

## LAB 5



### Containerized applications deployment and management using Kubernetes

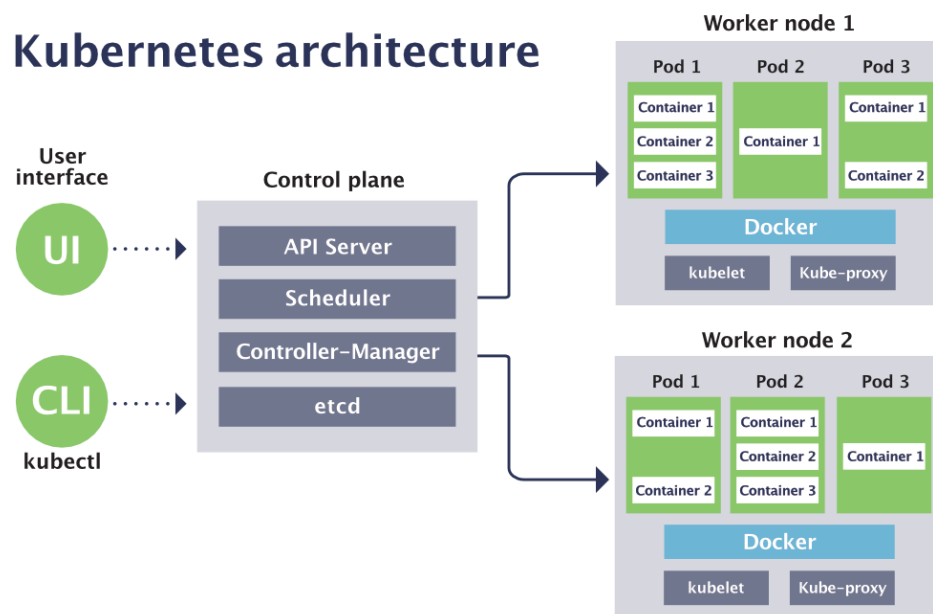
Full name: Trương Đăng Trúc Lâm

Student ID: B2111933

- 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

### Kubernetes architecture



**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

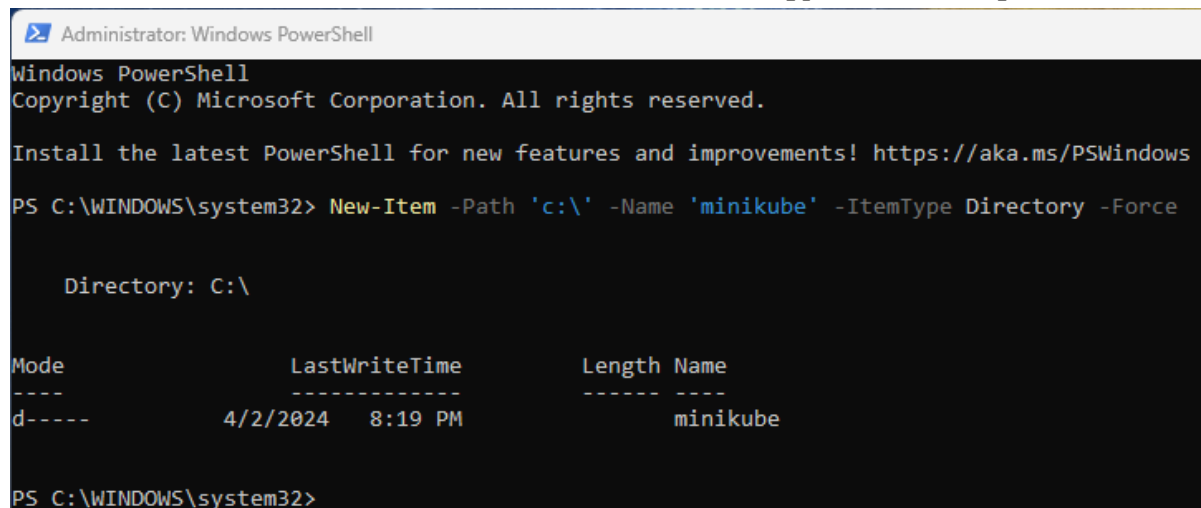
**Services:** A method for exposing an application that is running as one or more pods in cluster.

