LAB 5

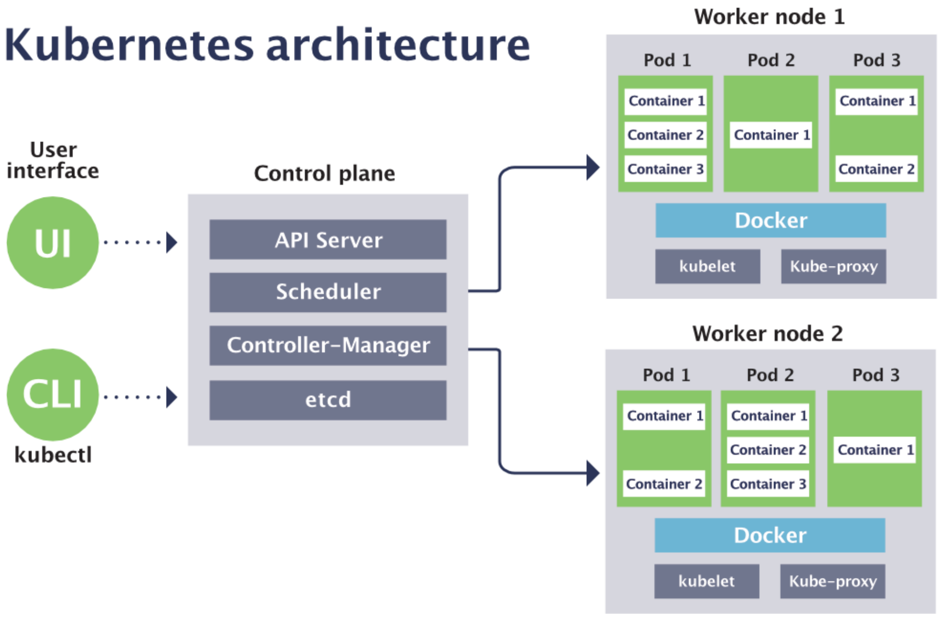
Containerized applications deployment and management using Kubernetes

Full name: Trần Đăng Khoa

Student ID: B2014926

- Note: screenshots need to be clear and good-looking; submissions must be in PDF format.

Kubernetes, also known as K8S, is an open-source system for automating deployment, scaling, and management of containerized applications. This 6-minutes video can help you quickly understand the concepts of Kubernetes. This lab provides a walkthrough of the basics of the Kubernetes cluster orchestration system



Containers: A containerized application is an application that has been packaged as one or more containers. **An image** is a static file that serves as a blueprint for creating a container. A container is a running instance of an image.

Pods: host and manage the containers that run containerized applications. A Pod can host a single container or multiple containers.

Nodes: physical or virtual machines that are used to run pods. The **master nodes** host the control plane, which is responsible for managing the state of a Kubernetes cluster. The

worker nodes are responsible for running containers. We never directly interact with the worker nodes. We send instructions to the control plane

Cluster: a Kubernetes cluster is a group of nodes used to run containerized applications

Deployment: An API object used to manage Pods to run an application workload

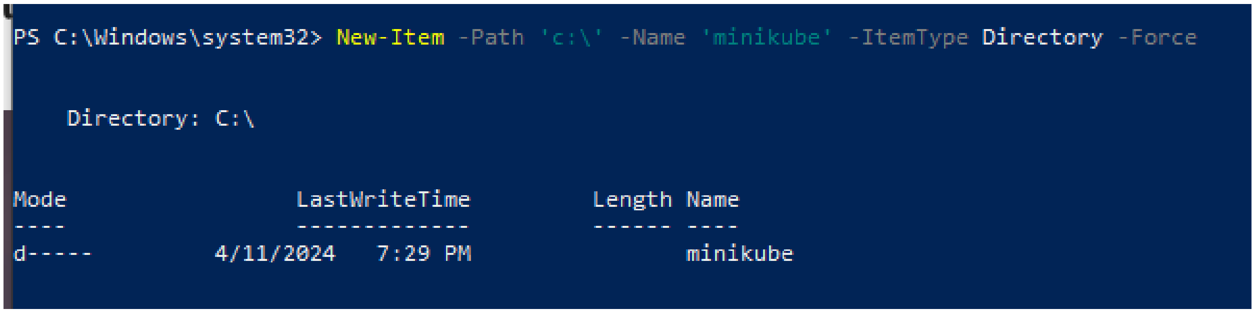
Services: A method for exposing an application that is running as one or more pods in a cluster.

1. Install minikube and kubectl on Windows OS

minikube is local Kubernetes, focusing on making it easy to learn and develop for Kubernetes. kubectl is a command line tool for communicating with a Kubernetes cluster's control plane, using the Kubernetes API.

- Download minikube and kubectl using PowerShell:

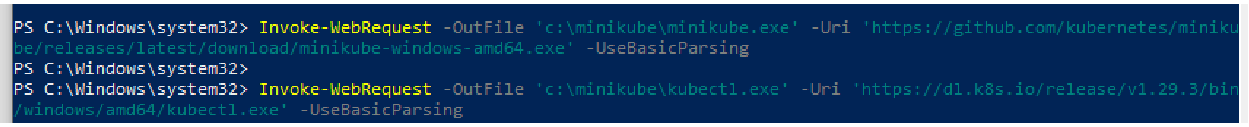
```
New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force
```



```
PS C:\Windows\system32> New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/11/2024   7:29 PM                minikube
```

```
Invoke-WebRequest    -OutFile    'c:\minikube\minikube.exe'    -Uri
'https://github.com/kubernetes/minikube/releases/latest/download/
minikube-windows-amd64.exe' -UseBasicParsing
```

```
Invoke-WebRequest    -OutFile    'c:\minikube\kubectl.exe'    -Uri
'https://dl.k8s.io/release/v1.29.3/bin/windows/amd64/kubectl.exe'
-UseBasicParsing
```



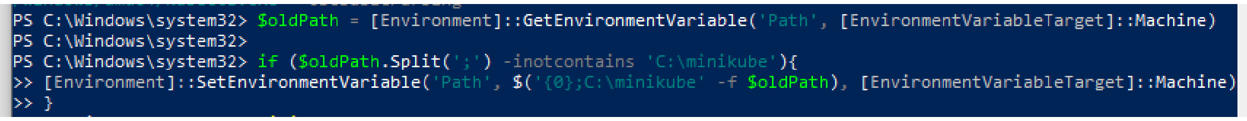
```
PS C:\Windows\system32> Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/miniku
be/releases/latest/download/minikube-windows-amd64.exe' -UseBasicParsing
PS C:\Windows\system32>
PS C:\Windows\system32> Invoke-WebRequest -OutFile 'c:\minikube\kubectl.exe' -Uri 'https://dl.k8s.io/release/v1.29.3/bin
/windows/amd64/kubectl.exe' -UseBasicParsing
```

Note: To find out the latest stable version (for example, for scripting), take a look at https://dl.k8s.io/release/stable.txt.

- Add the `minikube.exe` and `kubectl.exe` binary to your PATH (make sure to run PowerShell as Administrator):

```
$oldPath       =       [Environment]::GetEnvironmentVariable('Path',
[EnvironmentVariableTarget]::Machine)
```

```
if ($oldPath.Split(';') -inotcontains 'C:\minikube'){
[Environment]::SetEnvironmentVariable('Path', $('{0};C:\minikube'
-f $oldPath), [EnvironmentVariableTarget]::Machine)
}
```



```
PS C:\Windows\system32> $oldPath = [Environment]::GetEnvironmentVariable('Path', [EnvironmentVariableTarget]::Machine)
PS C:\Windows\system32>
PS C:\Windows\system32> if ($oldPath.Split(';') -inotcontains 'C:\minikube'){
>> [Environment]::SetEnvironmentVariable('Path', $('{0};C:\minikube' -f $oldPath), [EnvironmentVariableTarget]::Machine)
>> }
```

- Download and install VirtualBox (if necessary)

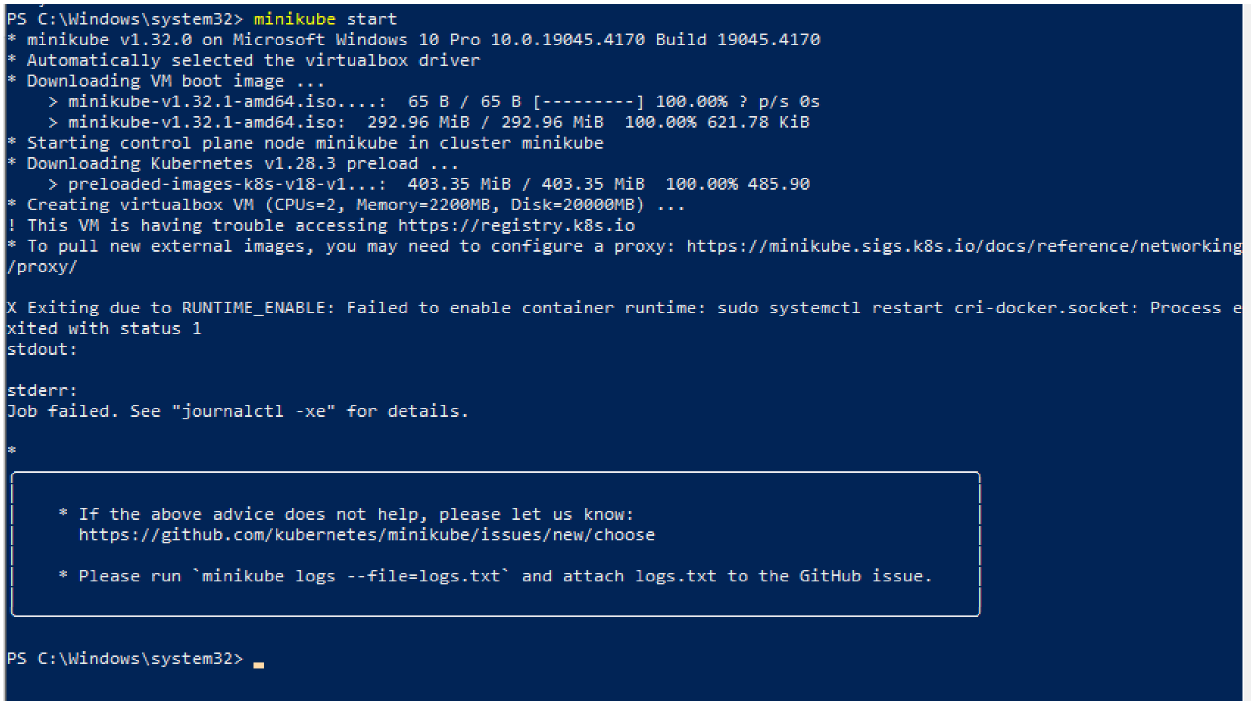
2. Learn Kubernetes Basics

2.1. Create a cluster

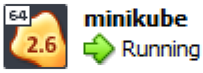
- Open Powershell with administrator privilege, using Minikube to create a cluster

```
minikube start
```

(take a screenshot)

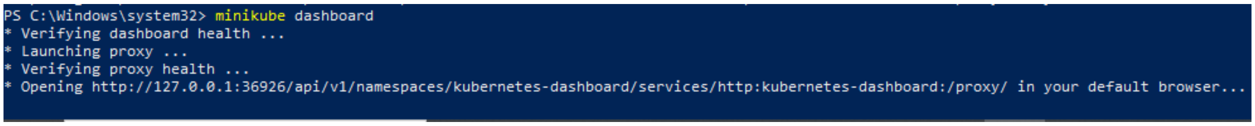


```
PS C:\Windows\system32> minikube start
* minikube v1.32.0 on Microsoft Windows 10 Pro 10.0.19045.4170 Build 19045.4170
* Automatically selected the virtualbox driver
* Downloading VM boot image ...
    > minikube-v1.32.1-amd64.iso....:  65 B / 65 B [---------] 100.00% ? p/s 0s
    > minikube-v1.32.1-amd64.iso:  292.96 MiB / 292.96 MiB  100.00% 621.78 KiB
* Starting control plane node minikube in cluster minikube
* Downloading Kubernetes v1.28.3 preload ...
    > preloaded-images-k8s-v18-v1...:  403.35 MiB / 403.35 MiB  100.00% 485.90
* Creating virtualbox VM (CPUs=2, Memory=2200MB, Disk=20000MB) ...
! This VM is having trouble accessing https://registry.k8s.io
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking
/proxy/

X Exiting due to RUNTIME_ENABLE: Failed to enable container runtime: sudo systemctl restart cri-docker.socket: Process e
xited with status 1
stdout:

stderr:
Job failed. See "journalctl -xe" for details.

*
╭──────────────────────────────────────────────────────────────────────────────────────────────╮
│                                                                                              │
│    * If the above advice does not help, please let us know:                                  │
│      https://github.com/kubernetes/minikube/issues/new/choose                                │
│                                                                                              │
│    * Please run `minikube logs --file=logs.txt` and attach logs.txt to the GitHub issue.     │
│                                                                                              │
╰──────────────────────────────────────────────────────────────────────────────────────────────╯

PS C:\Windows\system32> _
```

