

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.

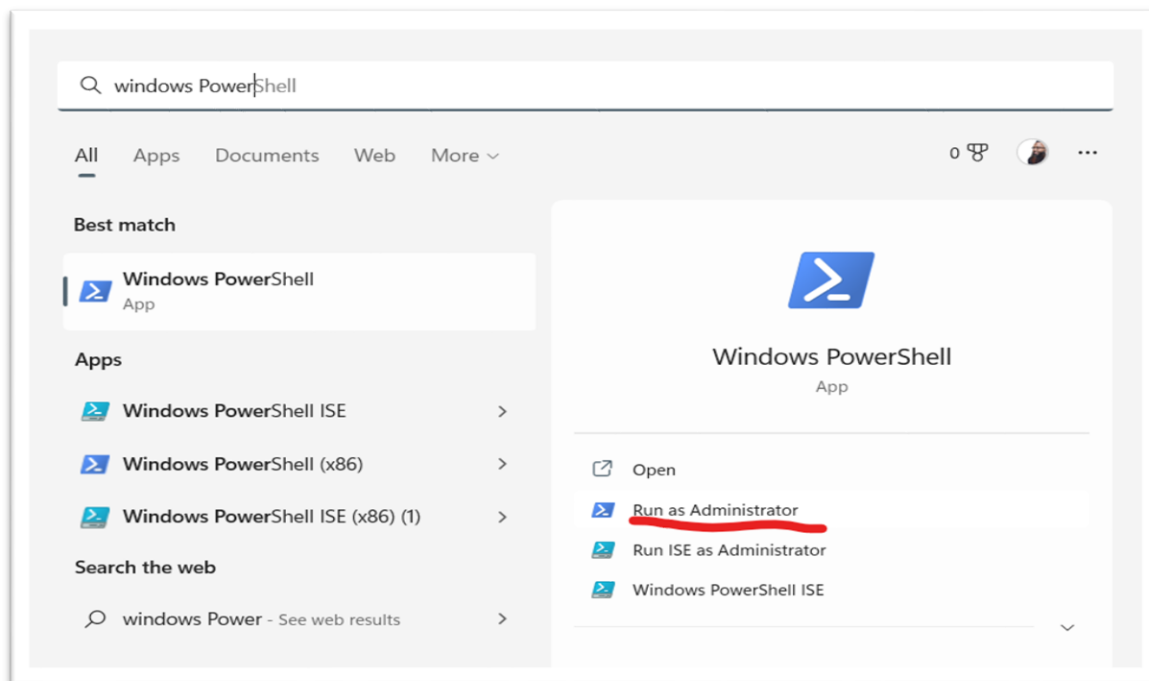
In a nutshell, Minikube is a one node Kubernetes cluster, which runs on your laptop.

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

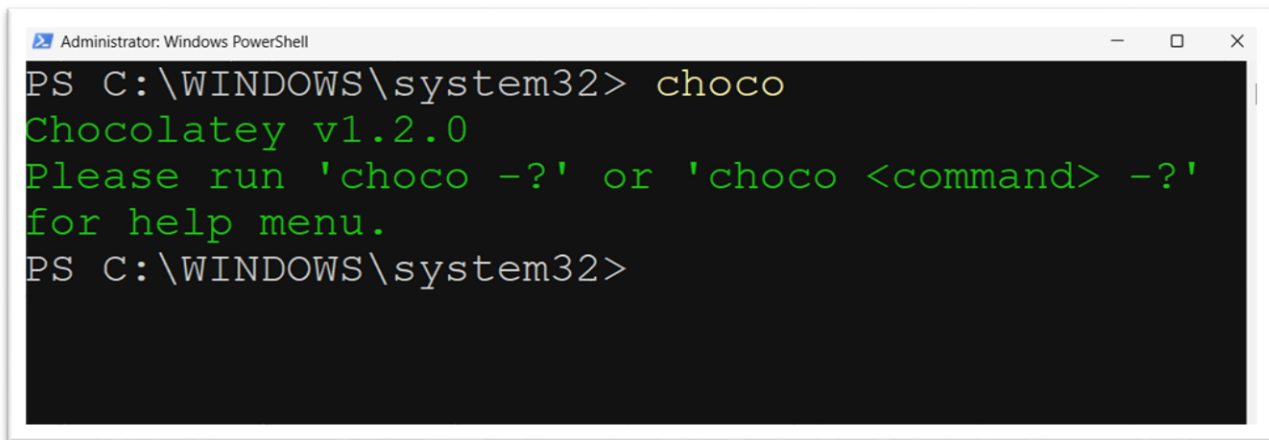
```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

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('https://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/1.2.0.
Downloading https://community.chocolatey.org/api/v2/package/chocolatey/1.2.0 to C:\Users\mzafa\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip
```