## 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
```

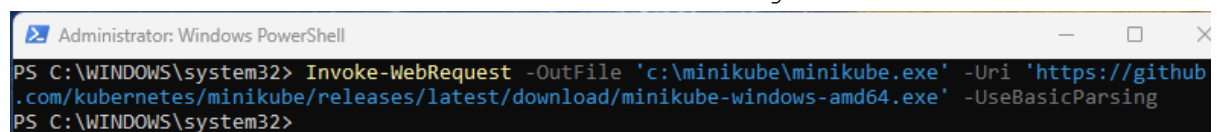


The screenshot shows a Windows PowerShell window titled 'Administrator: Windows PowerShell'. It displays the command `New-Item -Path 'c:\' -Name 'minikube' -ItemType Directory -Force` being executed. The output shows the directory 'minikube' is created under 'C:\'. A table-like output is shown below the command:

Mode	LastWriteTime	Length	Name
d-----	4/2/2024 8:19 PM		minikube

Create a directory named minikube

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



The screenshot shows a Windows PowerShell window titled 'Administrator: Windows PowerShell'. It displays the command `Invoke-WebRequest -OutFile 'c:\minikube\minikube.exe' -Uri 'https://github.com/kubernetes/minikube/releases/latest/download/minikube-windows-amd64.exe' -UseBasicParsing` being executed. The command is partially visible in the terminal.

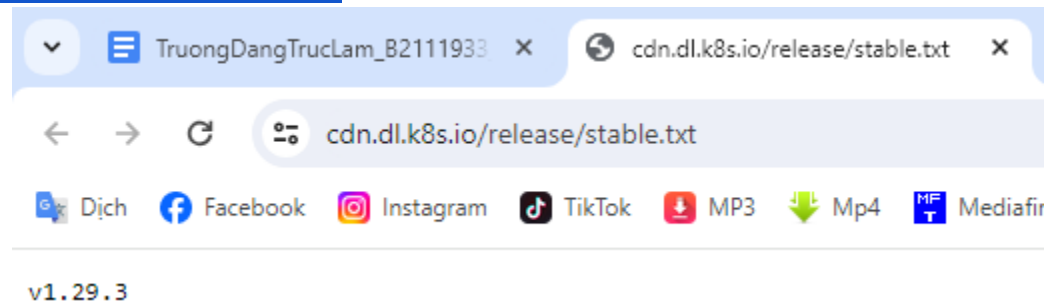
Download the latest Minikube installer for Windows (64-bit)

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

```
Administrator: Windows PowerShell
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
PS C:\WINDOWS\system32>
```

Downloads the latest kubectl executable for Windows (64-bit)

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



v1.29.3 is the latest stable version

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

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> $oldPath = [Environment]::GetEnvironmentVariable('Path', [EnvironmentVariableTarget]::Machine)
PS C:\WINDOWS\system32>
```

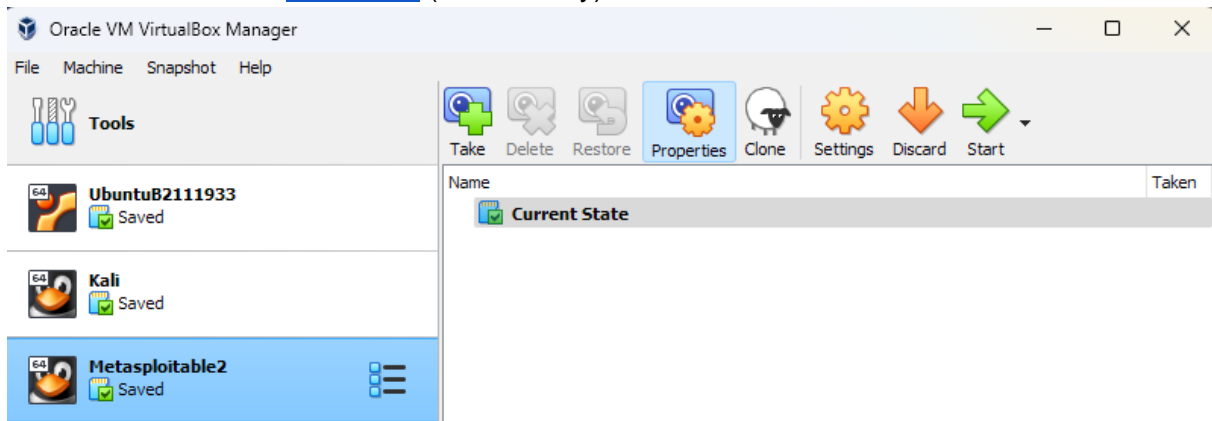
Add the minikube.exe and kubectl.exe binary to the PATH

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

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

Check if the directory "C:\minikube" is already in the system PATH and adds it if not

- Download and install [VirtualBox](#) (if necessary)



