Kubernetes

PROF. P. M. JADAV

ASSOCIATE PROFESSOR

COMPUTER ENGINEERING DEPARTMENT

FACULTY OF TECHNOLOGY

DHARMSINH DESAI UNIVERSITY, NADIAD

Content

Kubernetes (K8s) Introduction

- Kubernetes is a
  - portable, extensible, open source platform
- for managing containerized workloads and services
- Cluster orchestration system
- It facilitates both declarative configuration and automation
- Kubernetes originates from Greek, meaning helmsman or pilot
- Google open-sourced the Kubernetes project in 2014

Deployment Evolution

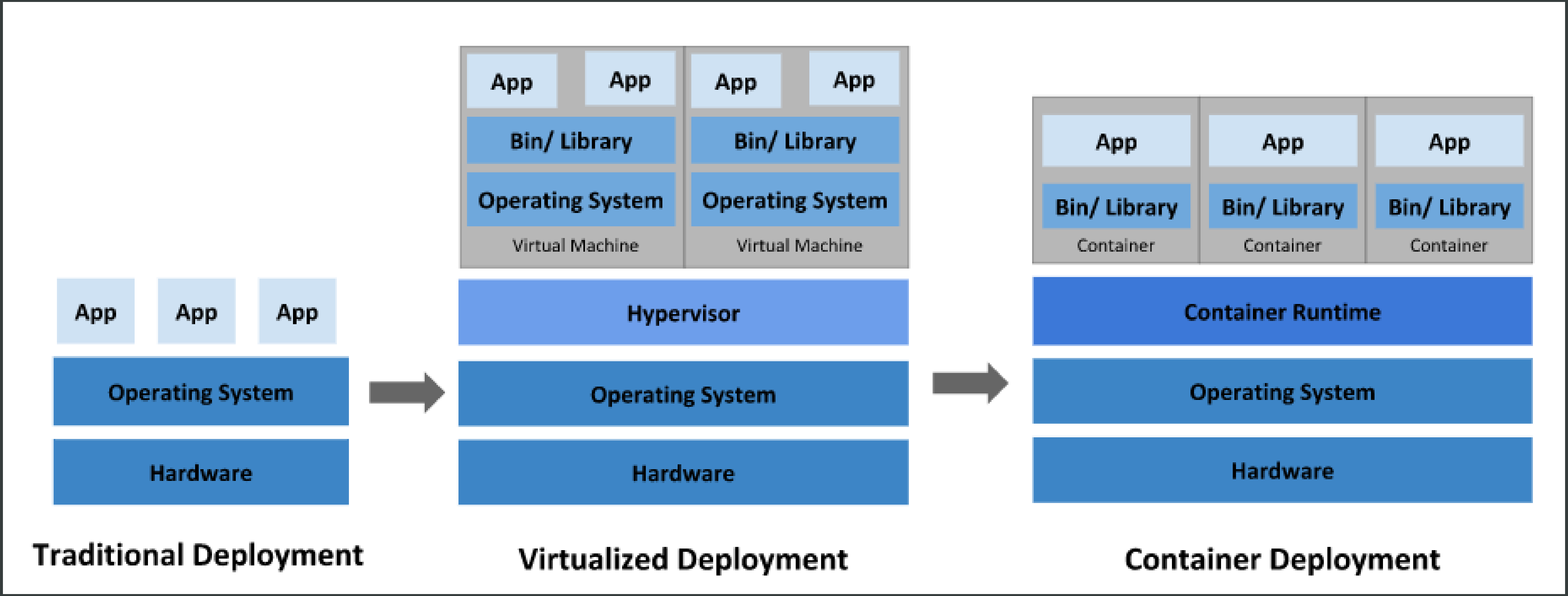


Image Source: https://kubernetes.io/docs/concepts/overview/

Why you need Kubernetes?

- Service discovery and load balancing Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.
- Storage orchestration Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.
- Automated rollouts and rollbacks You can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container.

Why you need Kubernetes? (..cont)

- Automatic bin packing You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.
- Self-healing Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.
- Secret and configuration management Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.