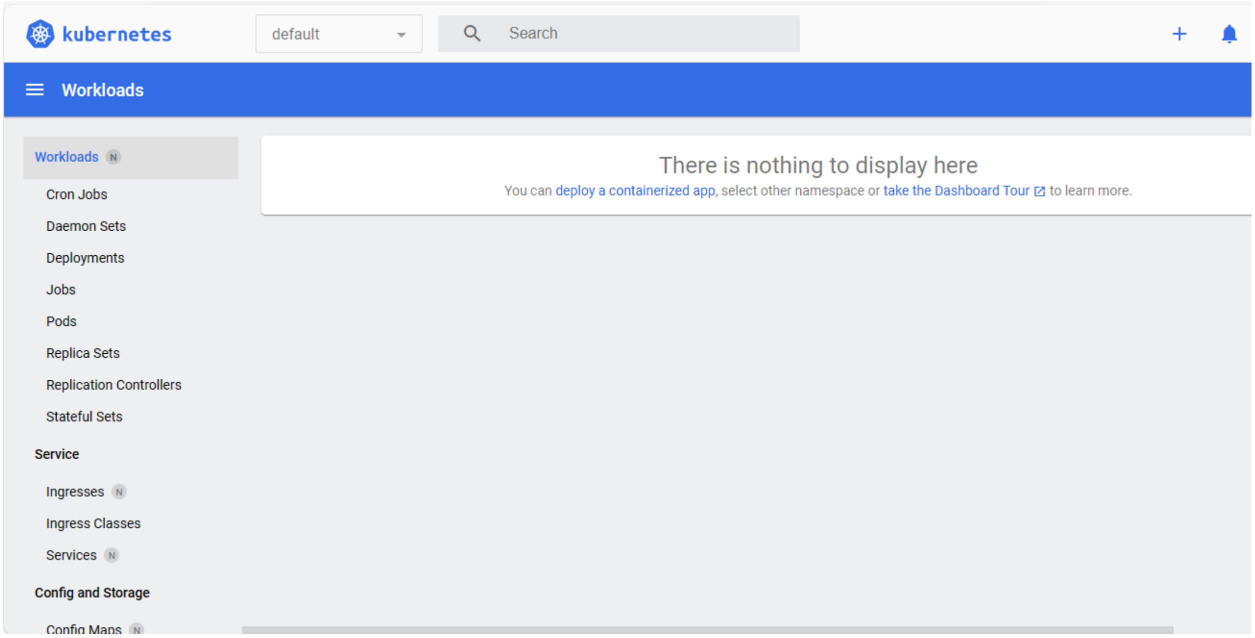


- Open the Dashboard

```
# Start a new terminal, and leave this running.
```



```
PS C:\Windows\system32> minikube dashboard
* Verifying dashboard health ...
* Launching proxy ...
* Verifying proxy health ...
* Opening http://127.0.0.1:36926/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```



There is nothing to display here

You can deploy a containerized app, select other namespace or take the Dashboard Tour to learn more.

(take a screenshot)

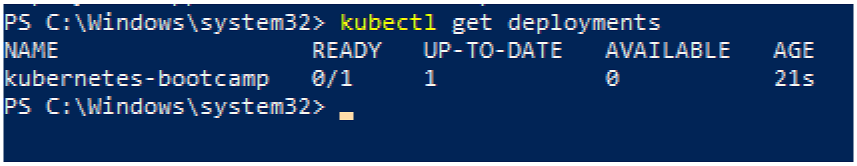
2.2. Create a Deployment

- Switch back to the terminal where you ran `minikube start`
- Using kubectl to Create a Deployment

```
kubectl create deployment kubernetes-bootcamp --image=gcr.io/google-samples/kubernetes-bootcamp:v1
```

- To list your deployments

```
kubectl get deployments
```

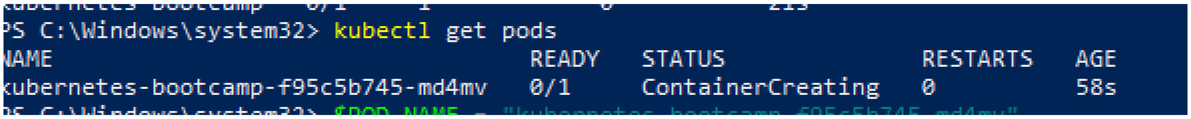


```
PS C:\Windows\system32> kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   0/1     1            0           21s
PS C:\Windows\system32>
```

(take a screenshot)

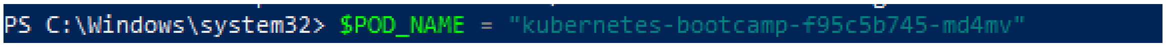
- Looking for existing Pods:

```
kubectl get pods
```



```
PS C:\Windows\system32> kubectl get pods
NAME                                   READY   STATUS              RESTARTS   AGE
kubernetes-bootcamp-f95c5b745-md4mv    0/1     ContainerCreating   0          58s
```

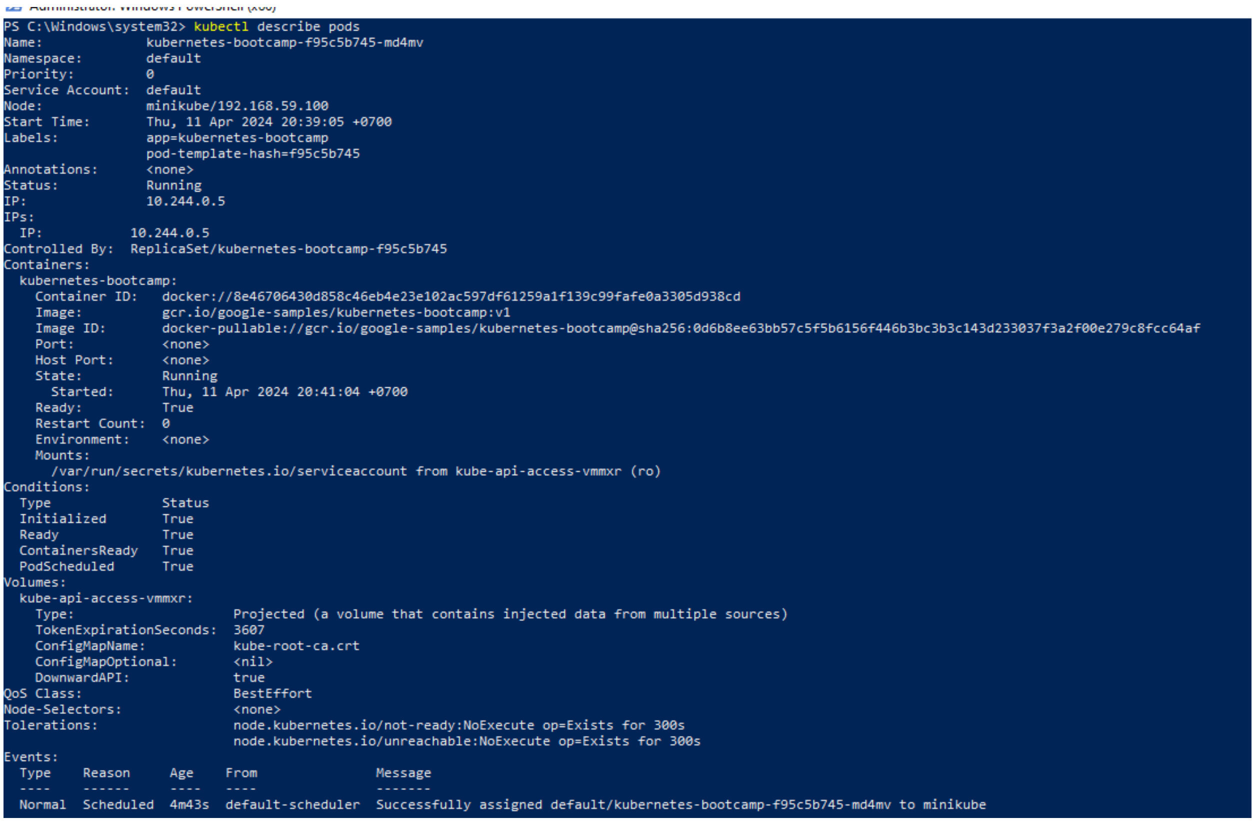
```
$POD_NAME = "<POD name>"
```



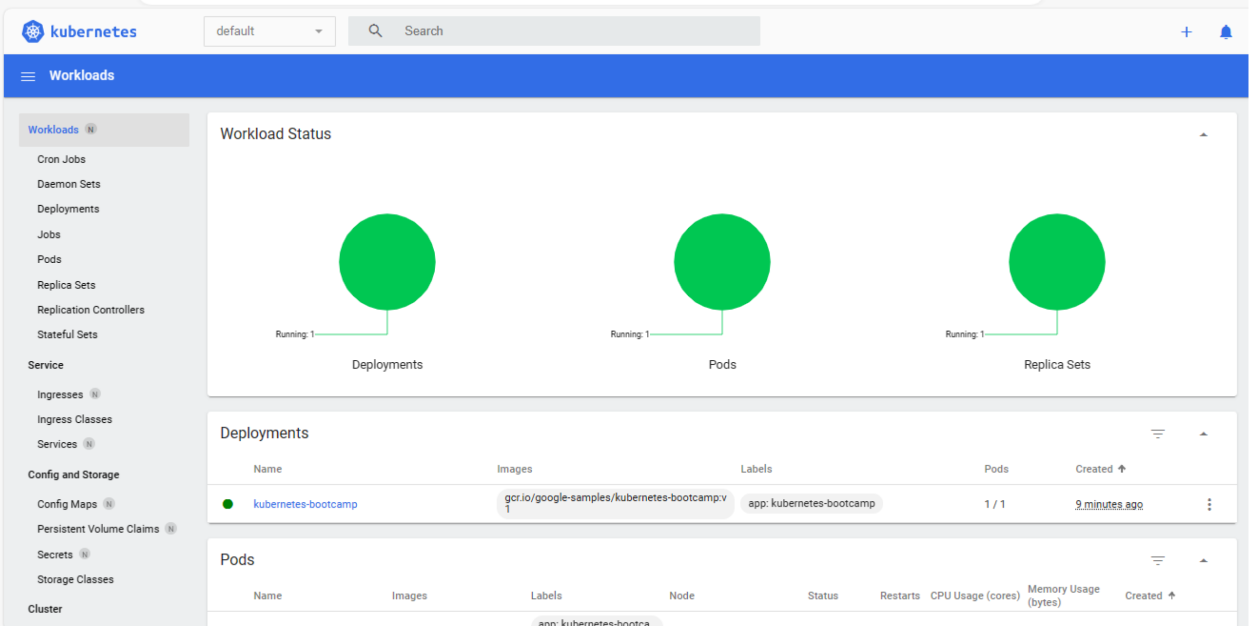
```
PS C:\Windows\system32> $POD_NAME = "kubernetes-bootcamp-f95c5b745-md4mv"
```

- To see details about the Pod's container