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



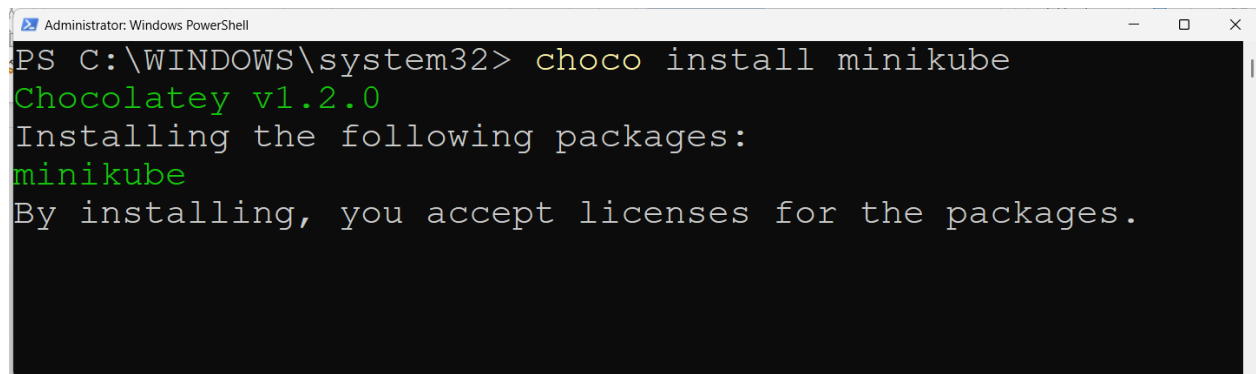
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> choco
Chocolatey v1.2.0
Please run 'choco -?' or 'choco <command> -?'
for help menu.
PS C:\WINDOWS\system32>
```

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
```



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> choco install minikube
Chocolatey v1.2.0
Installing the following packages:
minikube
By installing, you accept licenses for the packages.
```

When asked give permission as **A**

```
PS C:\WINDOWS\system32> choco install minikube
Chocolatey v1.2.0
Installing the following packages:
minikube
By installing, you accept licenses for the packages.
Progress: Downloading kubernetes-cli 1.25.3... 100%
Progress: Downloading Minikube 1.27.1... 100%

kubernetes-cli v1.25.3 [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): Y/A

Timeout or your choice of 'Y/A' is not a valid selection.
You must select an answer
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): A
```

```
s-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 kubect1-convert.exe
ShimGen has successfully created a shim for kubect1.exe
The install of kubernetes-cli was successful.
Software installed to 'C:\ProgramData\chocolatey\lib\kubernetes-cli\tools'

Minikube v1.27.1 [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>
```

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube
minikube provisions and manages local Kubernetes clusters optimized for dev
elopment 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 minik
ube cluster
  pause          pause Kubernetes
  unpause        unpause Kubernetes

Images Commands:
  docker-env      Provides instructions to point your terminal's docker-cl
i 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
```

A verbose screen is evidence of successful installation of minikube.