Basic use case of Kubernetes

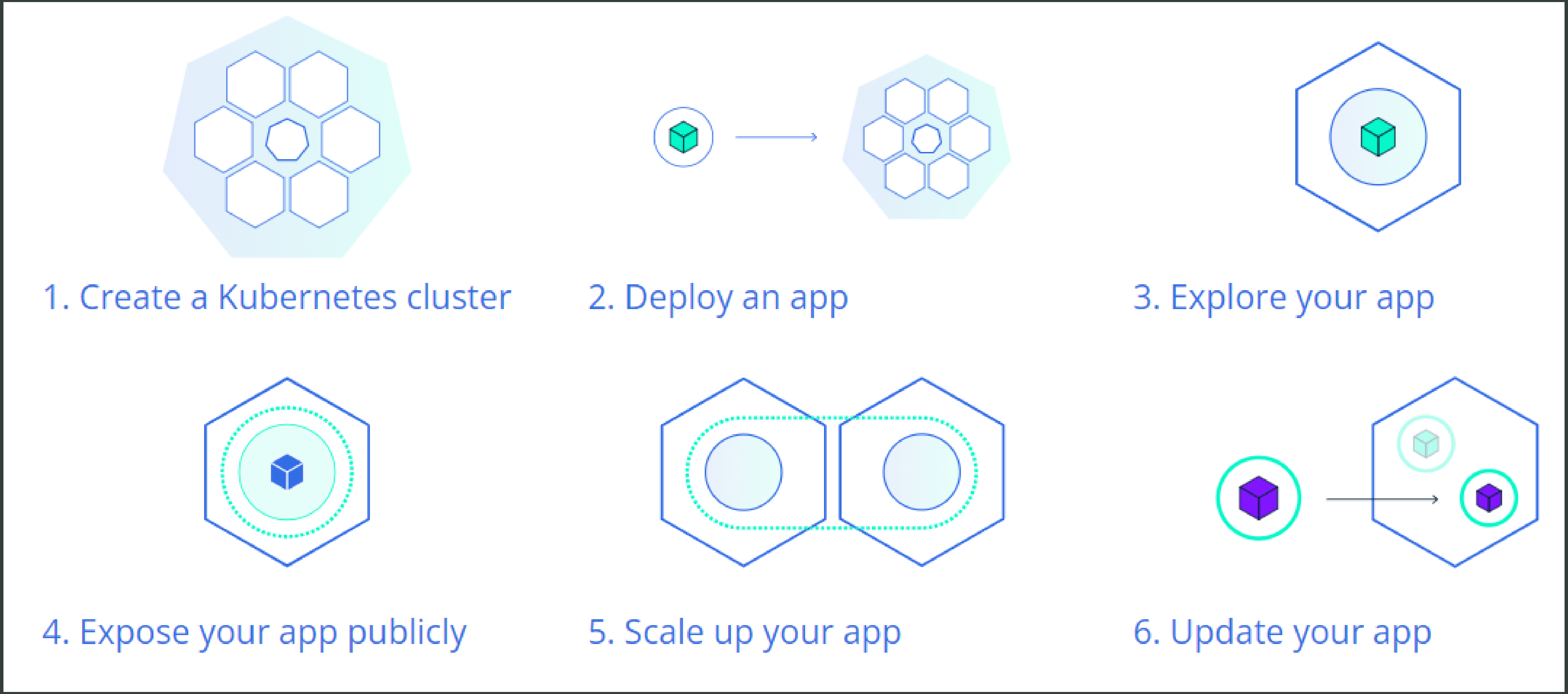


Image Source: https://kubernetes.io/docs/tutorials/kubernetes-basics/

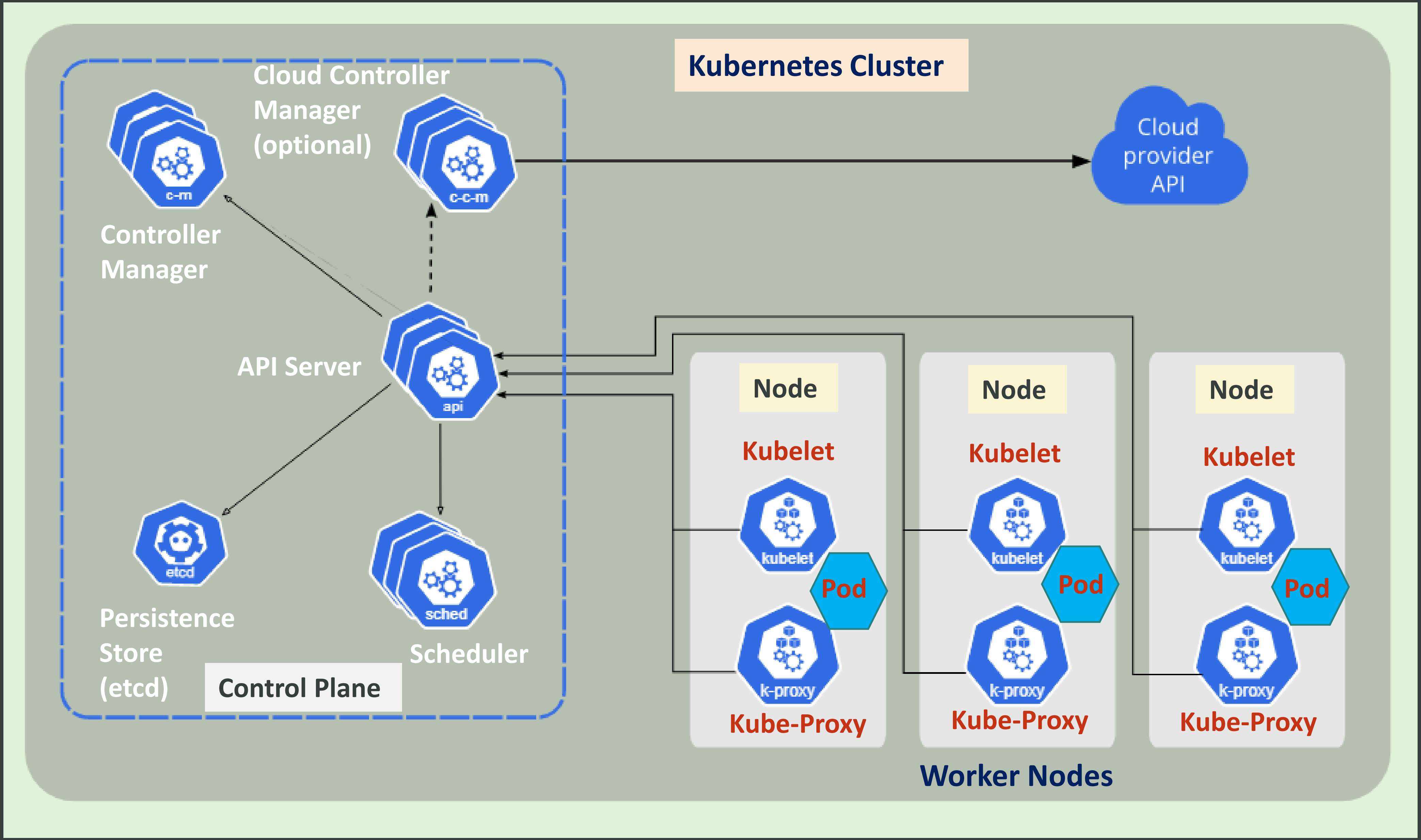


Image Source: https://kubernetes.io/docs/concepts/overview/components/

Kubernetes Components

- A Kubernetes cluster consists of a set of worker machines, called nodes, that run containerized applications
- Every cluster has at least one worker node
- The worker node(s) host the Pods that are the components of the application workload
- The control plane manages the worker nodes and the Pods in the cluster

- Note: In production environments, the control plane usually runs across multiple computers and a cluster usually runs multiple nodes, providing fault-tolerance and high availability.

Control Plane Components

1) kube-apiserver

- It exposes the Kubernetes API
- It is designed to scale horizontally—that is, it scales by deploying more instances
- You can run several instances of kube-apiserver and balance traffic between those instances

Control Plane Components

2) etcd

- Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data

- Note: If your Kubernetes cluster uses etcd as its backing store, make sure you have a back up plan for those data

Control Plane Components

3) kube-scheduler

- Assigns a node for a newly created Pod
- Factors taken into account for scheduling decisions include: individual and collective resource requirements, hardware/software/policy constraints, affinity and anti-affinity specifications, data locality, inter-workload interference, and deadlines

Control Plane Components

4) kube-controller-manager

- Logically, each controller is a separate process, but to reduce complexity, they are all compiled into a single binary and run in a single process
- Node controller: Noticing and responding when nodes go down
- Job controller: Watches for Job objects that represent one-off tasks, then creates Pods to run those tasks to completion
- EndpointSlice controller: Populates EndpointSlice objects (to provide a link between Services and Pods)
- ServiceAccount controller: Create default ServiceAccounts for new namespaces

Control Plane Components

5) cloud-controller-manager

- Embeds cloud-specific control logic
- Links your cluster into your cloud provider's API, and separates out the components that interact with that cloud platform from components that only interact with your cluster
- Only runs controllers that are specific to your cloud provider

Note:

i) Own premises or learning environment cluster does not have a cloud controller manager

ii) It combines several logically independent control loops into a single binary that you run as a single process. You can scale horizontally to improve performance or to help tolerate failures

Control Plane Components

5) cloud-controller-manager

- The following controllers can have cloud provider dependencies:
- Node controller: For checking the cloud provider to determine if a node has been deleted in the cloud after it stops responding
- Route controller: For setting up routes in the underlying cloud infrastructure
- Service controller: For creating, updating and deleting cloud provider load balancers

Node Components

1) kubelet

- An agent that runs on each node in the cluster
- It makes sure that containers are running in a Pod
- It takes a set of PodSpecs and ensures that the containers described in those PodSpecs are running and healthy

Node Components

2) kube-proxy

- kube-proxy is a network proxy that runs on each node in your cluster
- Maintains network rules which allow network communication to your Pods from network sessions inside or outside of your cluster
- Uses the OS packet filtering layer if there is one and it is available, otherwise, forwards the traffic itself

Node Components

2) Container Runtime

- Software responsible for running containers
- Kubernetes supports container runtimes such as
  - containerd, CRI-O, and
  - other implementation of the Kubernetes CRI (Container Runtime Interface)

Addons

- Addons use Kubernetes resources (DaemonSet, Deployment, etc.) to implement cluster features
- Namespaced resources for addons belong within the kube-system namespace

Addons

1) DNS

- Cluster DNS is a DNS server, in addition to the other DNS server(s) in your environment, which serves DNS records for Kubernetes services
- Containers started by Kubernetes automatically include this DNS server in their DNS searches

Addons

2) Web UI (Dashboard)

- Dashboard is a general purpose, web-based UI for Kubernetes clusters
- It allows users to manage and troubleshoot applications running in the cluster, as well as the cluster itself

Addons

3) Container Resource Monitoring

- It records generic time-series metrics about containers in a central database
- It provides a UI for browsing that data

Addons

4) Cluster Level Logging

- It is responsible for saving container logs to a central log store with search/browsing interface

Tools

1) kubectl

- The Kubernetes command-line tool, kubectl, allows you to run commands against Kubernetes clusters
- Used to
  - deploy applications,
  - inspect and manage cluster resources, and
  - view logs.

