25. **To list the event log**