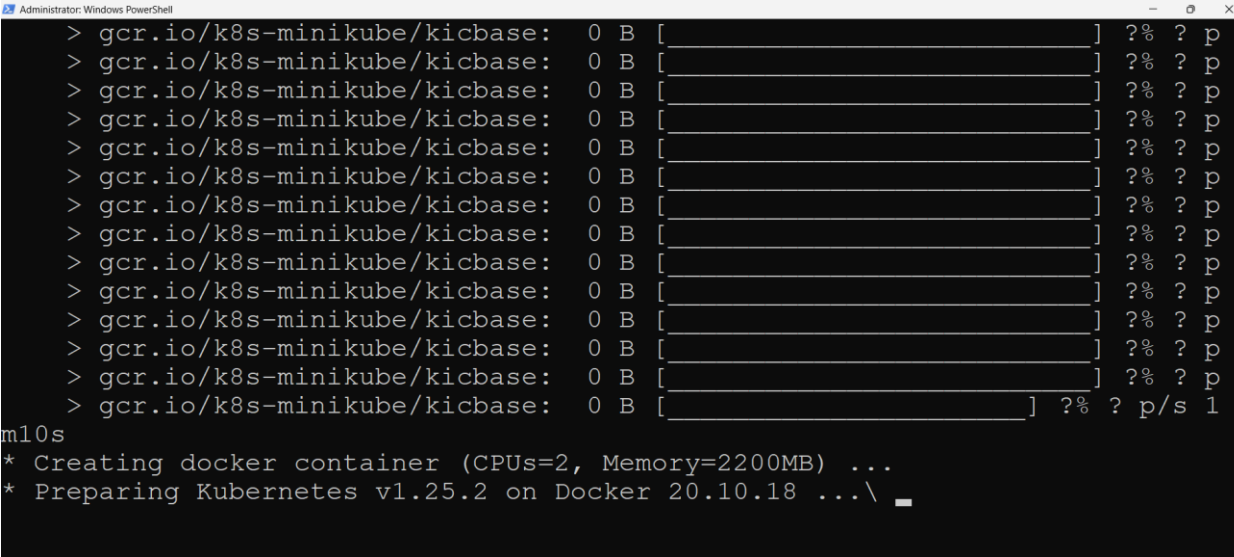
You have installed a usecase cluster on Kubernetes of your system.