Tools

2) kind

- Lets you run Kubernetes on your local computer
- This tool requires that you have Docker installed and configured

Tools

3) minikube

- Like kind, it lets you run Kubernetes locally
- Runs an all-in-one or a multi-node local Kubernetes cluster on your personal computer (Windows, macOS and Linux PCs)

Tools

4) kubeadm

- Used to create and manage Kubernetes clusters
- It performs the actions necessary to get a minimum viable, secure cluster up and running in a user friendly way

Demo

1) Use minikube for creating cluster
2) Use kubectl to access the cluster
3) Deploy an application
4) Manage our Cluster

Prerequisites for minikube

- 2 CPUs or more
- 2GB of free memory
- 20GB of free disk space
- Internet connection
- Container or virtual machine manager, such as: Docker, QEMU, Hyperkit, Hyper-V, KVM, Parallels, Podman, VirtualBox, or VMware Fusion/Workstation

Installing kubectl

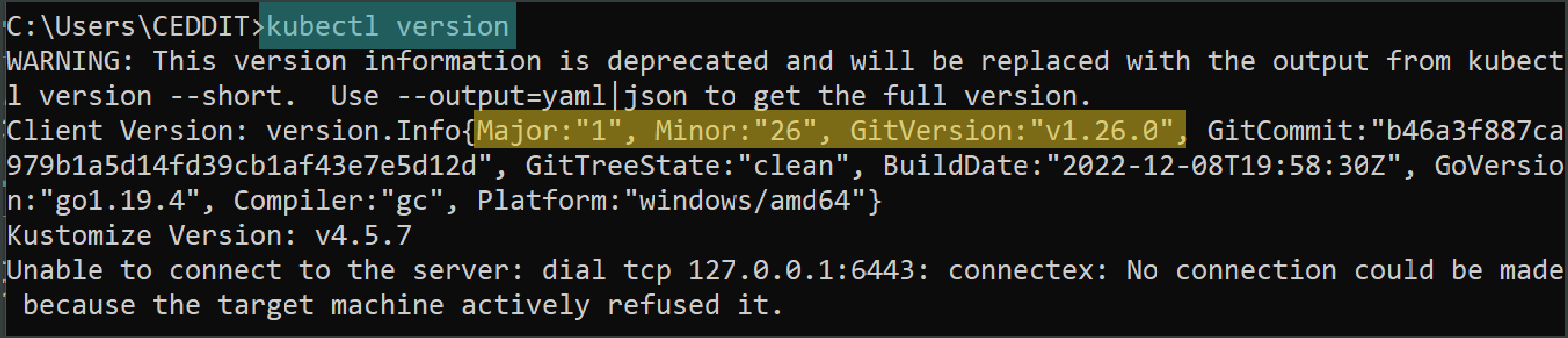
Download the latest release v1.26.0.

1) Create a directory “kubectl” and download the kubectl.exe using curl command:

```
curl.exe -LO
"https://dl.k8s.io/release/v1.26.0/bin/windows/amd64/kubectl.exe"
```

2) Add the directory “kubectl” to the PATH variable

Checking the version of kubectl

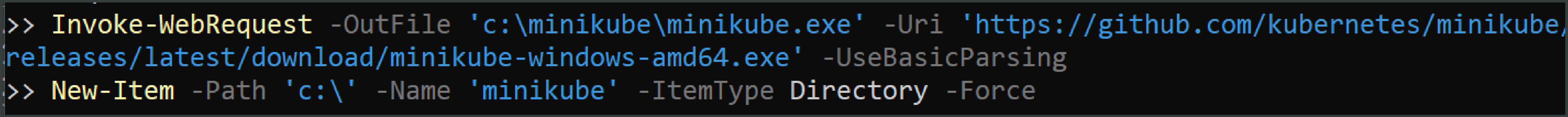


```
C:\Users\CEDDIT>kubectl version
WARNING: This version information is deprecated and will be replaced with the output from kubect
l version --short.  Use --output=yaml|json to get the full version.
Client Version: version.Info{Major:"1", Minor:"26", GitVersion:"v1.26.0", GitCommit:"b46a3f887ca
979b1a5d14fd39cb1af43e7e5d12d", GitTreeState:"clean", BuildDate:"2022-12-08T19:58:30Z", GoVersio
n:"go1.19.4", Compiler:"gc", Platform:"windows/amd64"}
Kustomize Version: v4.5.7
Unable to connect to the server: dial tcp 127.0.0.1:6443: connectex: No connection could be made
 because the target machine actively refused it.
```

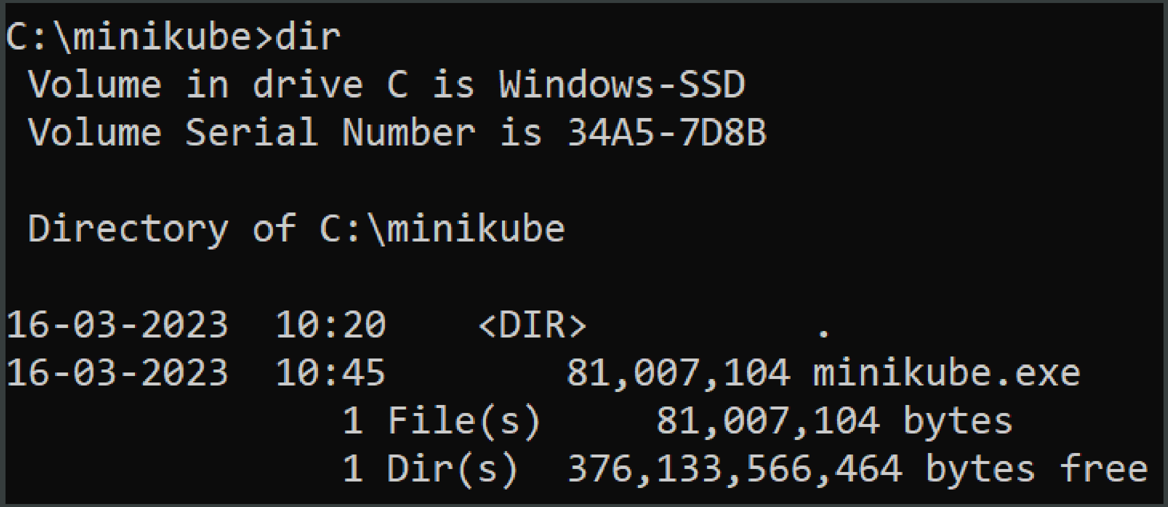
Installing minikube

Follow the installation steps given on below web page:

https://minikube.sigs.k8s.io/docs/start/



```
>> Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/minikube
releases/latest/download/minikube-windows-amd64.exe' -UseBasicParsing
>> New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force
```