```
kubectl describe pods
```



```
PS C:\Windows\system32> kubectl describe pods
Name:             kubernetes-bootcamp-f95c5b745-md4mv
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.59.100
Start Time:       Thu, 11 Apr 2024 20:39:05 +0700
Labels:           app=kubernetes-bootcamp
                  pod-template-hash=f95c5b745
Annotations:      <none>
Status:           Running
IP:               10.244.0.5
IPs:
  IP:           10.244.0.5
Controlled By:  ReplicaSet/kubernetes-bootcamp-f95c5b745
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://8e46706430d858c46eb4e23e102ac597df61259a1f139c99fafe0a3305d938cd
    Image:          gcr.io/google-samples/kubernetes-bootcamp:v1
    Image ID:       docker-pullable://gcr.io/google-samples/kubernetes-bootcamp@sha256:0d6b8ee63bb57c5f5b6156f446b3bc3b3c143d233037f3a2f00e279c8fcc64af
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 11 Apr 2024 20:41:04 +0700
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-vmmxr (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-vmmxr:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:
  Type    Reason     Age    From               Message
  ----    ------     ----   ----               -------
  Normal  Scheduled  4m43s  default-scheduler  Successfully assigned default/kubernetes-bootcamp-f95c5b745-md4mv to minikube
```



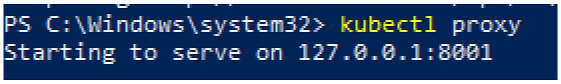
Deployments

	Name	Images	Labels	Pods	Created ↑
●	kubernetes-bootcamp	gcr.io/google-samples/kubernetes-bootcamp:v1	app: kubernetes-bootcamp	1 / 1	9 minutes ago

Pods

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
		app: kubernetes-bootca						

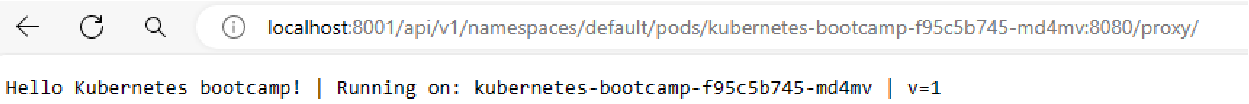
- Create a proxy that will forward communications into the cluster-wide private network
  `kubectl proxy`



```
PS C:\Windows\system32> kubectl proxy
Starting to serve on 127.0.0.1:8001
```

- Show your app in the terminal (or a web browser)
  `Curl`

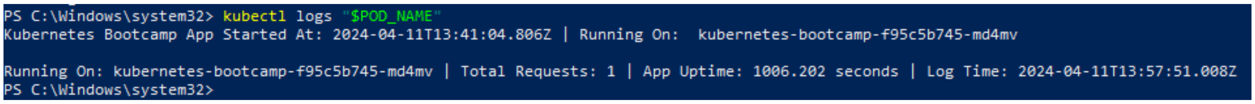
`http://localhost:8001/api/v1/namespaces/default/pods/kubernetes-bootcamp-f95c5b745-md4mv:8080/proxy/`



```
localhost:8001/api/v1/namespaces/default/pods/kubernetes-bootcamp-f95c5b745-md4mv:8080/proxy/

Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-f95c5b745-md4mv | v=1
```

- View the container logs

  `kubectl logs "$POD_NAME"`



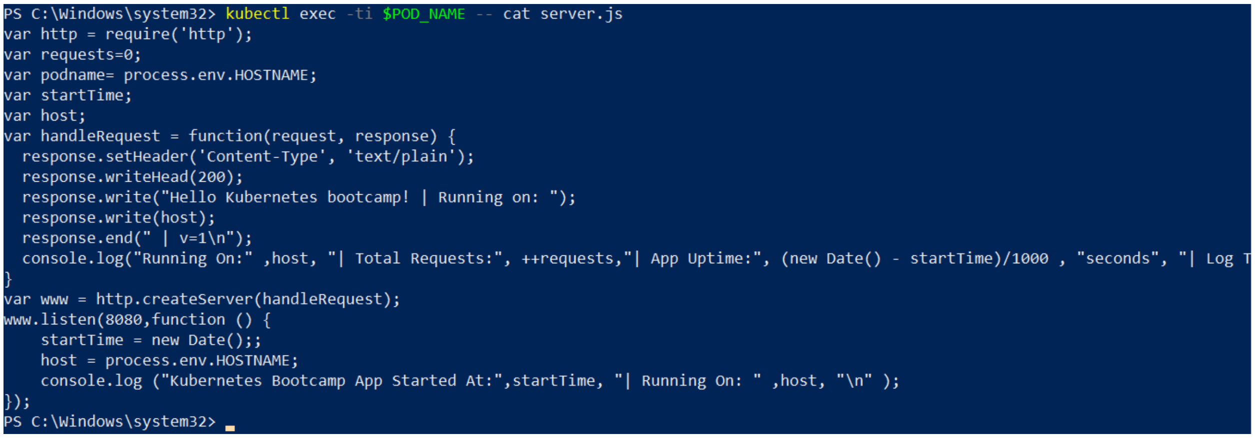
```
PS C:\Windows\system32> kubectl logs "$POD_NAME"
Kubernetes Bootcamp App Started At: 2024-04-11T13:41:04.806Z | Running On:  kubernetes-bootcamp-f95c5b745-md4mv

Running On: kubernetes-bootcamp-f95c5b745-md4mv | Total Requests: 1 | App Uptime: 1006.202 seconds | Log Time: 2024-04-11T13:57:51.008Z
PS C:\Windows\system32>
```

- Execute command on the container

  `kubectl exec "$POD_NAME" -- env`

- View the source code of the app is in the `server.js` file

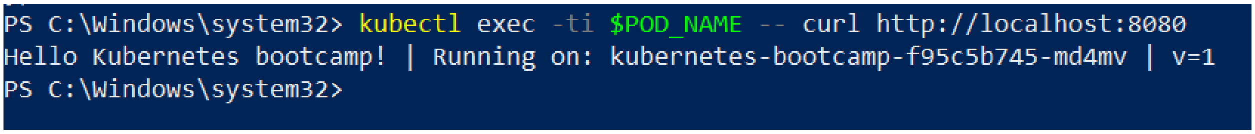
  `kubectl exec -ti $POD_NAME -- cat server.js`



```
PS C:\Windows\system32> kubectl exec -ti $POD_NAME -- cat server.js
var http = require('http');
var requests=0;
var podname= process.env.HOSTNAME;
var startTime;
var host;
var handleRequest = function(request, response) {
  response.setHeader('Content-Type', 'text/plain');
  response.writeHead(200);
  response.write("Hello Kubernetes bootcamp! | Running on: ");
  response.write(host);
  response.end(" | v=1\n");
  console.log("Running On:" ,host, "| Total Requests:", ++requests,"| App Uptime:", (new Date() - startTime)/1000 , "seconds", "| Log T
}
var www = http.createServer(handleRequest);
www.listen(8080,function () {
    startTime = new Date();;
    host = process.env.HOSTNAME;
    console.log ("Kubernetes Bootcamp App Started At:",startTime, "| Running On: " ,host, "\n" );
});
PS C:\Windows\system32> _
```

- View the application status

  `kubectl exec -ti $POD_NAME -- curl http://localhost:8080`



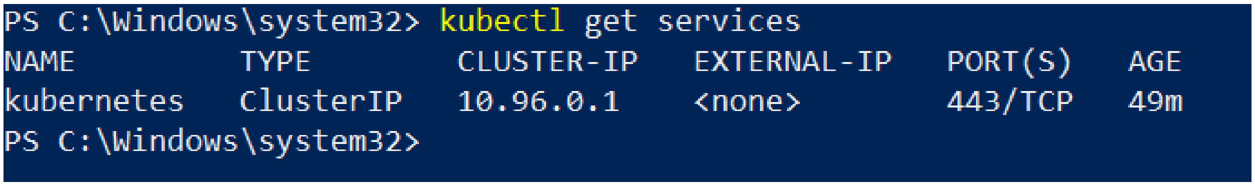
```
PS C:\Windows\system32> kubectl exec -ti $POD_NAME -- curl http://localhost:8080
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-f95c5b745-md4mv | v=1
PS C:\Windows\system32>
```

(take a screenshot)

2.3. Expose Your Appicaton

- List the current services from our cluster:

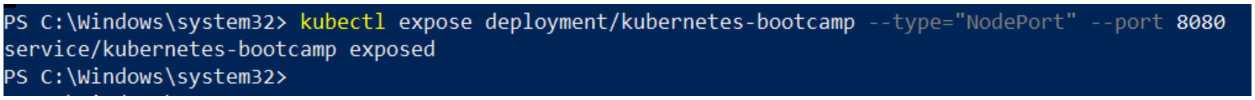
  `kubectl get services`



```
PS C:\Windows\system32> kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   49m
PS C:\Windows\system32>
```

- Create a new service and expose it to external traffic

```
kubectl expose deployment/kubernetes-bootcamp --type="NodePort" --port 8080
```

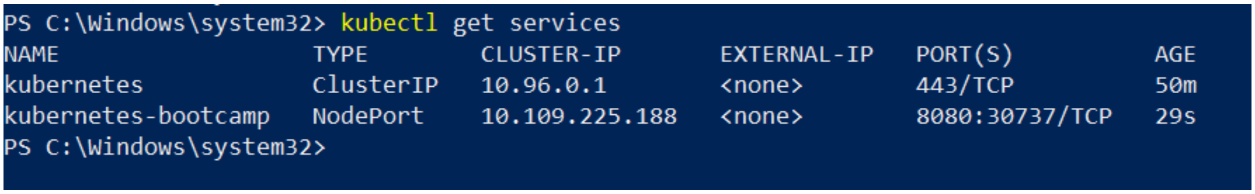


```
PS C:\Windows\system32> kubectl expose deployment/kubernetes-bootcamp --type="NodePort" --port 8080
service/kubernetes-bootcamp exposed
PS C:\Windows\system32>
```

Note: see types of services in Kubernetes

- List the current services from our cluster again:

```
kubectl get services
```

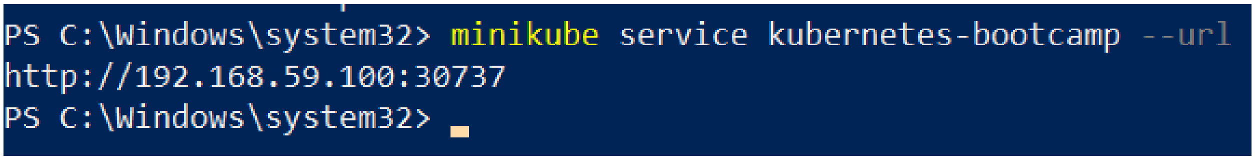


```
PS C:\Windows\system32> kubectl get services
NAME                  TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes            ClusterIP   10.96.0.1        <none>        443/TCP          50m
kubernetes-bootcamp   NodePort    10.109.225.188   <none>        8080:30737/TCP   29s
PS C:\Windows\system32>
```

(take a screenshot)

- To see the service URL

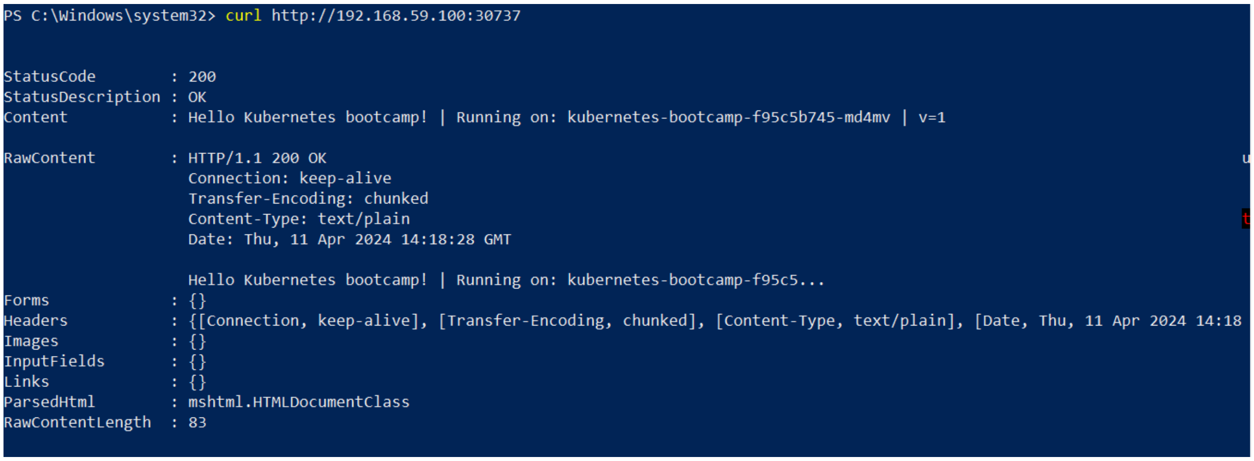
```
minikube service kubernetes-bootcamp –url
```



```
PS C:\Windows\system32> minikube service kubernetes-bootcamp --url
http://192.168.59.100:30737
PS C:\Windows\system32>
```

- Access your service in the terminal (or a web browser)