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube start
* minikube v1.27.1 on Microsoft Windows 11 Home 10.0.25231 Build 25231
* 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 v1.25.2 preload ...
> gcr.io/k8s-minikube/kicbase: 0 B [ ] 0% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] 0% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] 0% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] 0% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] 0% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] 0% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] 0% ? p
> gcr.io/k8s-minikube/kicbase: 27.32 KiB / 387.11 MiB [> ] 0.01% ? p
> gcr.io/k8s-minikube/kicbase: 27.32 KiB / 387.11 MiB [> ] 0.01% ? p
> preloaded-images-k8s-v18-v1...: 14.85 KiB / 385.41 MiB [> ] 0.00% ? p
> gcr.io/k8s-minikube/kicbase: 27.32 KiB / 387.11 MiB 0.01% 45.08 KiB
> preloaded-images-k8s-v18-v1...: 62.84 KiB / 385.41 MiB [> ] 0.02% ? p
> gcr.io/k8s-minikube/kicbase: 27.32 KiB / 387.11 MiB 0.01% 45.08 KiB
> preloaded-images-k8s-v18-v1...: 110.84 KiB / 385.41 MiB [ ] 0.03% ? p
> gcr.io/k8s-minikube/kicbase: 27.32 KiB / 387.11 MiB 0.01% 45.08 KiB
```

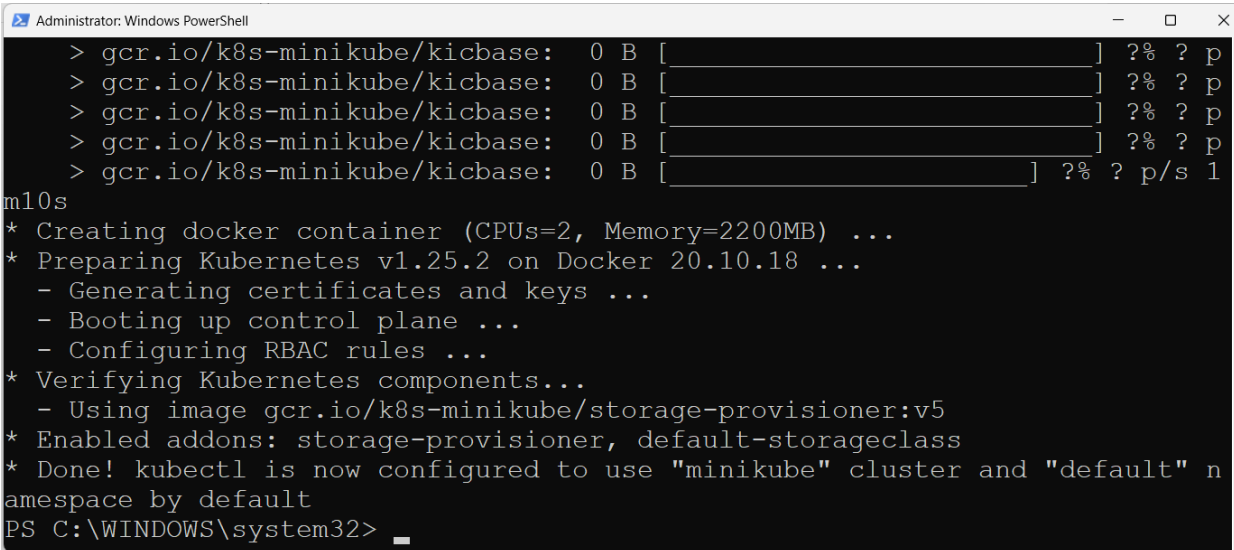
Minikube will start downloading required packages and libraries, depending upon network speed, it may take a few to several minutes.

Once download is complete, cluster preparation will begin



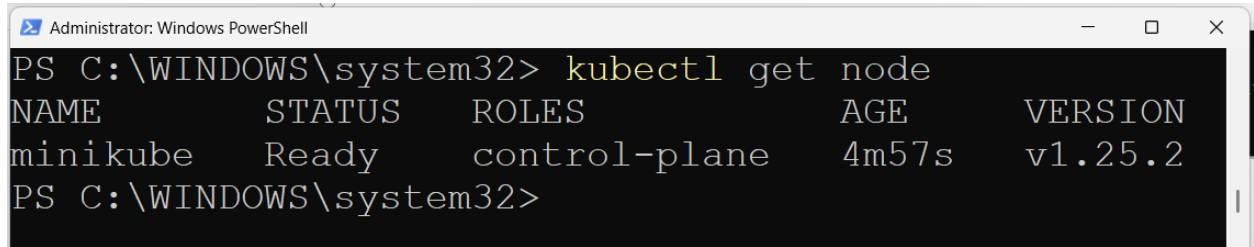
```
Administrator: Windows PowerShell
> gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p
> gcr.io/k8s-minikube/kicbase: 0 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/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



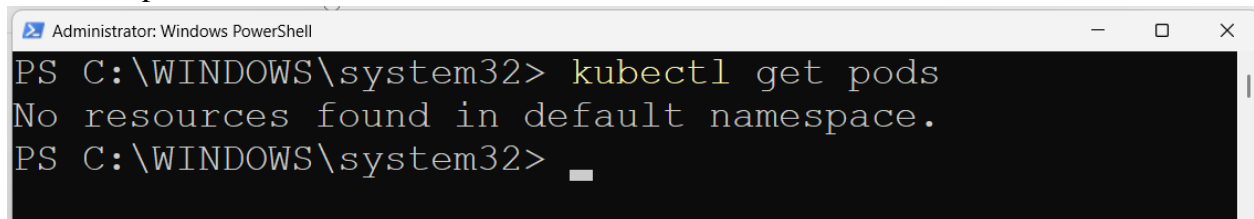
```
Administrator: Windows PowerShell
> gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p
> gcr.io/k8s-minikube/kicbase: 0 B [ ] ?% ? p/s 1
m10s
* Creating docker container (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" namespace by default
PS C:\WINDOWS\system32>
```