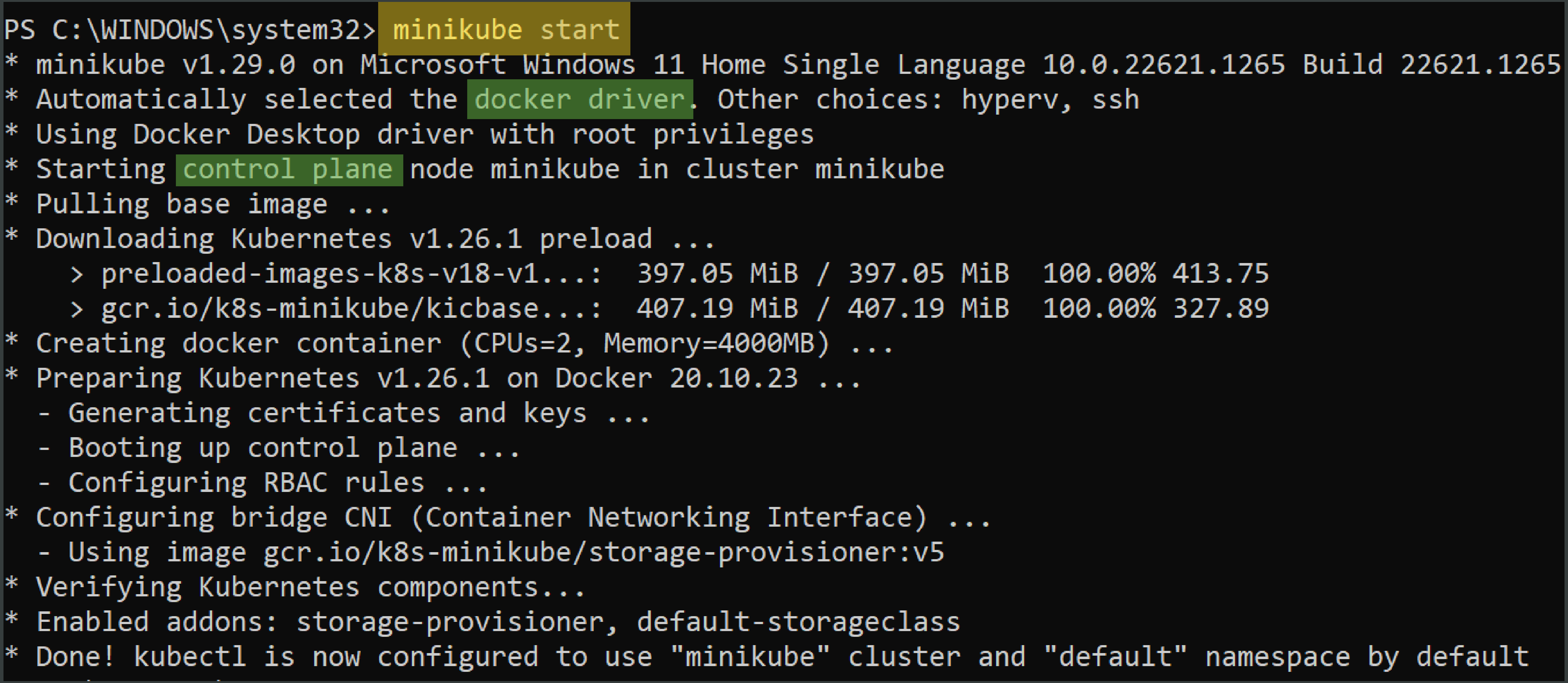


```
C:\minikube>dir
 Volume in drive C is Windows-SSD
 Volume Serial Number is 34A5-7D8B

 Directory of C:\minikube

16-03-2023  10:20    <DIR>          .
16-03-2023  10:45        81,007,104 minikube.exe
               1 File(s)     81,007,104 bytes
               1 Dir(s)  376,133,566,464 bytes free
```