```
curl <Service URL>
```



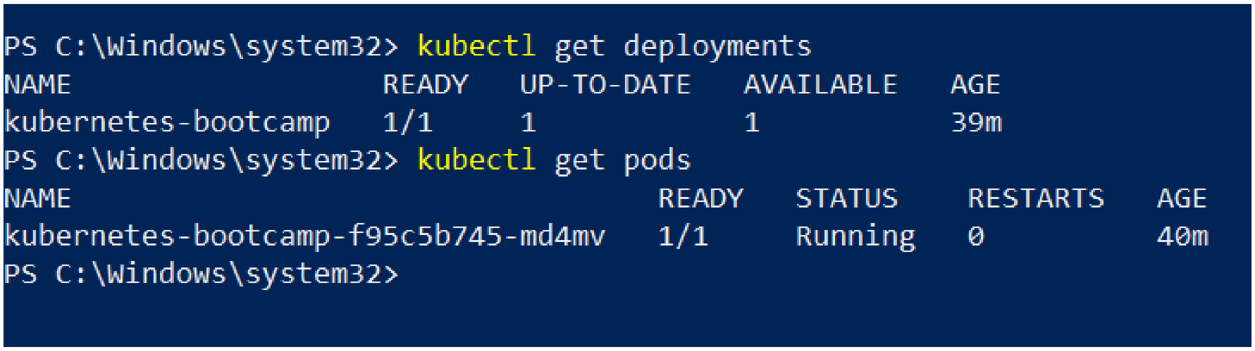
```
PS C:\Windows\system32> curl http://192.168.59.100:30737


StatusCode        : 200
StatusDescription : OK
Content           : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-f95c5b745-md4mv | v=1

RawContent        : HTTP/1.1 200 OK                                                                              u
                    Connection: keep-alive
                    Transfer-Encoding: chunked
                    Content-Type: text/plain                                                                     t
                    Date: Thu, 11 Apr 2024 14:18:28 GMT

                    Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-f95c5...
Forms             : {}
Headers           : {[Connection, keep-alive], [Transfer-Encoding, chunked], [Content-Type, text/plain], [Date, Thu, 11 Apr 2024 14:18
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 83
```

(take a screenshot)

2.4. Scale your application

- To list your deployments

```
kubectl get deployments
kubectl get pods
```

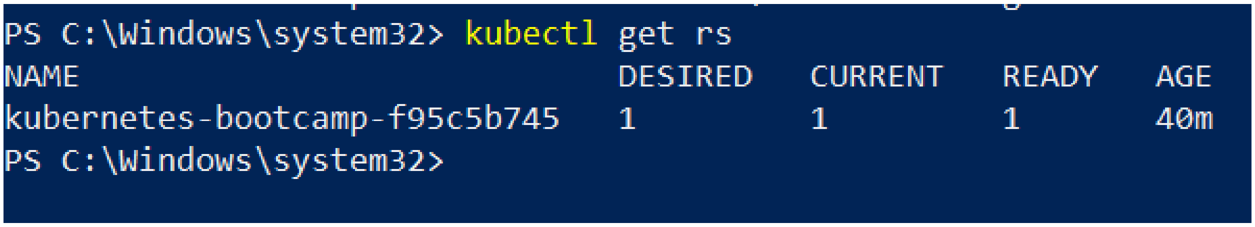


```
PS C:\Windows\system32> kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   1/1     1            1           39m
PS C:\Windows\system32> kubectl get pods
NAME                                   READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-f95c5b745-md4mv    1/1     Running   0          40m
PS C:\Windows\system32>
```

(take a screenshot)

- To see the ReplicaSet created by the Deployment

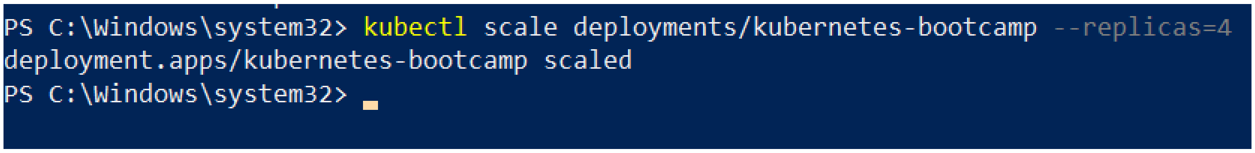
```
kubectl get rs
```



```
PS C:\Windows\system32> kubectl get rs
NAME                            DESIRED   CURRENT   READY   AGE
kubernetes-bootcamp-f95c5b745   1         1         1       40m
PS C:\Windows\system32>
```

- Scale the Deployment to 4 replicas

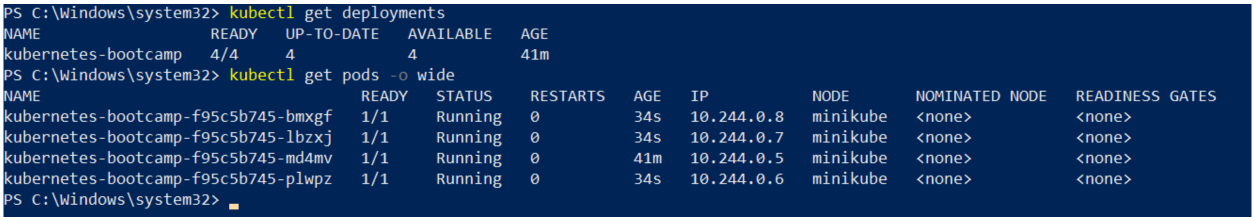
```
kubectl scale deployments/kubernetes-bootcamp --replicas=4
```



```
PS C:\Windows\system32> kubectl scale deployments/kubernetes-bootcamp --replicas=4
deployment.apps/kubernetes-bootcamp scaled
PS C:\Windows\system32>
```

- View your Deployments once again

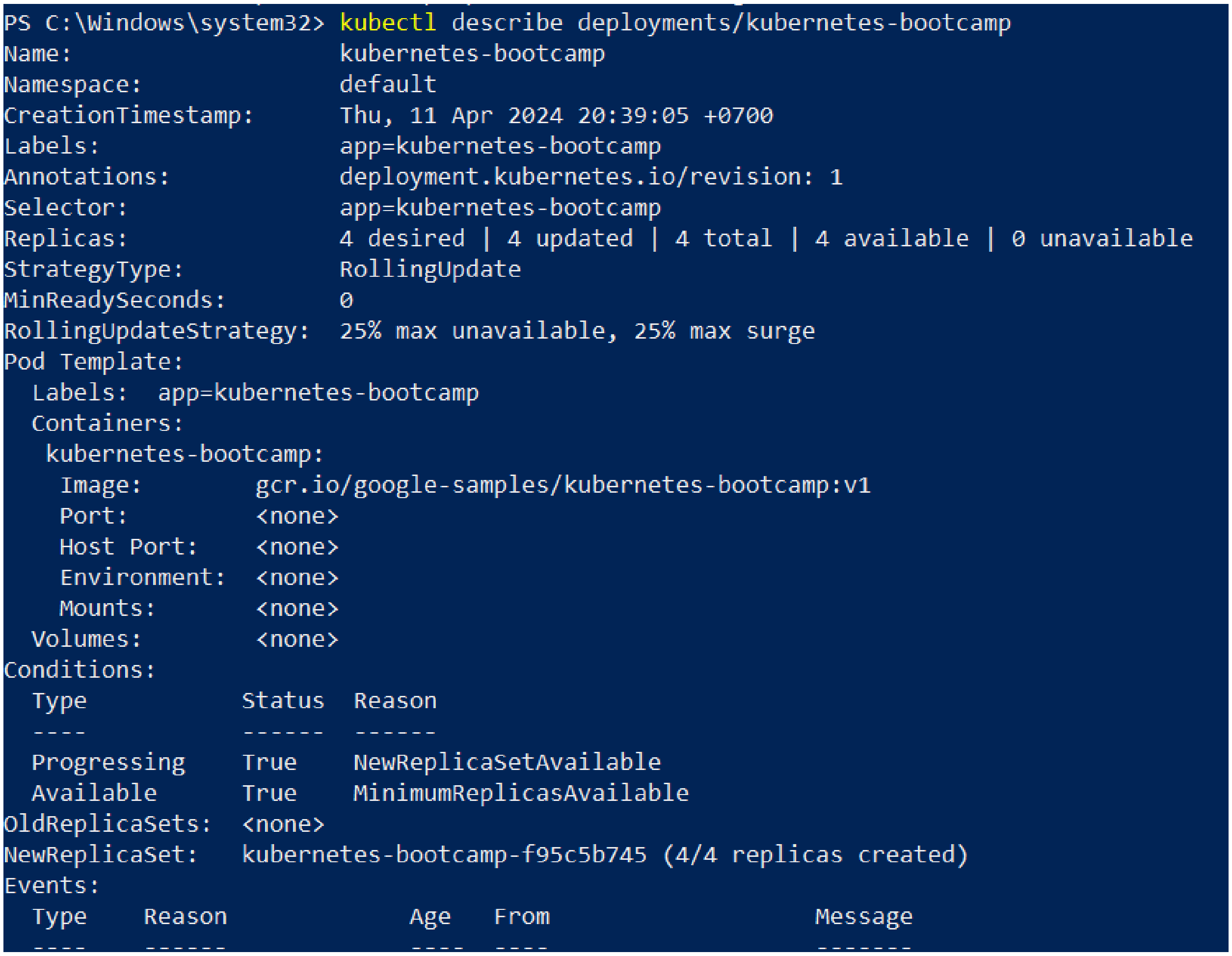
```
kubectl get deployments
kubectl get pods -o wide
```



```
PS C:\Windows\system32> kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   4/4     4            4           41m
PS C:\Windows\system32> kubectl get pods -o wide
NAME                                  READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
kubernetes-bootcamp-f95c5b745-bmxgf   1/1     Running   0          34s   10.244.0.8   minikube   <none>           <none>
kubernetes-bootcamp-f95c5b745-lbzxj   1/1     Running   0          34s   10.244.0.7   minikube   <none>           <none>
kubernetes-bootcamp-f95c5b745-md4mv   1/1     Running   0          41m   10.244.0.5   minikube   <none>           <none>
kubernetes-bootcamp-f95c5b745-plwpz   1/1     Running   0          34s   10.244.0.6   minikube   <none>           <none>
PS C:\Windows\system32>
```

(take a screenshot)

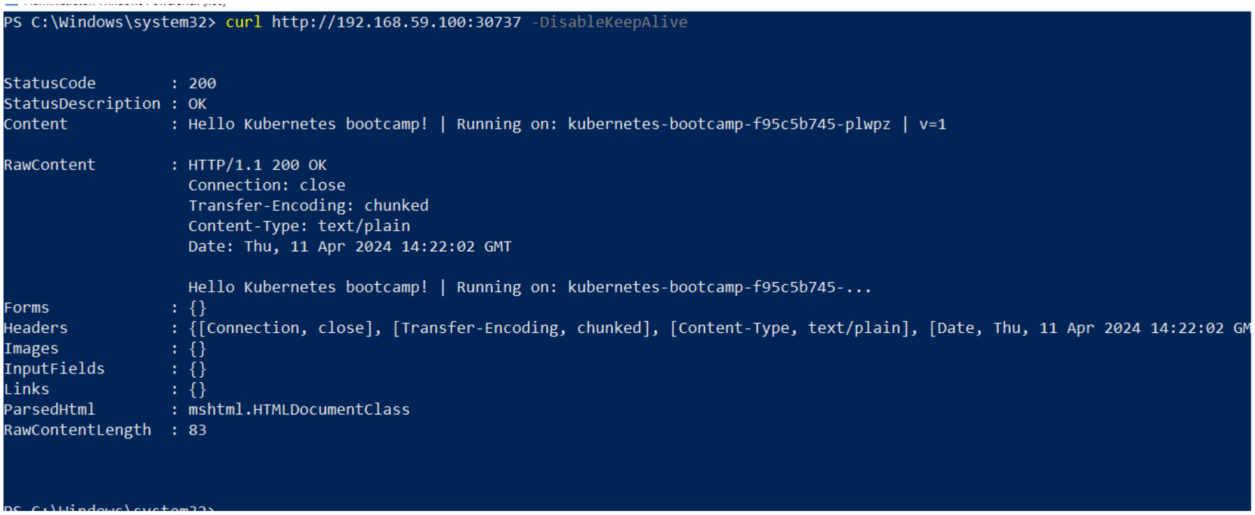
```
kubectl describe deployments/kubernetes-bootcamp
```



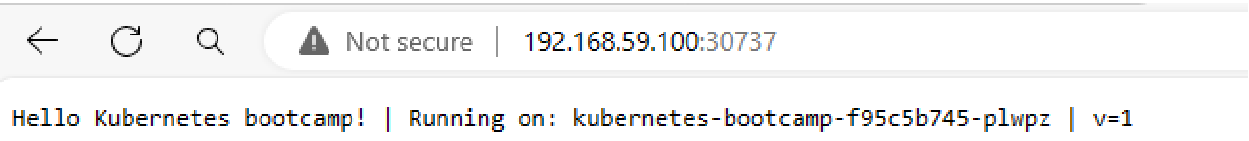
```
PS C:\Windows\system32> kubectl describe deployments/kubernetes-bootcamp
Name:                   kubernetes-bootcamp
Namespace:              default
CreationTimestamp:      Thu, 11 Apr 2024 20:39:05 +0700
Labels:                 app=kubernetes-bootcamp
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=kubernetes-bootcamp
Replicas:               4 desired | 4 updated | 4 total | 4 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=kubernetes-bootcamp
  Containers:
   kubernetes-bootcamp:
    Image:        gcr.io/google-samples/kubernetes-bootcamp:v1
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Progressing    True    NewReplicaSetAvailable
  Available      True    MinimumReplicasAvailable
OldReplicaSets:  <none>
NewReplicaSet:   kubernetes-bootcamp-f95c5b745 (4/4 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
```

(take a screenshot)

- Access your service in the terminal (or a web browser) multiple times. We hit a different Pod with every request. This demonstrates that the load-balancing is working.
  `curl <Service URL> -DisableKeepAlive`