- Now check the number of nodes in the Kubernetes, with kubectl



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get node
NAME          STATUS    ROLES    AGE     VERSION
minikube      Ready    control-plane  4m57s   v1.25.2
PS C:\WINDOWS\system32>
```

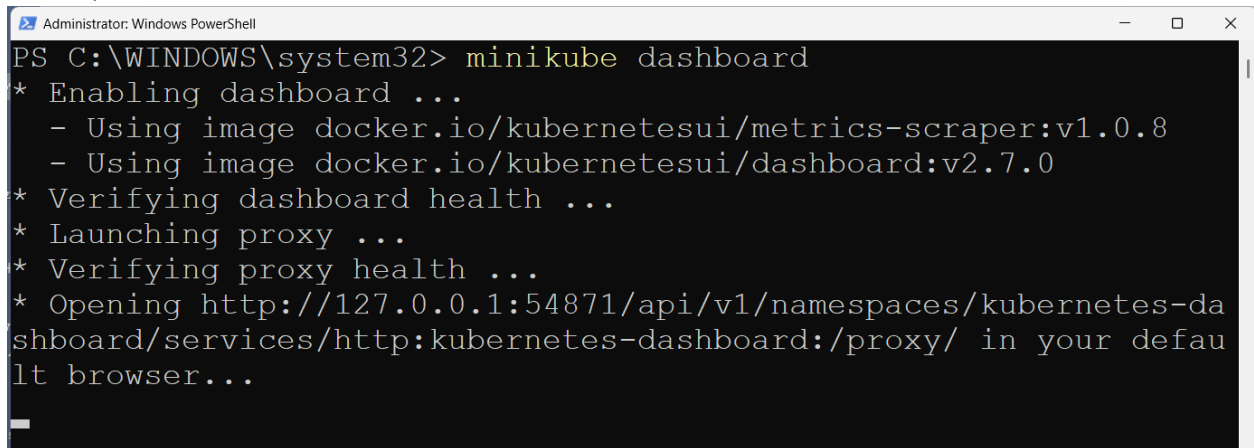
- Check for pods



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

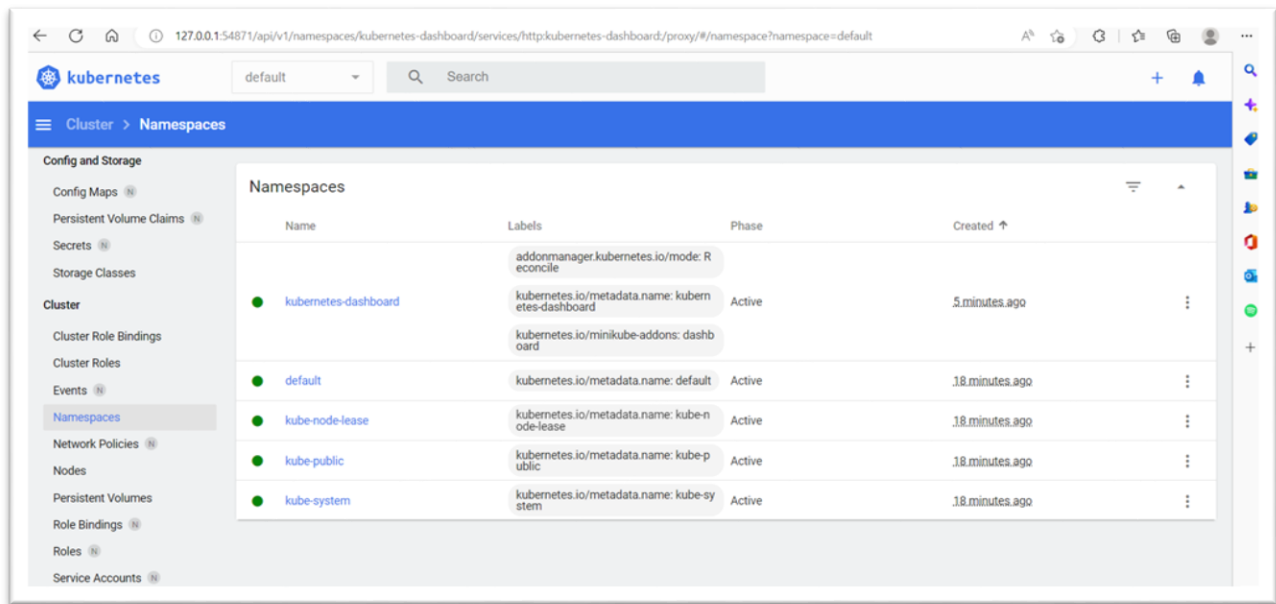
which means there are no active pods in the deployment

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



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube dashboard
* Enabling dashboard ...
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:54871/api/v1/namespaces/kubernetes-da
shboard/services/http:kubernetes-dashboard:/proxy/ in your defau
lt browser...
█
```