Add directory minikube to the PATH variable

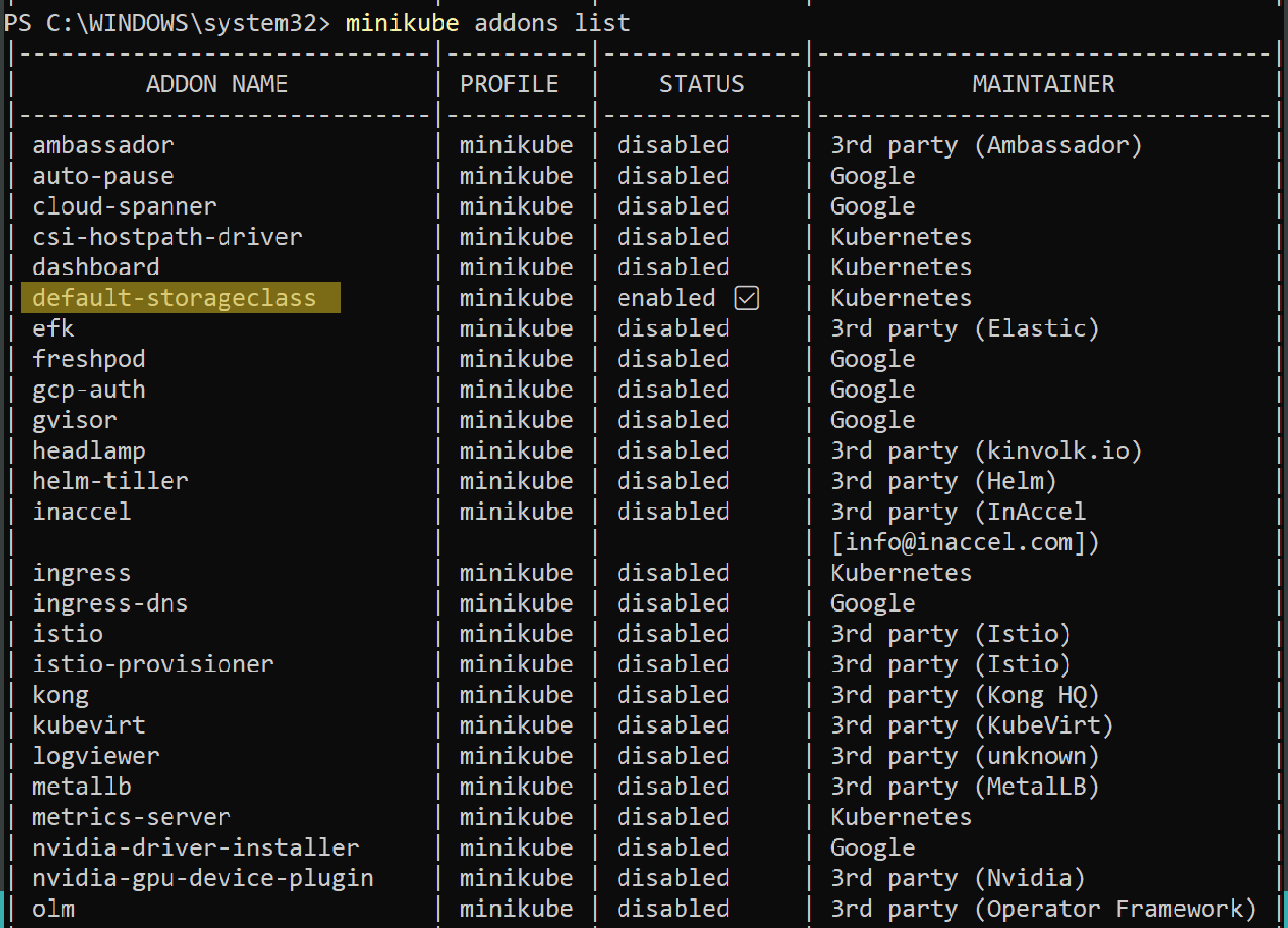
Start your Cluster

Run Powershell as Administrator



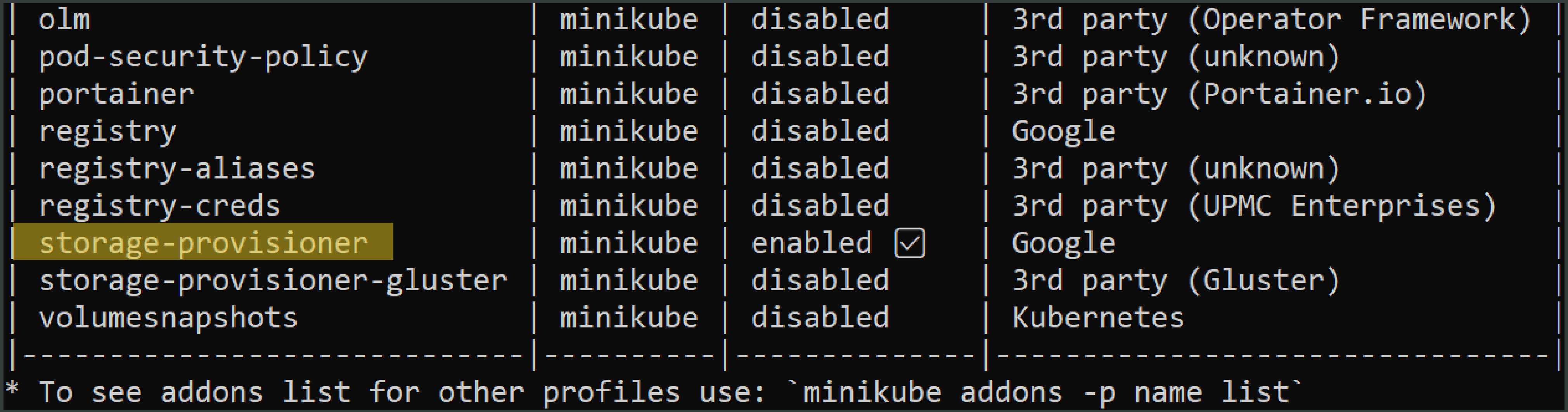
```
PS C:\WINDOWS\system32> minikube start
* minikube v1.29.0 on Microsoft Windows 11 Home Single Language 10.0.22621.1265 Build 22621.1265
* Automatically selected the docker driver. Other choices: hyperv, ssh
* Using Docker Desktop driver with root privileges
* Starting control plane node minikube in cluster minikube
* Pulling base image ...
* Downloading Kubernetes v1.26.1 preload ...
    > preloaded-images-k8s-v18-v1...:  397.05 MiB / 397.05 MiB  100.00% 413.75
    > gcr.io/k8s-minikube/kicbase...:  407.19 MiB / 407.19 MiB  100.00% 327.89
* Creating docker container (CPUs=2, Memory=4000MB) ...
* Preparing Kubernetes v1.26.1 on Docker 20.10.23 ...
  - 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
```

List the Addons



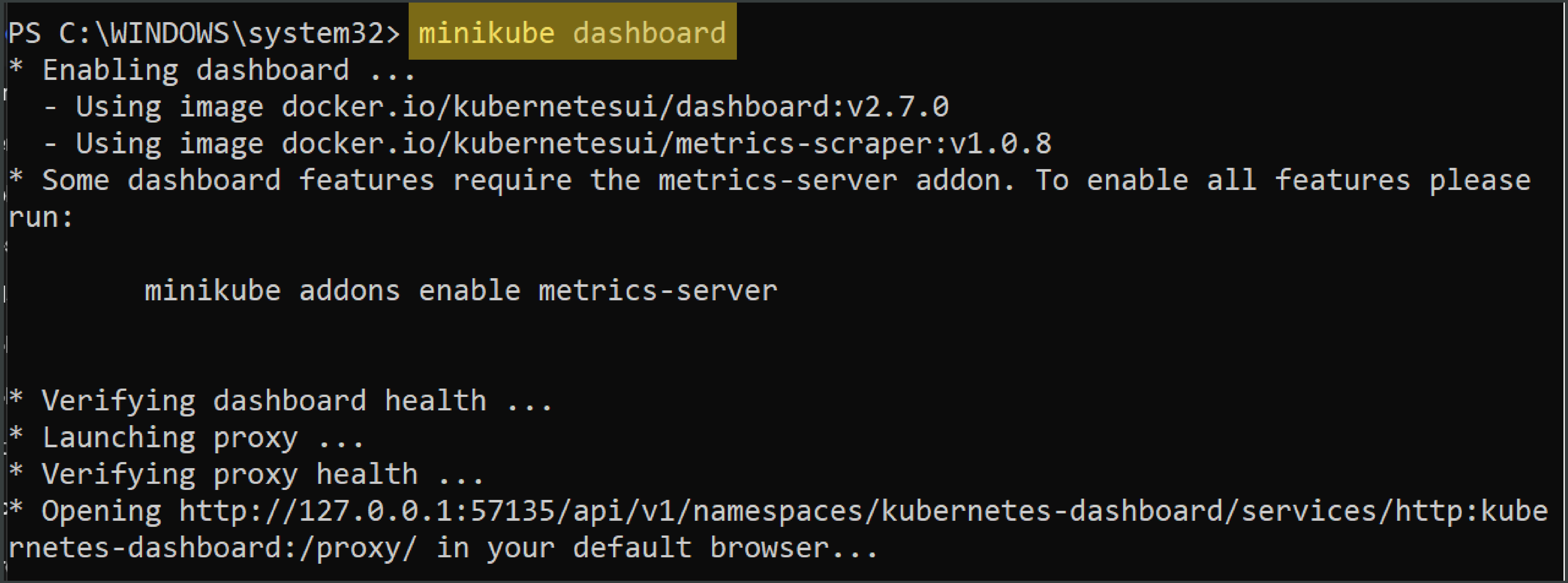
```
PS C:\WINDOWS\system32> minikube addons list
|-----------------------------|----------|-------------|--------------------------------|
|         ADDON NAME          | PROFILE  |   STATUS    |           MAINTAINER           |
|-----------------------------|----------|-------------|--------------------------------|
| ambassador                  | minikube | disabled    | 3rd party (Ambassador)         |
| auto-pause                  | minikube | disabled    | Google                         |
| cloud-spanner               | minikube | disabled    | Google                         |
| csi-hostpath-driver         | minikube | disabled    | Kubernetes                     |
| dashboard                   | minikube | disabled    | Kubernetes                     |
| default-storageclass        | minikube | enabled ✅  | Kubernetes                     |
| efk                         | minikube | disabled    | 3rd party (Elastic)            |
| freshpod                    | minikube | disabled    | Google                         |
| gcp-auth                    | minikube | disabled    | Google                         |
| gvisor                      | minikube | disabled    | Google                         |
| headlamp                    | minikube | disabled    | 3rd party (kinvolk.io)         |
| helm-tiller                 | minikube | disabled    | 3rd party (Helm)               |
| inaccel                     | minikube | disabled    | 3rd party (InAccel             |
|                             |          |             | [info@inaccel.com])            |
| ingress                     | minikube | disabled    | Kubernetes                     |
| ingress-dns                 | minikube | disabled    | Google                         |
| istio                       | minikube | disabled    | 3rd party (Istio)              |
| istio-provisioner           | minikube | disabled    | 3rd party (Istio)              |
| kong                        | minikube | disabled    | 3rd party (Kong HQ)            |
| kubevirt                    | minikube | disabled    | 3rd party (KubeVirt)           |
| logviewer                   | minikube | disabled    | 3rd party (unknown)            |
| metallb                     | minikube | disabled    | 3rd party (MetalLB)            |
| metrics-server              | minikube | disabled    | Kubernetes                     |
| nvidia-driver-installer     | minikube | disabled    | Google                         |
| nvidia-gpu-device-plugin    | minikube | disabled    | 3rd party (Nvidia)             |
| olm                         | minikube | disabled    | 3rd party (Operator Framework) |
```