```
PS C:\Windows\system32> curl http://192.168.59.100:30737 -DisableKeepAlive


StatusCode        : 200
StatusDescription : OK
Content           : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-f95c5b745-plwpz | v=1

RawContent        : HTTP/1.1 200 OK
                    Connection: close
                    Transfer-Encoding: chunked
                    Content-Type: text/plain
                    Date: Thu, 11 Apr 2024 14:22:02 GMT

                    Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-f95c5b745-...
Forms             : {}
Headers           : {[Connection, close], [Transfer-Encoding, chunked], [Content-Type, text/plain], [Date, Thu, 11 Apr 2024 14:22:02 GM
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 83



PS C:\Windows\system32>
```

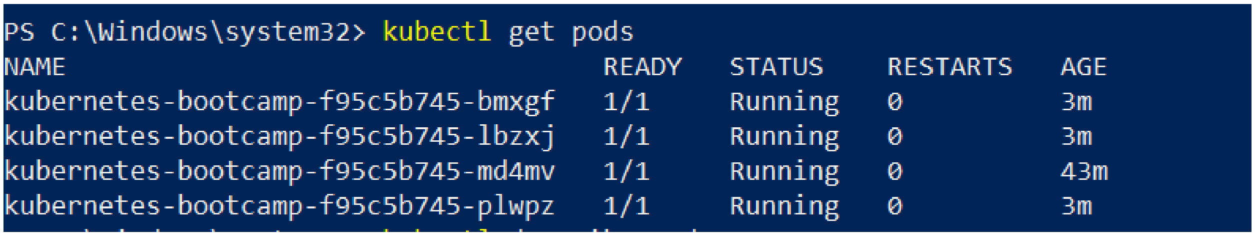


```
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-f95c5b745-plwpz | v=1
```

2.5. Update your application

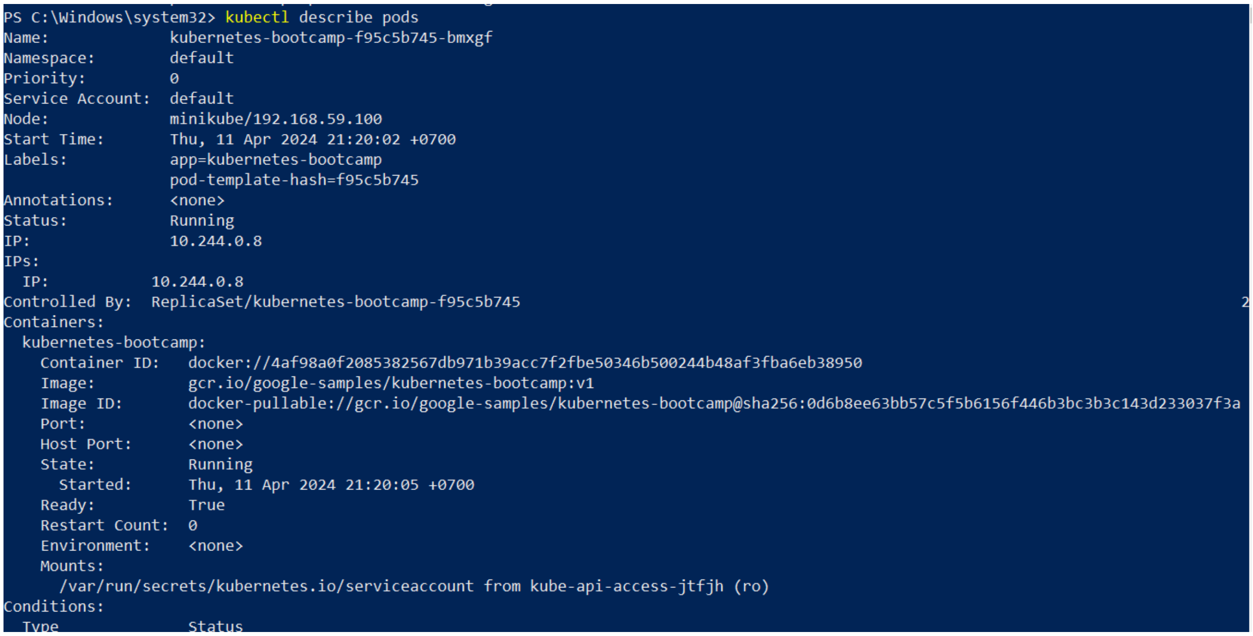
- To view the current image version of the app

  `kubectl get pods`



```
PS C:\Windows\system32> kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-f95c5b745-bmxgf   1/1     Running   0          3m
kubernetes-bootcamp-f95c5b745-lbzxj   1/1     Running   0          3m
kubernetes-bootcamp-f95c5b745-md4mv   1/1     Running   0          43m
kubernetes-bootcamp-f95c5b745-plwpz   1/1     Running   0          3m
```

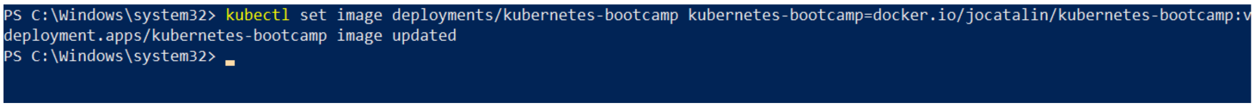
  `kubectl describe pods`



```
PS C:\Windows\system32> kubectl describe pods
Name:             kubernetes-bootcamp-f95c5b745-bmxgf
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.59.100
Start Time:       Thu, 11 Apr 2024 21:20:02 +0700
Labels:           app=kubernetes-bootcamp
                  pod-template-hash=f95c5b745
Annotations:      <none>
Status:           Running
IP:               10.244.0.8
IPs:
  IP:           10.244.0.8
Controlled By:  ReplicaSet/kubernetes-bootcamp-f95c5b745
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://4af98a0f2085382567db971b39acc7f2fbe50346b500244b48af3fba6eb38950
    Image:          gcr.io/google-samples/kubernetes-bootcamp:v1
    Image ID:       docker-pullable://gcr.io/google-samples/kubernetes-bootcamp@sha256:0d6b8ee63bb57c5f5b6156f446b3bc3b3c143d233037f3a
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 11 Apr 2024 21:20:05 +0700
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-jtfjh (ro)
Conditions:
  Type              Status
```

- Update the image of the application to version 2

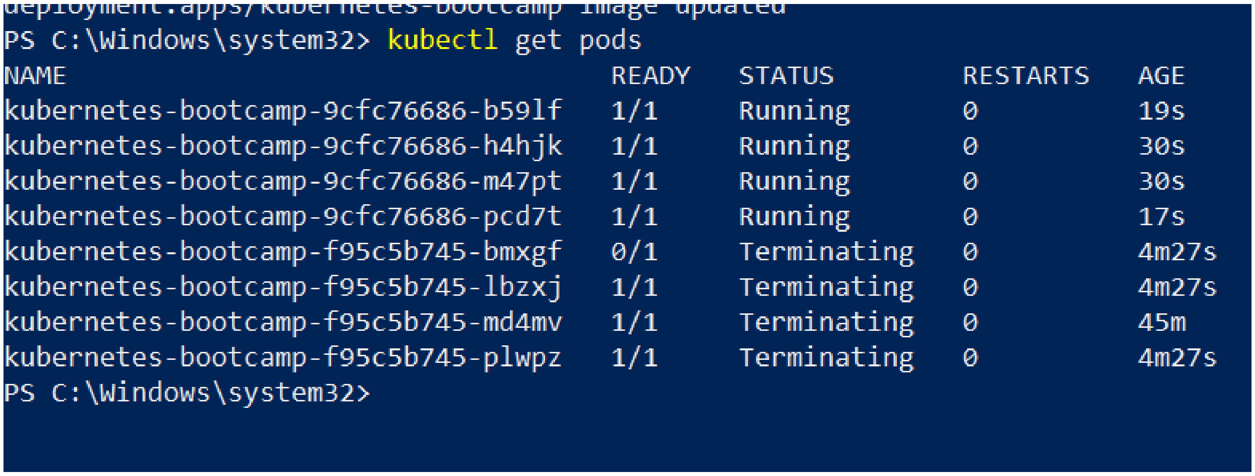
  `kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=docker.io/jocatalin/kubernetes-bootcamp:v2`



```
PS C:\Windows\system32> kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=docker.io/jocatalin/kubernetes-bootcamp:v
deployment.apps/kubernetes-bootcamp image updated
PS C:\Windows\system32> _
```

- Check the status of the new Pods, and view the old one terminating

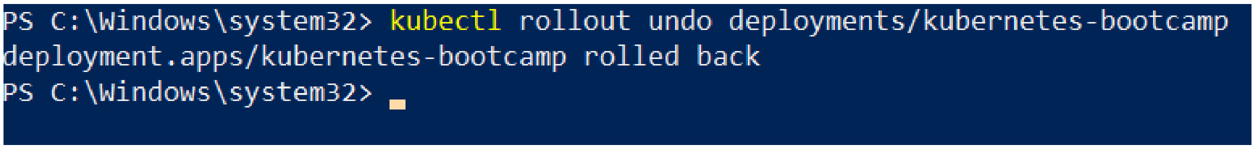
  `kubectl get pods`



```
deployment.apps/kubernetes-bootcamp image updated
PS C:\Windows\system32> kubectl get pods
NAME                                  READY   STATUS        RESTARTS   AGE
kubernetes-bootcamp-9cfc76686-b59lf   1/1     Running       0          19s
kubernetes-bootcamp-9cfc76686-h4hjk   1/1     Running       0          30s
kubernetes-bootcamp-9cfc76686-m47pt   1/1     Running       0          30s
kubernetes-bootcamp-9cfc76686-pcd7t   1/1     Running       0          17s
kubernetes-bootcamp-f95c5b745-bmxgf   0/1     Terminating   0          4m27s
kubernetes-bootcamp-f95c5b745-lbzxj   1/1     Terminating   0          4m27s
kubernetes-bootcamp-f95c5b745-md4mv   1/1     Terminating   0          45m
kubernetes-bootcamp-f95c5b745-plwpz   1/1     Terminating   0          4m27s
PS C:\Windows\system32>
```

- Roll back the deployment to your last working version

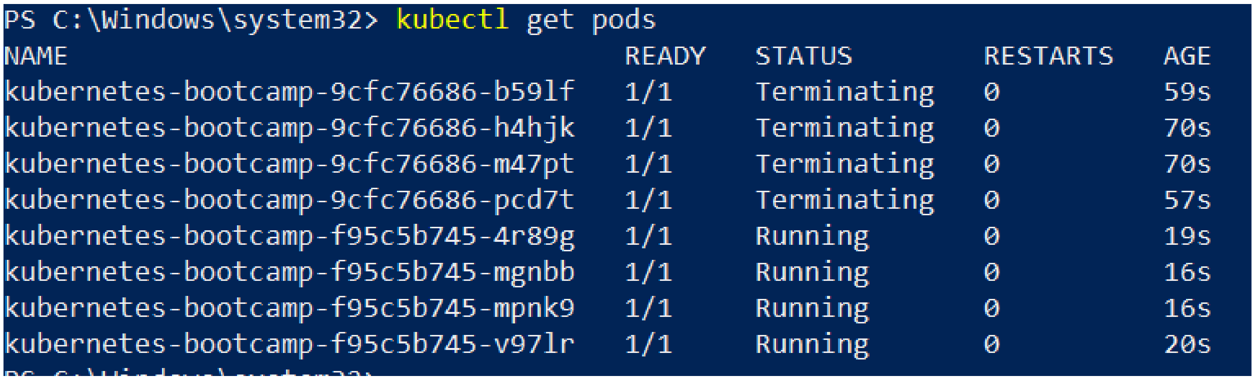
  `kubectl rollout undo deployments/kubernetes-bootcamp`