- 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
```

```
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>
```

9. Let's check the cluster again for the deployment, we have just created with following command,

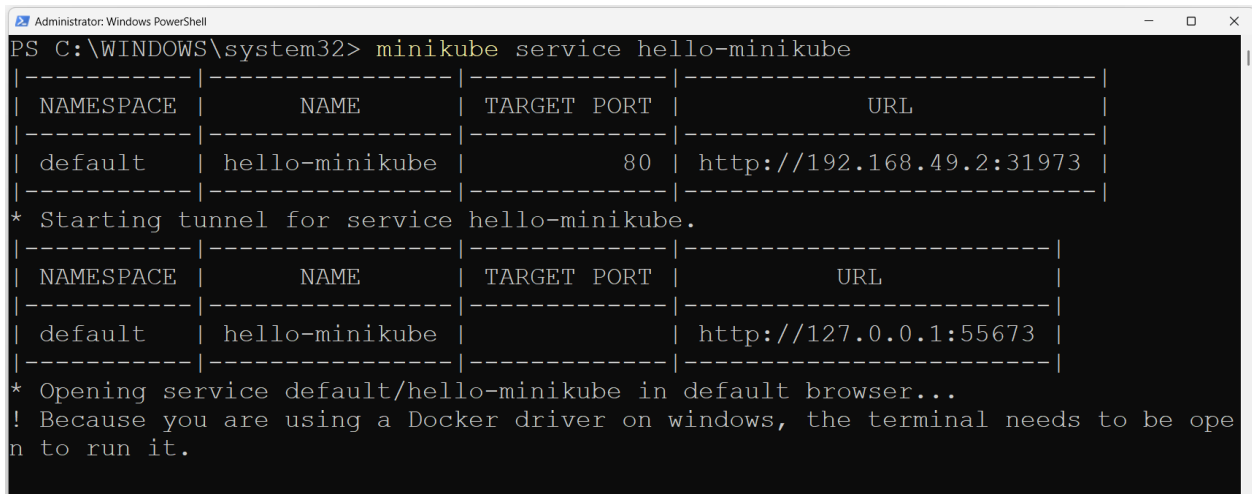
```
kubectl get services hello-minikube
```

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get services hello-minikube
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
hello-minikube NodePort 10.110.142.58 <none> 80:31973/TCP 117s
PS C:\WINDOWS\system32>
```