List the Addons

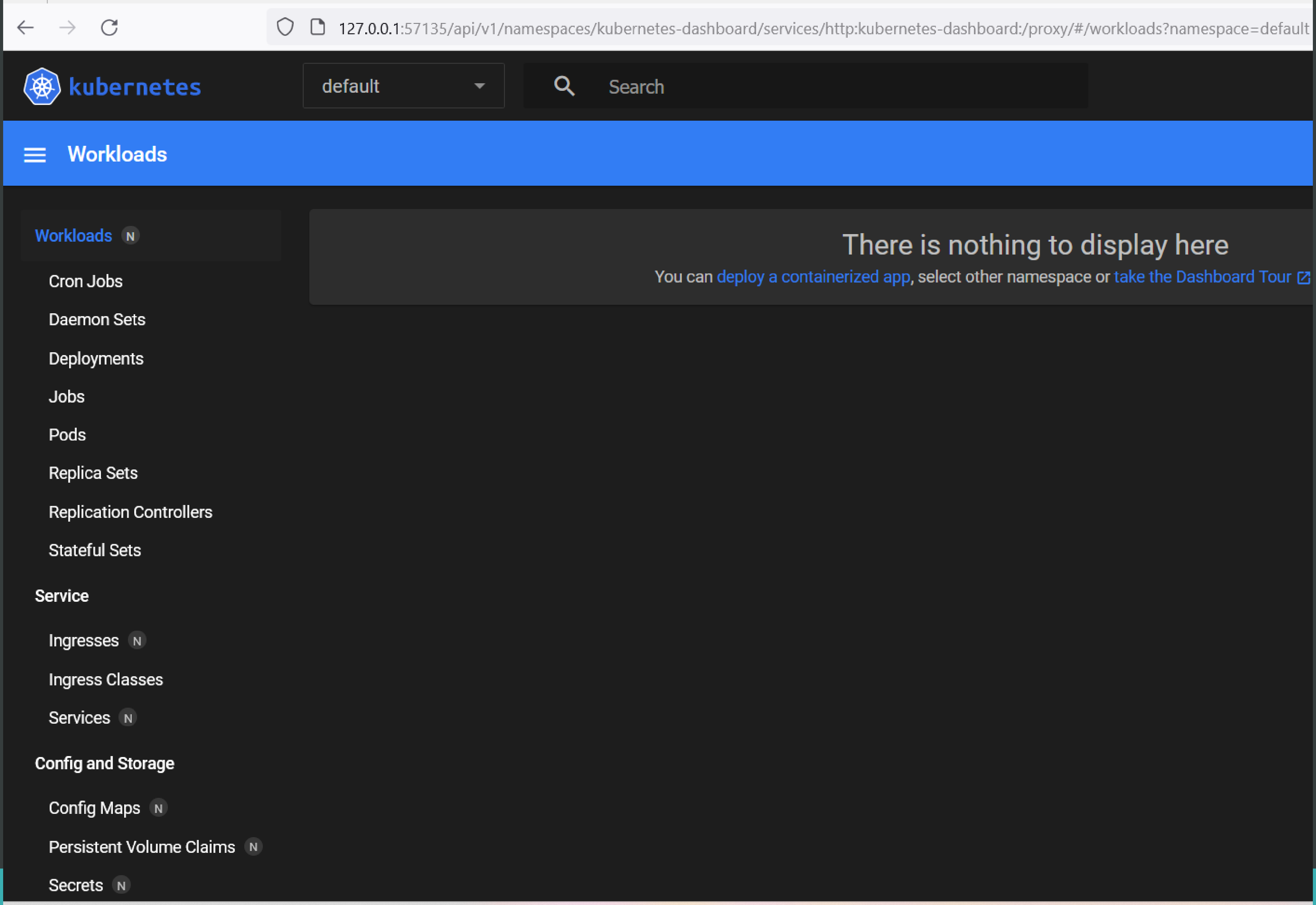


```
| olm                         | minikube | disabled     | 3rd party (Operator Framework) |
| pod-security-policy         | minikube | disabled     | 3rd party (unknown)            |
| portainer                   | minikube | disabled     | 3rd party (Portainer.io)       |
| registry                    | minikube | disabled     | Google                         |
| registry-aliases            | minikube | disabled     | 3rd party (unknown)            |
| registry-creds              | minikube | disabled     | 3rd party (UPMC Enterprises)   |
| storage-provisioner         | minikube | enabled ✓    | Google                         |
| storage-provisioner-gluster | minikube | disabled     | 3rd party (Gluster)            |
| volumesnapshots             | minikube | disabled     | Kubernetes                     |
|-----------------------------|----------|--------------|--------------------------------|
* To see addons list for other profiles use: `minikube addons -p name list`
```

Start a Dashboard



```
PS C:\WINDOWS\system32> minikube dashboard
* Enabling dashboard ...
  - Using image docker.io/kubernetesui/dashboard:v2.7.0
  - Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
* 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://127.0.0.1:57135/api/v1/namespaces/kubernetes-dashboard/services/http:kube
rnetes-dashboard:/proxy/ in your default browser...
```



127.0.0.1:57135/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/workloads?namespace=default

kubernetes

default

Search

Workloads

Workloads N

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Service

Ingresses N

Ingress Classes

Services N

Config and Storage

Config Maps N

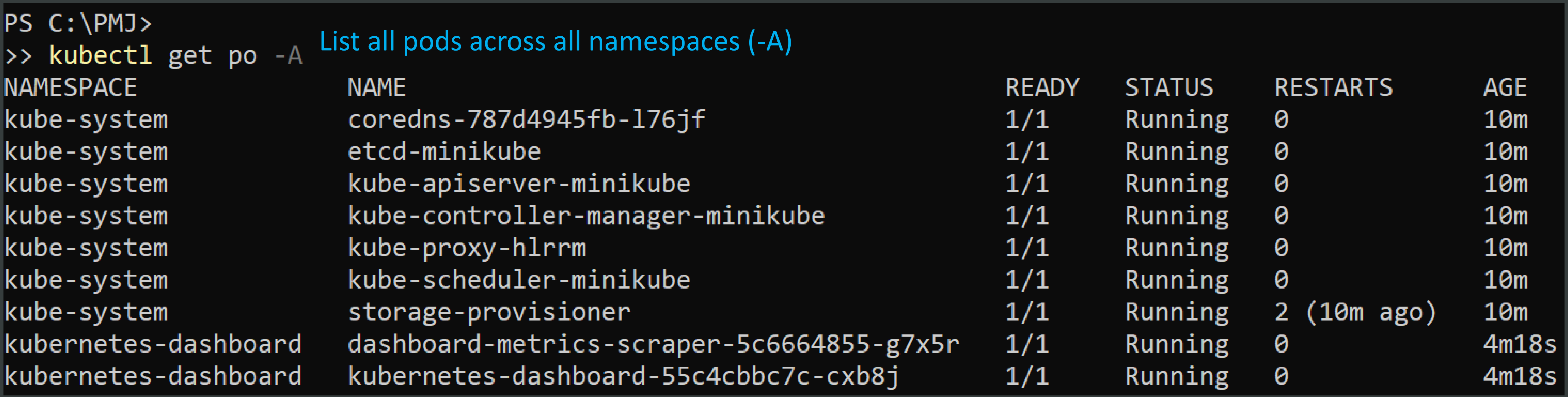
Persistent Volume Claims N

Secrets N

There is nothing to display here

You can deploy a containerized app, select other namespace or take the Dashboard Tour

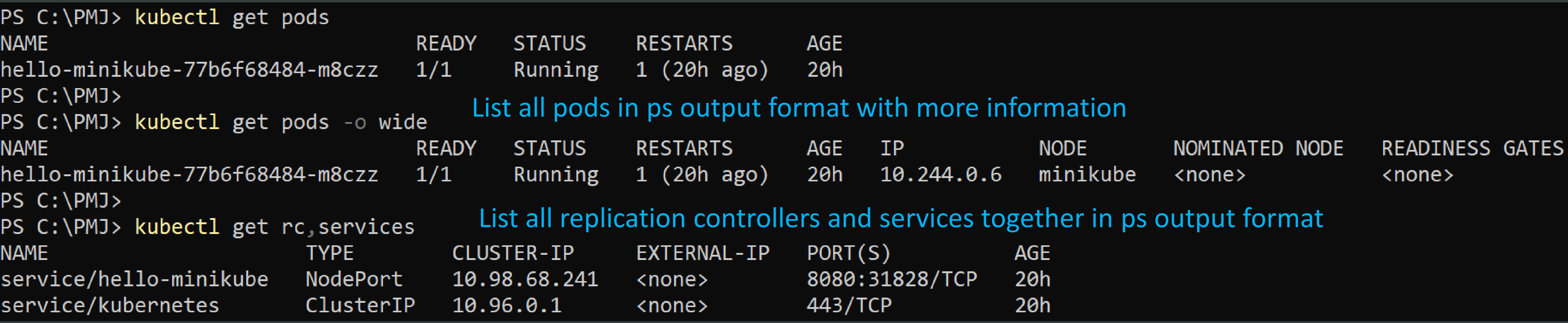
Interact with your Cluster



List all pods across all namespaces (-A)

```
PS C:\PMJ>
>> kubectl get po -A
NAMESPACE              NAME                                         READY   STATUS    RESTARTS       AGE
kube-system            coredns-787d4945fb-l76jf                     1/1     Running   0              10m
kube-system            etcd-minikube                                1/1     Running   0              10m
kube-system            kube-apiserver-minikube                      1/1     Running   0              10m
kube-system            kube-controller-manager-minikube             1/1     Running   0              10m
kube-system            kube-proxy-hlrrm                             1/1     Running   0              10m
kube-system            kube-scheduler-minikube                      1/1     Running   0              10m
kube-system            storage-provisioner                          1/1     Running   2 (10m ago)    10m
kubernetes-dashboard   dashboard-metrics-scraper-5c6664855-g7x5r    1/1     Running   0              4m18s
kubernetes-dashboard   kubernetes-dashboard-55c4cbbc7c-cxb8j        1/1     Running   0              4m18s
```