```
PS C:\Windows\system32> kubectl rollout undo deployments/kubernetes-bootcamp
deployment.apps/kubernetes-bootcamp rolled back
PS C:\Windows\system32>
```

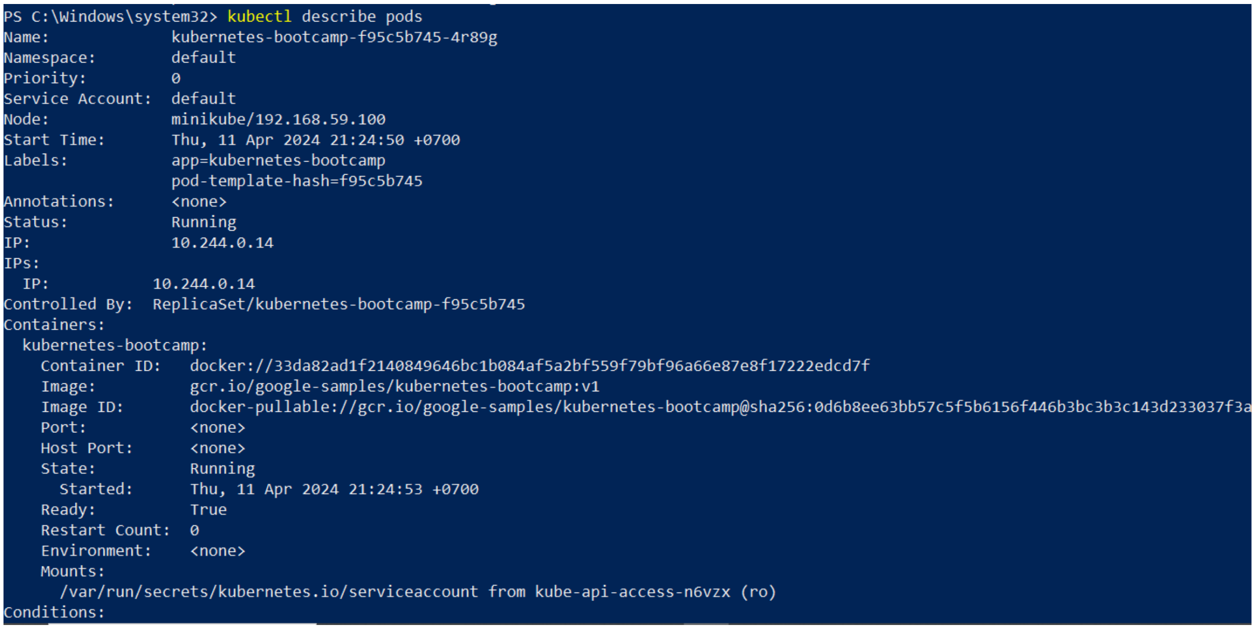
- To view the current image version of the app

  `kubectl get pods`



```
PS C:\Windows\system32> kubectl get pods
NAME                                  READY   STATUS        RESTARTS   AGE
kubernetes-bootcamp-9cfc76686-b59lf   1/1     Terminating   0          59s
kubernetes-bootcamp-9cfc76686-h4hjk   1/1     Terminating   0          70s
kubernetes-bootcamp-9cfc76686-m47pt   1/1     Terminating   0          70s
kubernetes-bootcamp-9cfc76686-pcd7t   1/1     Terminating   0          57s
kubernetes-bootcamp-f95c5b745-4r89g   1/1     Running       0          19s
kubernetes-bootcamp-f95c5b745-mgnbb   1/1     Running       0          16s
kubernetes-bootcamp-f95c5b745-mpnk9   1/1     Running       0          16s
kubernetes-bootcamp-f95c5b745-v97lr   1/1     Running       0          20s
PS C:\Windows\system32>
```

  `kubectl describe pods`



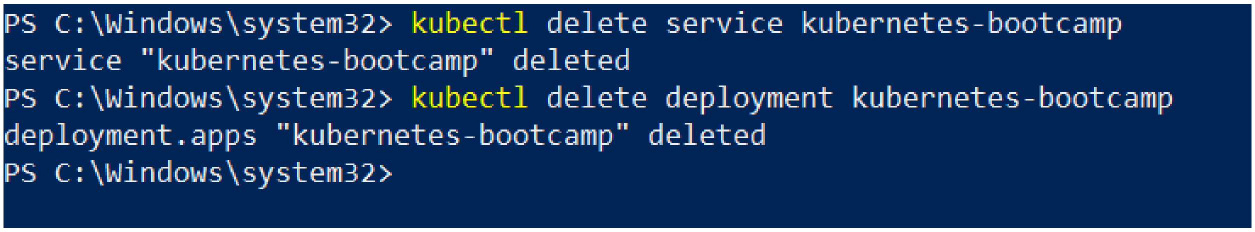
```
PS C:\Windows\system32> kubectl describe pods
Name:             kubernetes-bootcamp-f95c5b745-4r89g
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.59.100
Start Time:       Thu, 11 Apr 2024 21:24:50 +0700
Labels:           app=kubernetes-bootcamp
                  pod-template-hash=f95c5b745
Annotations:      <none>
Status:           Running
IP:               10.244.0.14
IPs:
  IP:           10.244.0.14
Controlled By:  ReplicaSet/kubernetes-bootcamp-f95c5b745
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://33da82ad1f2140849646bc1b084af5a2bf559f79bf96a66e87e8f17222edcd7f
    Image:          gcr.io/google-samples/kubernetes-bootcamp:v1
    Image ID:       docker-pullable://gcr.io/google-samples/kubernetes-bootcamp@sha256:0d6b8ee63bb57c5f5b6156f446b3bc3b3c143d233037f3a
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 11 Apr 2024 21:24:53 +0700
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-n6vzx (ro)
Conditions:
```

(take a screenshot)

2.6. Cleaning up

- We can clean up the resources we created in the cluster:

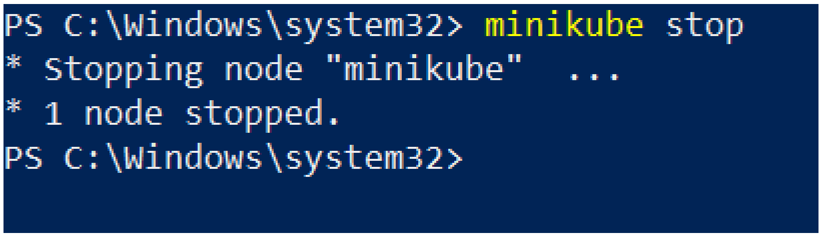
```
kubectl delete service kubernetes-bootcamp
kubectl delete deployment kubernetes-bootcamp
```



```
PS C:\Windows\system32> kubectl delete service kubernetes-bootcamp
service "kubernetes-bootcamp" deleted
PS C:\Windows\system32> kubectl delete deployment kubernetes-bootcamp
deployment.apps "kubernetes-bootcamp" deleted
PS C:\Windows\system32>
```

- Stop the Minikube cluster

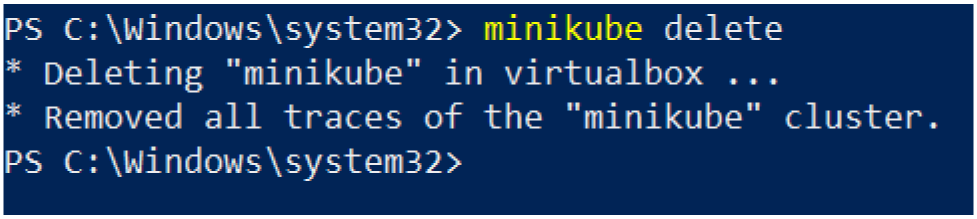
```
minikube stop
```



```
PS C:\Windows\system32> minikube stop
* Stopping node "minikube"  ...
* 1 node stopped.
PS C:\Windows\system32>
```

- Delete the Minikube VM (optional)

```
minikube delete
```

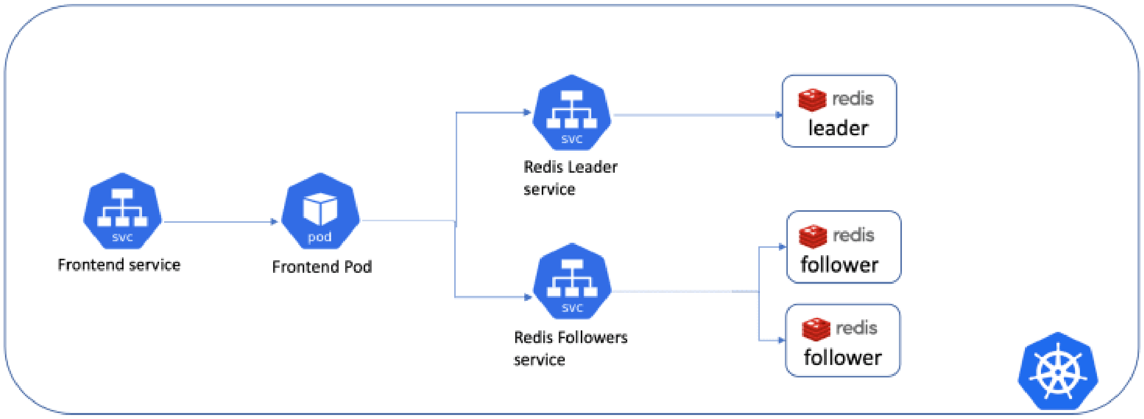


```
PS C:\Windows\system32> minikube delete
* Deleting "minikube" in virtualbox ...
* Removed all traces of the "minikube" cluster.
PS C:\Windows\system32>
```

3. Deploying PHP Guestbook application with Redis

This tutorial shows you how to build and deploy a simple (not production ready), multi-tier web application using Kubernetes and Docker. This example consists of the following components:

+ A single-instance Redis to store guestbook entries
+ Multiple web frontend instances



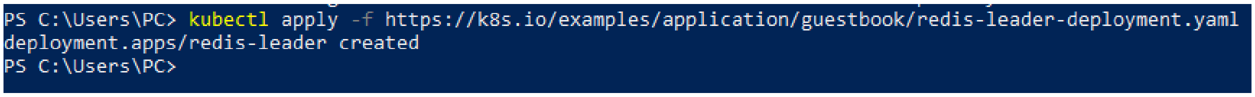
3.1. Start up the Redis Database

The guestbook application uses Redis to store its data.

a. Creating the Redis Deployment

- Apply the Redis Deployment from the redis-leader-deployment.yaml file:

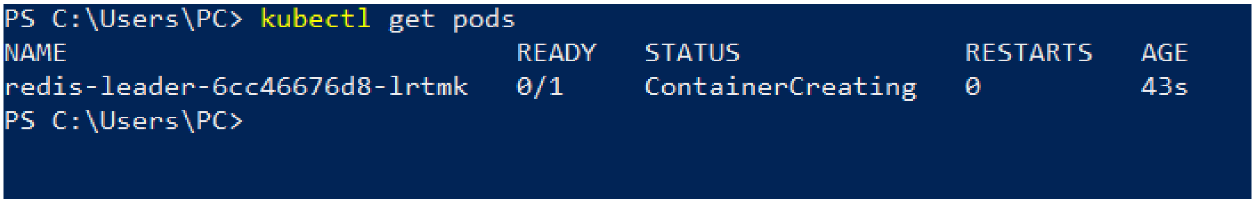
```
\kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-deployment.yaml
```



```
PS C:\Users\PC> kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-deployment.yaml
deployment.apps/redis-leader created
PS C:\Users\PC>
```

- Query the list of Pods to verify that the Redis Pod is running:

```
kubectl get pods
```

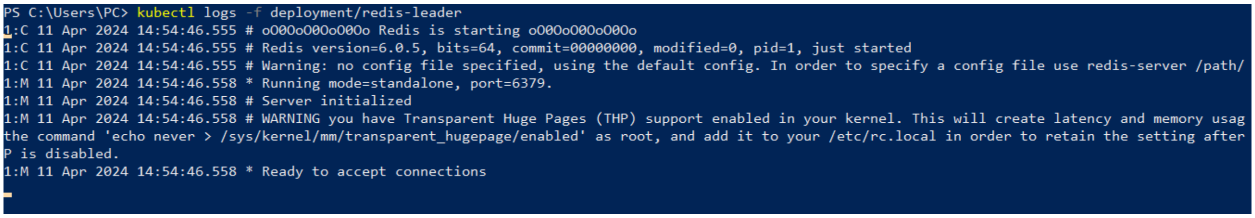


```
PS C:\Users\PC> kubectl get pods
NAME                           READY   STATUS              RESTARTS   AGE
redis-leader-6cc46676d8-lrtmk  0/1     ContainerCreating   0          43s
PS C:\Users\PC>
```

(take a screenshot)

- Run the following command to view the logs from the Redis leader Pod:

```
kubectl logs -f deployment/redis-leader
```

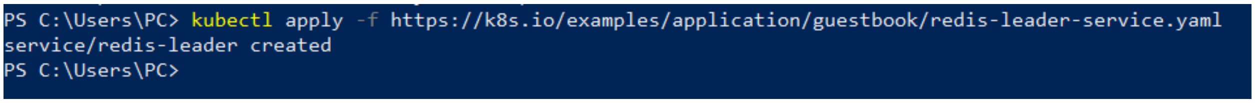


```
PS C:\Users\PC> kubectl logs -f deployment/redis-leader
1:C 11 Apr 2024 14:54:46.555 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
1:C 11 Apr 2024 14:54:46.555 # Redis version=6.0.5, bits=64, commit=00000000, modified=0, pid=1, just started
1:C 11 Apr 2024 14:54:46.555 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/
1:M 11 Apr 2024 14:54:46.558 * Running mode=standalone, port=6379.
1:M 11 Apr 2024 14:54:46.558 # Server initialized
1:M 11 Apr 2024 14:54:46.558 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usag
the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after
P is disabled.
1:M 11 Apr 2024 14:54:46.558 * Ready to accept connections
```

b. Creating the Redis leader Service

- Apply the Redis Service from the following redis-leader-service.yaml file

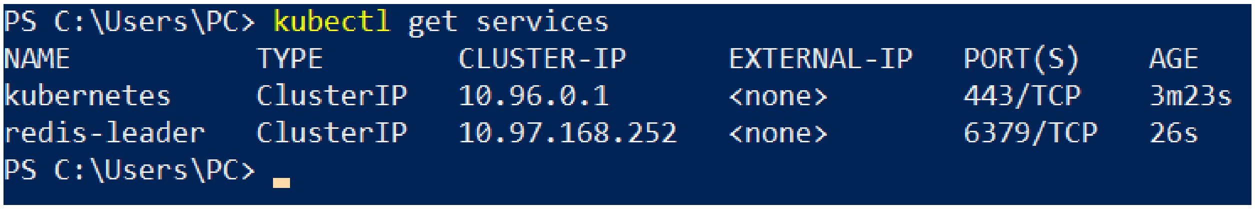
  `kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-service.yaml`



```
PS C:\Users\PC> kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-service.yaml
service/redis-leader created
PS C:\Users\PC>
```

- Query the list of Services to verify that the Redis Service is running:

  `kubectl get services`



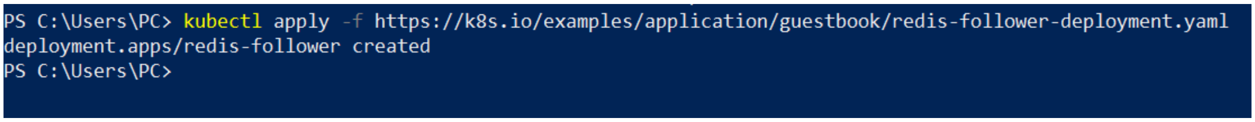
```
PS C:\Users\PC> kubectl get services
NAME           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
kubernetes     ClusterIP   10.96.0.1       <none>        443/TCP    3m23s
redis-leader   ClusterIP   10.97.168.252   <none>        6379/TCP   26s
PS C:\Users\PC>
```

(take a screenshot)

c. Set up Redis followers

- Apply the Redis Deployment from the following redis-follower-deployment.yaml file:

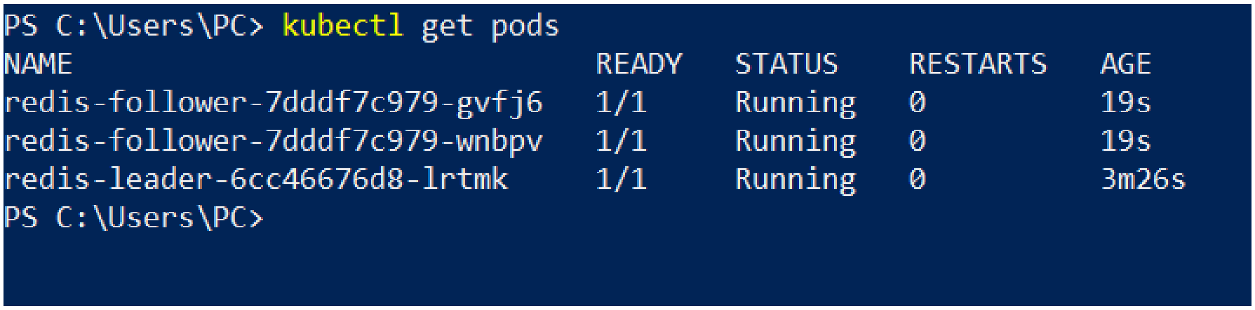
  `kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-deployment.yaml`



```
PS C:\Users\PC> kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-deployment.yaml
deployment.apps/redis-follower created
PS C:\Users\PC>
```

- Verify that the two Redis follower replicas are running by querying the list of Pods:

  `kubectl get pods`

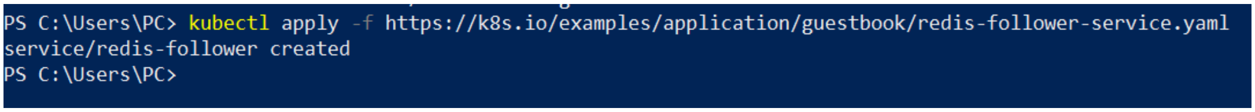


```
PS C:\Users\PC> kubectl get pods
NAME                             READY   STATUS    RESTARTS   AGE
redis-follower-7dddf7c979-gvfj6  1/1     Running   0          19s
redis-follower-7dddf7c979-wnbpv  1/1     Running   0          19s
redis-leader-6cc46676d8-lrtmk    1/1     Running   0          3m26s
PS C:\Users\PC>
```

(take a screenshot)

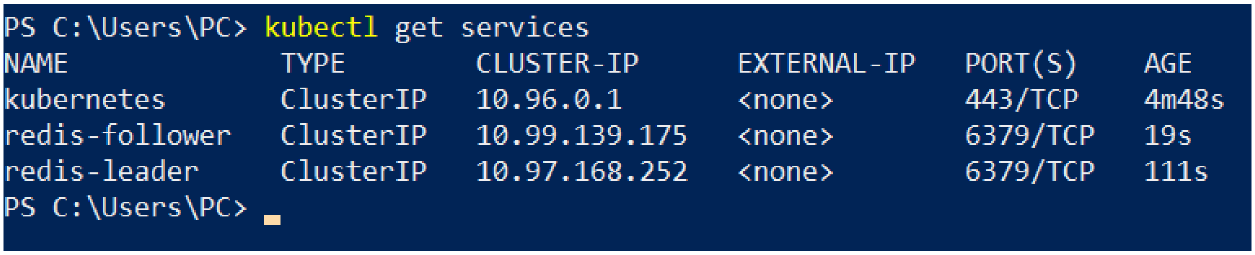
d. Creating the Redis follower service

- Apply the Redis Service from the following redis-follower-service.yaml file:
  `kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-service.yaml`



```
PS C:\Users\PC> kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-service.yaml
service/redis-follower created
PS C:\Users\PC>
```

- Query the list of Services to verify that the Redis Service is running:
  `kubectl get services`



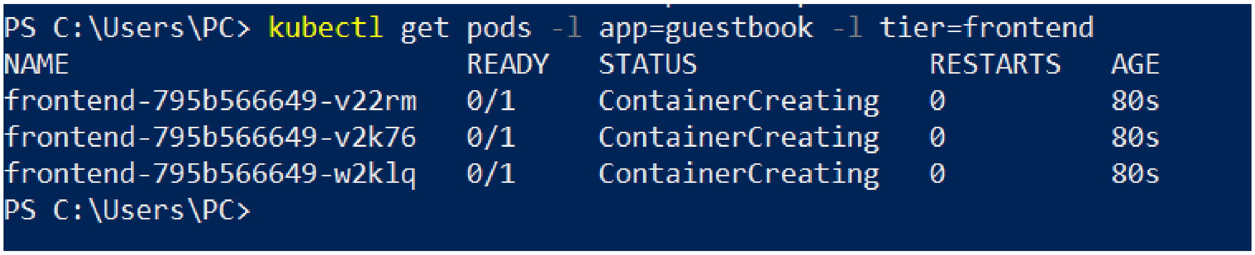
```
PS C:\Users\PC> kubectl get services
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
kubernetes       ClusterIP   10.96.0.1       <none>        443/TCP    4m48s
redis-follower   ClusterIP   10.99.139.175   <none>        6379/TCP   19s
redis-leader     ClusterIP   10.97.168.252   <none>        6379/TCP   111s
PS C:\Users\PC>
```

(take a screenshot)

3.2. Set up and Expose the Guestbook Frontend

a. Creating the Guestbook Frontend Deployment

- Apply the frontend Deployment from the frontend-deployment.yaml file:
  `kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-deployment.yaml`
- Query the list of Pods to verify that the three frontend replicas are running:
  `kubectl get pods -l app=guestbook -l tier=frontend`

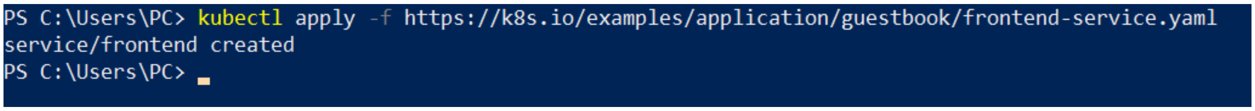


```
PS C:\Users\PC> kubectl get pods -l app=guestbook -l tier=frontend
NAME                        READY   STATUS              RESTARTS   AGE
frontend-795b566649-v22rm   0/1     ContainerCreating   0          80s
frontend-795b566649-v2k76   0/1     ContainerCreating   0          80s
frontend-795b566649-w2klq   0/1     ContainerCreating   0          80s
PS C:\Users\PC>
```

(take a screenshot)

b. Creating the Frontend Service

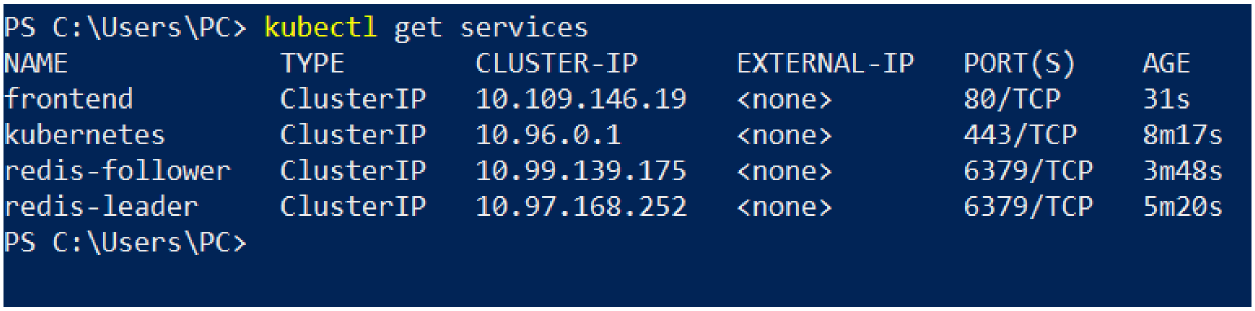
- Apply the frontend Service from the frontend-service.yaml file:
  `kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-service.yaml`



```
PS C:\Users\PC> kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-service.yaml
service/frontend created
PS C:\Users\PC>
```

- Query the list of Services to verify that the frontend Service is running:

```
kubectl get services
```



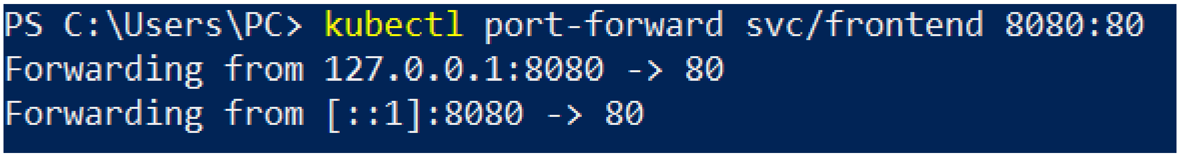
```
PS C:\Users\PC> kubectl get services
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
frontend         ClusterIP   10.109.146.19   <none>        80/TCP     31s
kubernetes       ClusterIP   10.96.0.1       <none>        443/TCP    8m17s
redis-follower   ClusterIP   10.99.139.175   <none>        6379/TCP   3m48s
redis-leader     ClusterIP   10.97.168.252   <none>        6379/TCP   5m20s
PS C:\Users\PC>
```

(take a screenshot)

c. Viewing the Frontend Service via `kubectl port-forward`

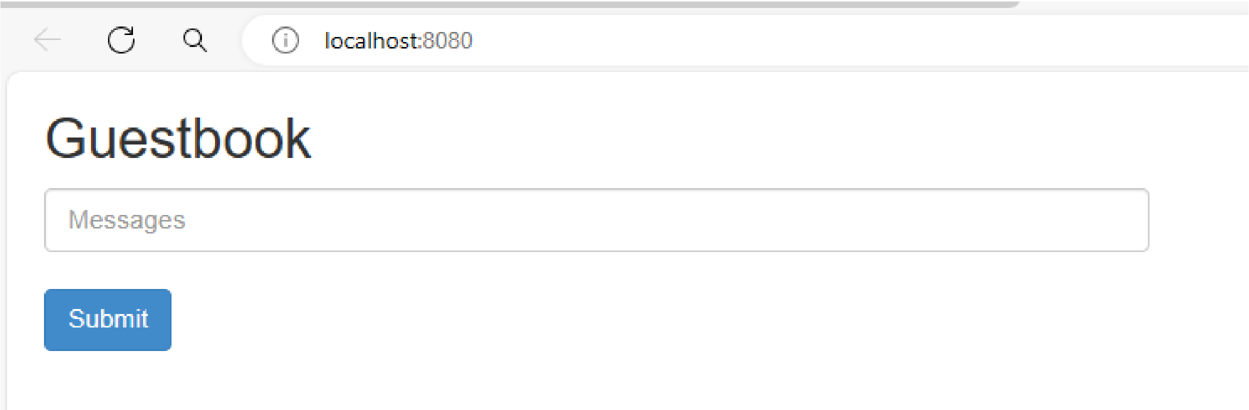
- Run the following command to forward port 8080 on your local machine to port 80 on the service.

```
kubectl port-forward svc/frontend 8080:80
```



```
PS C:\Users\PC> kubectl port-forward svc/frontend 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

- Load the page http://localhost:8080 in your browser to view your guestbook. (take a screenshot)



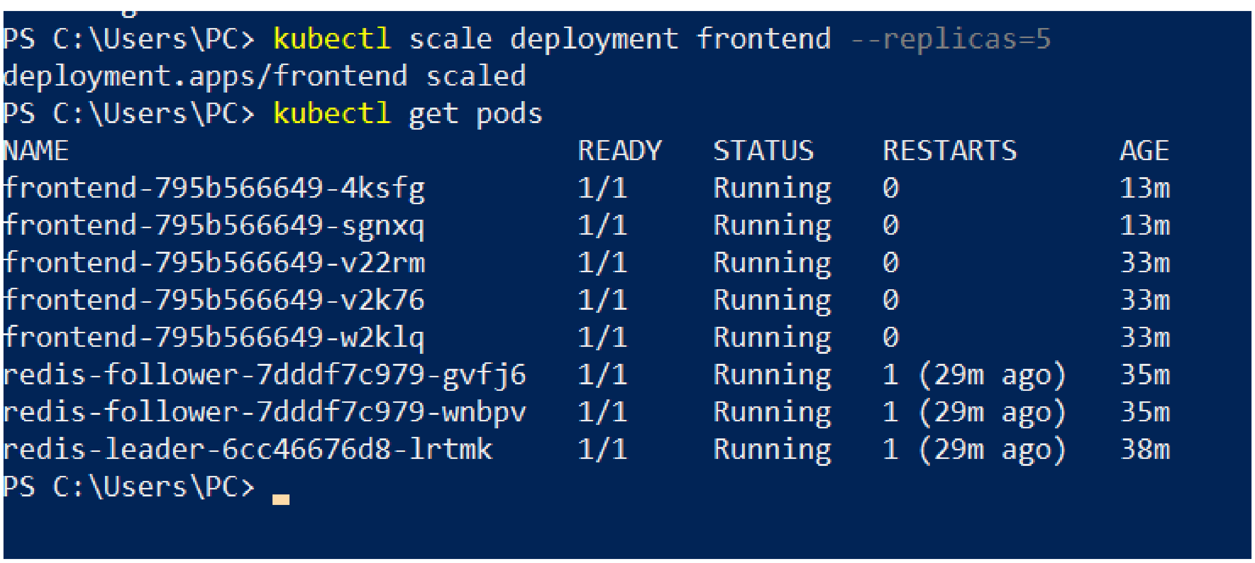
d. Viewing the Frontend Service via LoadBalancer

- If you deployed the `frontend-service.yaml` manifest with type: LoadBalancer you need to find the IP address to view your Guestbook.
- Copy the external IP address, and load the page in your browser to view your guestbook.

3.3. Scale the Web Frontend

- Run the following command to scale up the number of frontend Pods:

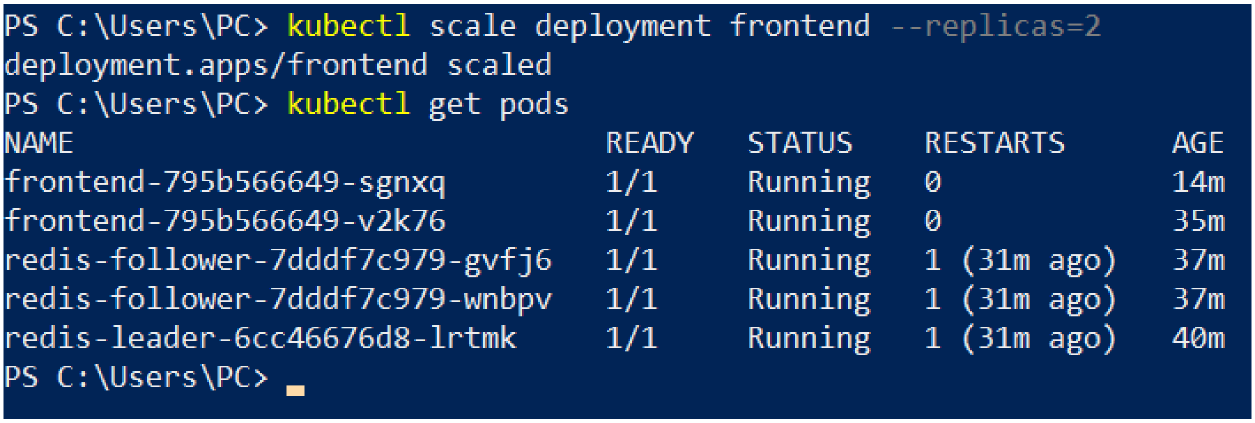
```
kubectl scale deployment frontend --replicas=5
kubectl get pods
```



```
PS C:\Users\PC> kubectl scale deployment frontend --replicas=5
deployment.apps/frontend scaled
PS C:\Users\PC> kubectl get pods
NAME                               READY   STATUS    RESTARTS      AGE
frontend-795b566649-4ksfg          1/1     Running   0             13m
frontend-795b566649-sgnxq          1/1     Running   0             13m
frontend-795b566649-v22rm          1/1     Running   0             33m
frontend-795b566649-v2k76          1/1     Running   0             33m
frontend-795b566649-w2klq          1/1     Running   0             33m
redis-follower-7dddf7c979-gvfj6    1/1     Running   1 (29m ago)   35m
redis-follower-7dddf7c979-wnbpv    1/1     Running   1 (29m ago)   35m
redis-leader-6cc46676d8-lrtmk      1/1     Running   1 (29m ago)   38m
PS C:\Users\PC>
```

- Run the following command to scale down the number of frontend Pods:

```
kubectl scale deployment frontend --replicas=2
kubectl get pods
```



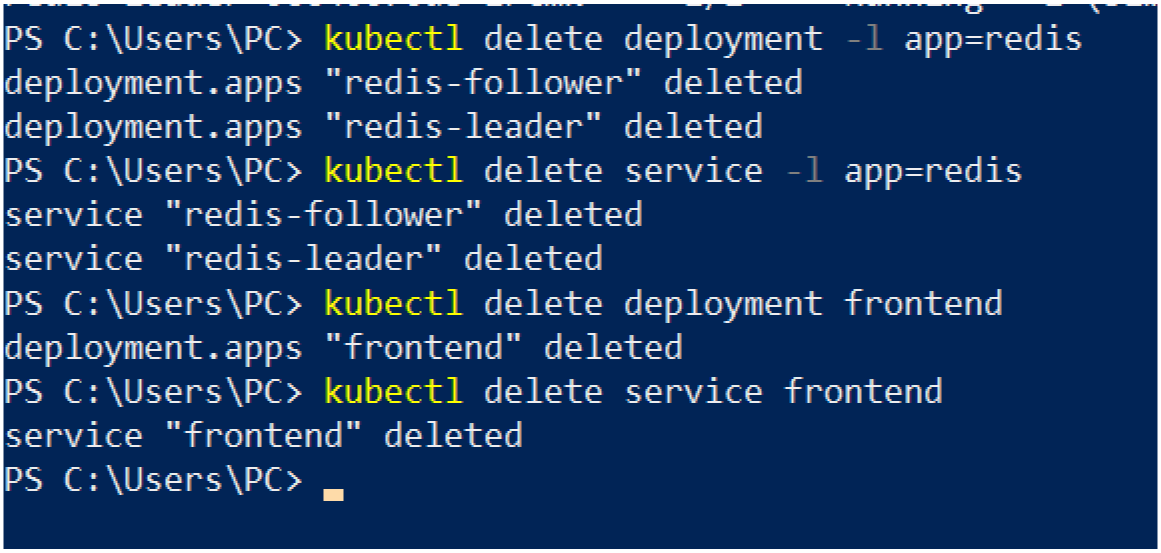
```
PS C:\Users\PC> kubectl scale deployment frontend --replicas=2
deployment.apps/frontend scaled
PS C:\Users\PC> kubectl get pods
NAME                               READY   STATUS    RESTARTS      AGE
frontend-795b566649-sgnxq          1/1     Running   0             14m
frontend-795b566649-v2k76          1/1     Running   0             35m
redis-follower-7dddf7c979-gvfj6    1/1     Running   1 (31m ago)   37m
redis-follower-7dddf7c979-wnbpv    1/1     Running   1 (31m ago)   37m
redis-leader-6cc46676d8-lrtmk      1/1     Running   1 (31m ago)   40m
PS C:\Users\PC>
```

(take a screenshot)

3.4. Cleaning up

- Run the following commands to delete all Pods, Deployments, and Services.

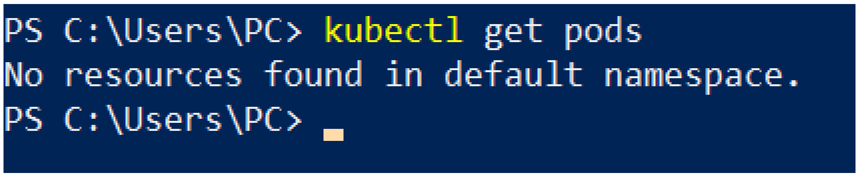
```
kubectl delete deployment -l app=redis
kubectl delete service -l app=redis
kubectl delete deployment frontend
kubectl delete service frontend
```



```
PS C:\Users\PC> kubectl delete deployment -l app=redis
deployment.apps "redis-follower" deleted
deployment.apps "redis-leader" deleted
PS C:\Users\PC> kubectl delete service -l app=redis
service "redis-follower" deleted
service "redis-leader" deleted
PS C:\Users\PC> kubectl delete deployment frontend
deployment.apps "frontend" deleted
PS C:\Users\PC> kubectl delete service frontend
service "frontend" deleted
PS C:\Users\PC> _
```

- Query the list of Pods to verify that no Pods are running

  `kubectl get pods`



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

(take a screenshot)

---END---