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**



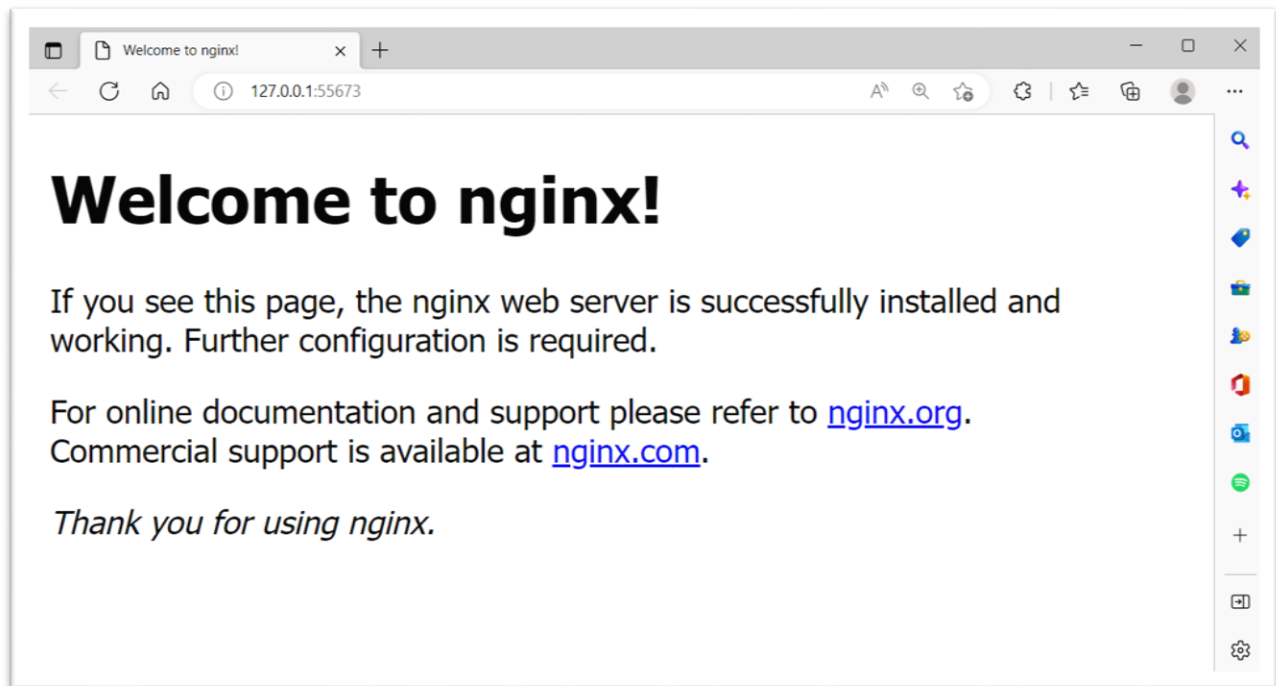
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube service hello-minikube
-----|-----|-----|-----|
NAMESPACE	NAME	TARGET PORT	URL
default	hello-minikube	80	http://192.168.49.2:31973
-----|-----|-----|-----|
* Starting tunnel for service hello-minikube.
-----|-----|-----|-----|
| NAMESPACE | NAME | TARGET PORT | URL |
-----|-----|-----|-----|
| default | hello-minikube | | http://127.0.0.1:55673 |
-----|-----|-----|-----|
* 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.

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
```

A screenshot of a Windows PowerShell terminal window. The window title is "Administrator: Windows PowerShell". The terminal shows the following commands and their outputs:  
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>

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube tunnel
* Tunnel successfully started

* NOTE: Please do not close this terminal as this process must stay a
live 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://minikube.sigs.k8
s.io/docs/handbook/accessing/#access-to-ports-1024-on-windows-require
s-root-permission
* Starting tunnel for service balanced.
```

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

13. To know the external IP, run minikube 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 10.97.9.136 127.0.0.1 80:30346/TCP 11m
PS C:\WINDOWS\system32>
```

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

## Some Administrative commands

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get pods
NAME READY STATUS RESTARTS AGE
balanced-59fdfb4746-89kvp 1/1 Running 0 26m
hello-minikube-65dc654df9-pssk7 1/1 Running 0 122m
PS C:\WINDOWS\system32>
```

15. To know about the nodes in the cluster

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get nodes
NAME STATUS ROLES AGE VERSION
minikube Ready control-plane 165m v1.25.2
PS C:\WINDOWS\system32>
```

## 16. To know about the namespaces

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get namespaces
NAME STATUS AGE
default Active 151m
kube-node-lease Active 151m
kube-public Active 151m
kube-system Active 151m
kubernetes-dashboard Active 138m
PS C:\WINDOWS\system32>
```