List all pods in ps output format



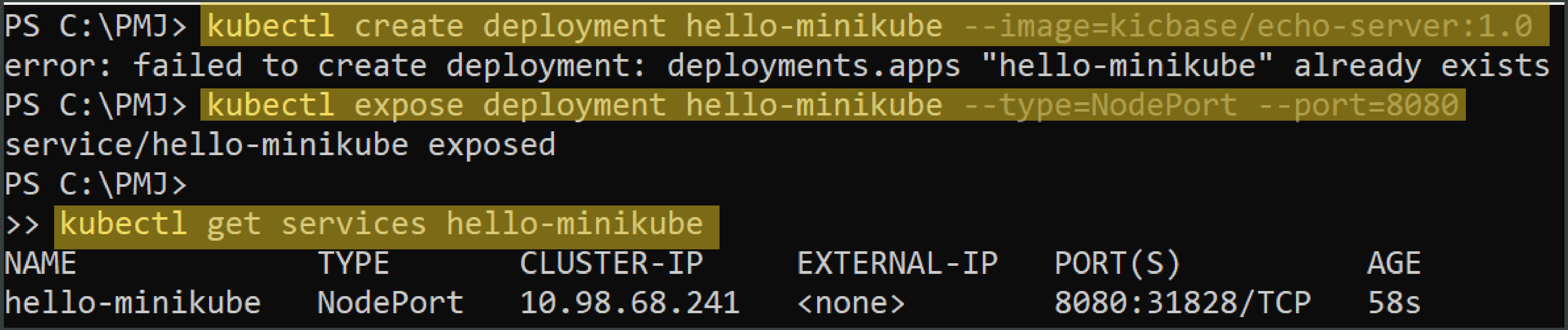
```
PS C:\PMJ> kubectl get pods
NAME                              READY   STATUS    RESTARTS       AGE
hello-minikube-77b6f68484-m8czz   1/1     Running   1 (20h ago)    20h
PS C:\PMJ>
PS C:\PMJ> kubectl get pods -o wide
NAME                              READY   STATUS    RESTARTS       AGE   IP            NODE       NOMINATED NODE   READINESS GATES
hello-minikube-77b6f68484-m8czz   1/1     Running   1 (20h ago)    20h   10.244.0.6    minikube   <none>           <none>
PS C:\PMJ>
PS C:\PMJ> kubectl get rc,services
NAME                      TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/hello-minikube    NodePort    10.98.68.241   <none>        8080:31828/TCP   20h
service/kubernetes        ClusterIP   10.96.0.1      <none>        443/TCP          20h
```

List all pods in ps output format with more information

List all replication controllers and services together in ps output format

Deploy Applications

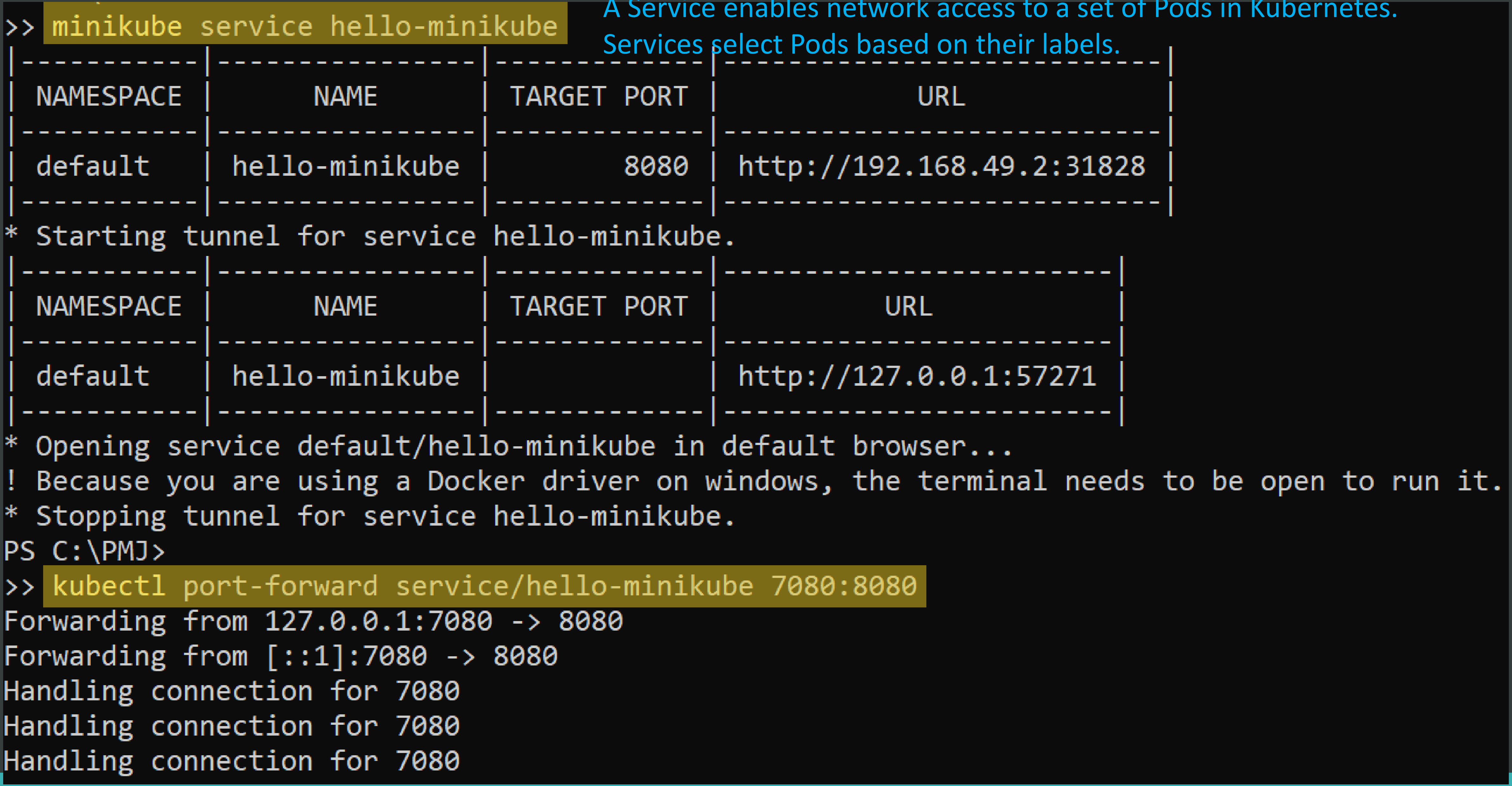
A deployment is responsible for keeping a set of pods running.



```
PS C:\PMJ> kubectl create deployment hello-minikube --image=kicbase/echo-server:1.0
error: failed to create deployment: deployments.apps "hello-minikube" already exists
PS C:\PMJ> kubectl expose deployment hello-minikube --type=NodePort --port=8080
service/hello-minikube exposed
PS C:\PMJ>
>> kubectl get services hello-minikube
NAME             TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort   10.98.68.241   <none>        8080:31828/TCP   58s
```