## 17. To know about the deployments in the cluster

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get deployments
NAME READY UP-TO-DATE AVAILABLE AGE
balanced 1/1 1 1 77m
hello-minikube 1/1 1 1 172m
PS C:\WINDOWS\system32>
```

## 18. To know about the services in the cluster

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get services
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
balanced LoadBalancer 10.97.9.136 127.0.0.1 80:30346/TCP 80m
hello-minikube NodePort 10.110.142.58 <none> 80:31973/TCP 175m
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 3h23m
PS C:\WINDOWS\system32>
```

## 19. To know about the cluster

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:54618
CoreDNS is running at https://127.0.0.1:54618/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>
```

## 20. To know about the status of the cluster

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

PS C:\WINDOWS\system32> █
```

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

```
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 <command> --help" for more information about a given command.
Use "kubectl options" for a list of global command-line options (applies to all commands).
PS C:\WINDOWS\system32> █
```

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube pause
* Pausing node minikube ...
* Paused 18 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
PS C:\WINDOWS\system32> minikube status
minikube
type: Control Plane
host: Running
kubelet: Stopped
apiserver: Paused
kubeconfig: Configured
PS C:\WINDOWS\system32> _
```

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube unpause
* Unpausing node minikube ...
* Unpaused 18 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
PS C:\WINDOWS\system32> minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
PS C:\WINDOWS\system32>
```

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube stop
* Stopping node "minikube" ...
* Powering off "minikube" via SSH ...
* 1 node stopped.
PS C:\WINDOWS\system32> minikube status
minikube
type: Control Plane
host: Stopped
kubelet: Stopped
apiserver: Stopped
kubeconfig: Stopped
PS C:\WINDOWS\system32> _
```

## 25. To list the event log

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get events
LAST SEEN TYPE REASON OBJECT MESSAGE
26m Warning NodeNotReady pod/balanced-59fd4b4746-89kvp Node is not ready
11m Normal SandboxChanged pod/balanced-59fd4b4746-89kvp Pod sandbox changed, it will be killed and re-created.
10m Normal Pulled pod/balanced-59fd4b4746-89kvp Container image "docker.io/nginx:1.23" already present on machine
10m Normal Created pod/balanced-59fd4b4746-89kvp Created container nginx
10m Normal Started pod/balanced-59fd4b4746-89kvp Started container nginx
26m Warning NodeNotReady pod/hello-minikube-65dc654df9-pssk7 Node is not ready
11m Normal SandboxChanged pod/hello-minikube-65dc654df9-pssk7 Pod sandbox changed, it will be killed and re-created.
10m Normal Pulled pod/hello-minikube-65dc654df9-pssk7 Container image "docker.io/nginx:1.23" already present on machine
10m Normal Created pod/hello-minikube-65dc654df9-pssk7 Created container nginx
10m Normal Started pod/hello-minikube-65dc654df9-pssk7 Started container nginx
26m Normal NodeNotReady node/minikube Node minikube status is now: NodeNotReady
26m Normal Starting node/minikube Starting kubelet.
26m Normal NodeHasSufficientMemory node/minikube Node minikube status is now: NodeHasSufficientMemory
26m Normal NodeHasNoDiskPressure node/minikube Node minikube status is now: NodeHasNoDiskPressure
26m Normal NodeHasSufficientPID node/minikube Node minikube status is now: NodeHasSufficientPID
26m Normal NodeNotReady node/minikube Node minikube status is now: NodeNotReady
26m Normal NodeAllocatableEnforced node/minikube Updated Node Allocatable limit across pods
26m Normal NodeReady node/minikube Node minikube status is now: NodeReady
11m Normal Starting node/minikube Starting kubelet.
11m Normal NodeHasSufficientMemory node/minikube Node minikube status is now: NodeHasSufficientMemory
11m Normal NodeHasNoDiskPressure node/minikube Node minikube status is now: NodeHasNoDiskPressure
11m Normal NodeHasSufficientPID node/minikube Node minikube status is now: NodeHasSufficientPID
11m Normal NodeAllocatableEnforced node/minikube Updated Node Allocatable limit across pods
10m Normal Starting node/minikube Starting kubelet.
10m Normal RegisteredNode node/minikube Node minikube event: Registered Node minikube in Controller
PS C:\WINDOWS\system32>
```