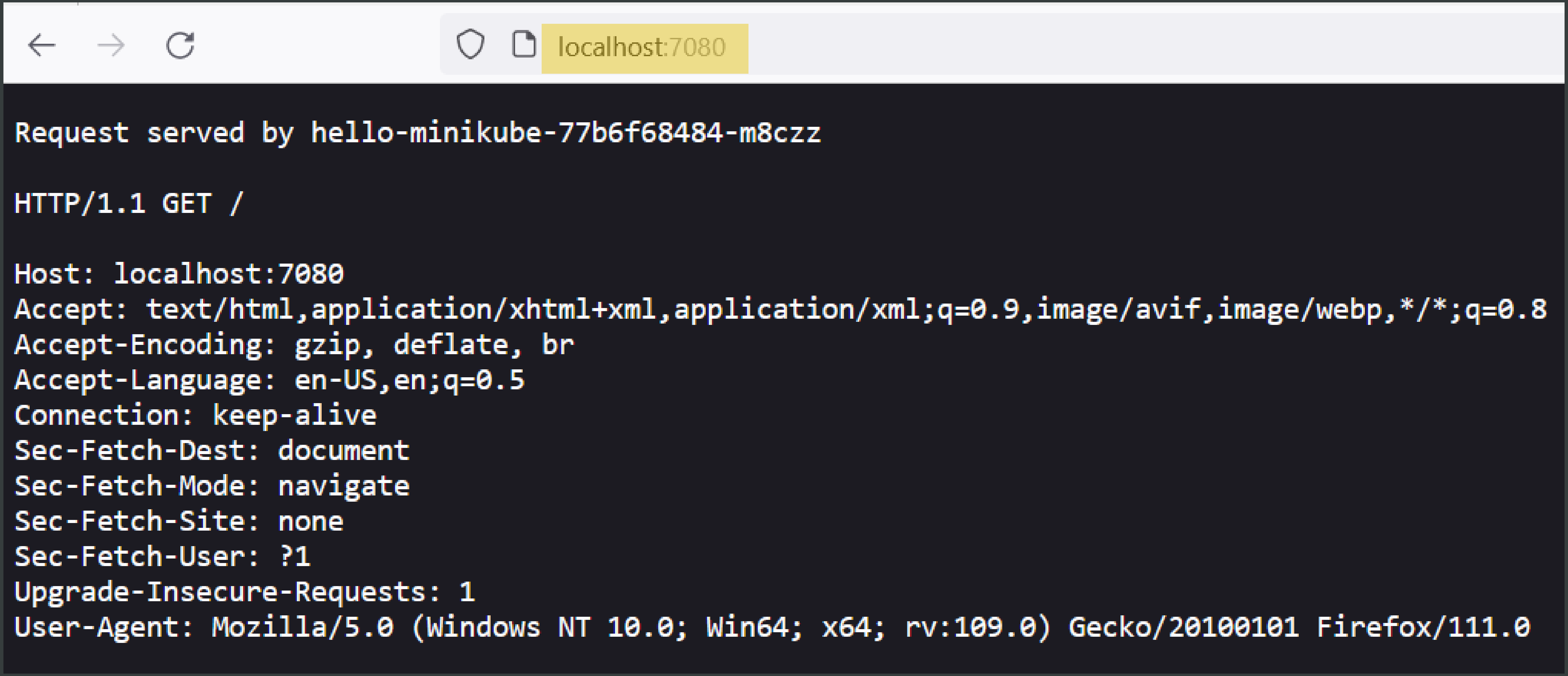
Deploy Applications

A Service enables network access to a set of Pods in Kubernetes.
Services select Pods based on their labels.



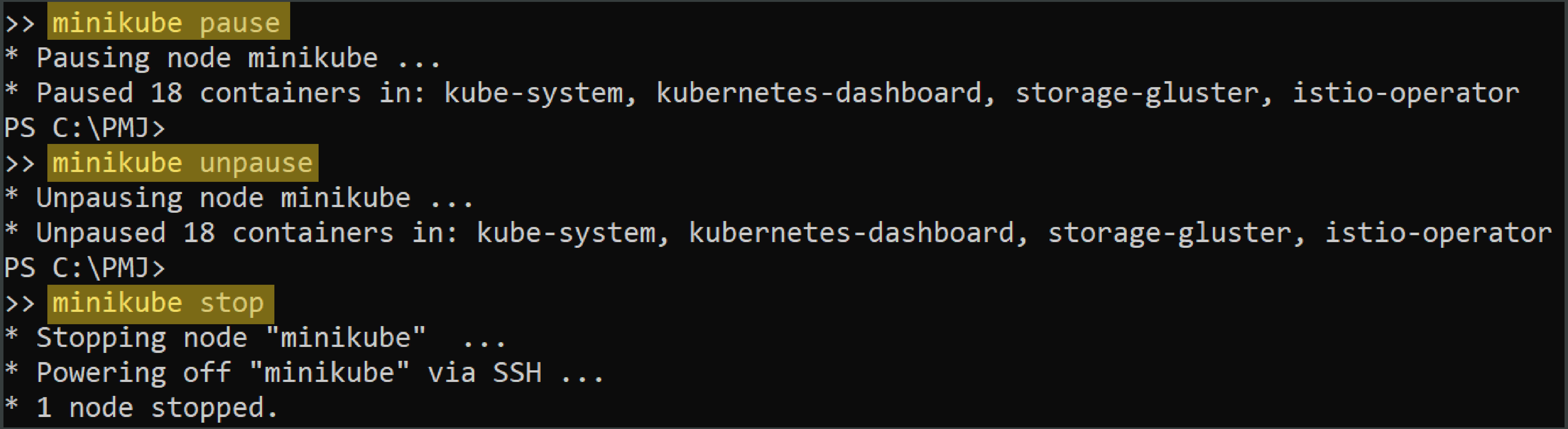
```
>> minikube service hello-minikube
|-----------|----------------|-------------|---------------------------|
| NAMESPACE |      NAME      | TARGET PORT |            URL            |
|-----------|----------------|-------------|---------------------------|
| default   | hello-minikube |        8080 | http://192.168.49.2:31828 |
|-----------|----------------|-------------|---------------------------|
* Starting tunnel for service hello-minikube.
|-----------|----------------|-------------|------------------------|
| NAMESPACE |      NAME      | TARGET PORT |          URL           |
|-----------|----------------|-------------|------------------------|
| default   | hello-minikube |             | http://127.0.0.1:57271 |
|-----------|----------------|-------------|------------------------|
* 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.
* Stopping tunnel for service hello-minikube.
PS C:\PMJ>
>> kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::1]:7080 -> 8080
Handling connection for 7080
Handling connection for 7080
Handling connection for 7080
```

Deploy Applications



```
Request served by hello-minikube-77b6f68484-m8czz

HTTP/1.1 GET /

Host: localhost:7080
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.5
Connection: keep-alive
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/111.0
```

Manage your Cluster



```
>> minikube pause
* Pausing node minikube ...
* Paused 18 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
PS C:\PMJ>
>> minikube unpause
* Unpausing node minikube ...
* Unpaused 18 containers in: kube-system, kubernetes-dashboard, storage-gluster, istio-operator
PS C:\PMJ>
>> minikube stop
* Stopping node "minikube"  ...
* Powering off "minikube" via SSH ...
* 1 node stopped.
```

References

- https://kubernetes.io/docs/concepts/overview/
- https://kubernetes.io/docs/tutorials/kubernetes-basics/
- https://kubernetes.io/docs/tasks/tools/