Already installed

## 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
W0402 20:43:28.284741 7764 main.go:291] Unable to resolve the current Docker CLI context "default": context "defa
: context not found: open C:\Users\lambo\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f41
3f0688f\meta.json: The system cannot find the path specified.
* minikube v1.32.0 on Microsoft Windows 11 Pro 10.0.22631.3296 Build 22631.3296
* Using the virtualbox driver based on existing profile
* Starting control plane node minikube in cluster minikube
* virtualbox "minikube" VM is missing, will recreate.
* Creating virtualbox VM (CPUs=2, Memory=4000MB, 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/networ
/proxy/
* Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
  - 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
PS C:\WINDOWS\system32>
```

Using Minikube to create a cluster

- Open the Dashboard

# Start a new terminal, and leave this running.

`minikube dashboard`

(take a screenshot)

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

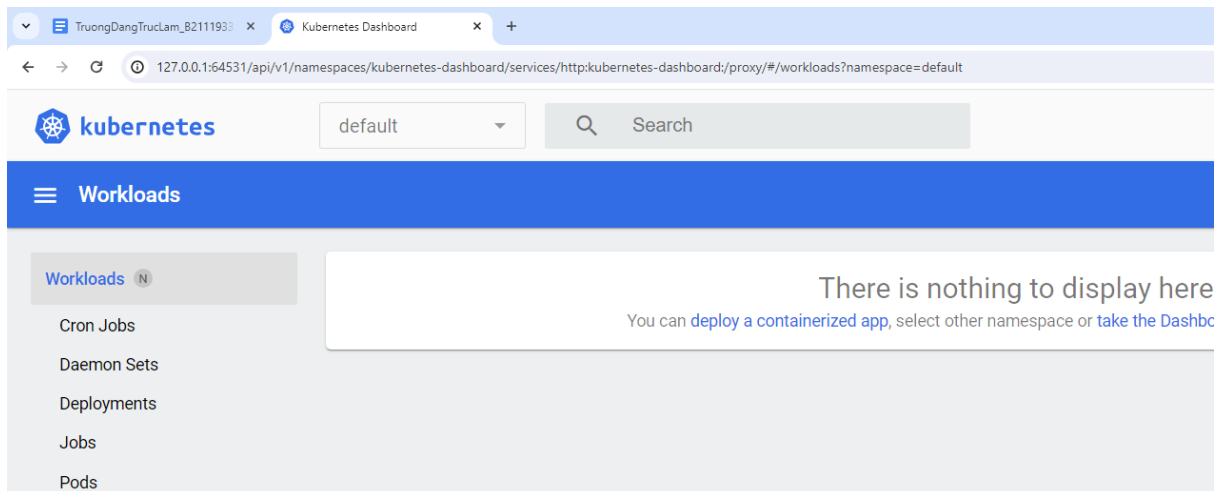
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> minikube dashboard
W0402 20:47:44.636569 17880 main.go:291] Unable to resolve the current Docker CLI context "default": context "default"
: context not found: open C:\Users\lambo\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a3
3f0688f\meta.json: The system cannot find the path specified.
* 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://127.0.0.1:64531/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in yo
ur default browser...
```

Open the Dashboard



Interface of the Dashboard

## 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
```

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl create deployment kubernetes-bootcamp --image=gcr.io/google
-samples/kubernetes-bootcamp:v1 --port=8080
deployment.apps/kubernetes-bootcamp created
PS C:\WINDOWS\system32>
```

Using `kubectl` to create a deployment

- To list your deployments  
`kubectl get deployments`  
(take a screenshot)

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get deployments
NAME                                READY    UP-TO-DATE    AVAILABLE    AGE
kubernetes-bootcamp                1/1      1              1            55s
PS C:\WINDOWS\system32>
```

List deployments

- Looking for existing Pods:

```
kubectl get pods
```

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get pods
NAME                                READY    STATUS    RESTARTS    AGE
kubernetes-bootcamp-7fb78bdcf8-dz64f 1/1      Running   0            72s
PS C:\WINDOWS\system32> $POD_NAME = "kubernetes-bootcamp-7fb78bdcf8-dz64f"
PS C:\WINDOWS\system32>
```

Looking for existing Pods and setup the Pod Name

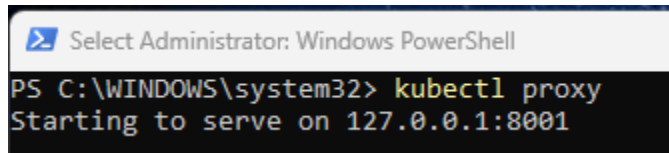
- To see details about the Pod's container

```
kubectl describe pods
```

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl describe pods
Name:                                kubernetes-bootcamp-7fb78bdcf8-dz64f
Namespace:                          default
Priority:                             0
Service Account:                     default
Node:                                minikube/192.168.59.101
Start Time:                          Wed, 03 Apr 2024 22:39:44 +0700
Labels:                              app=kubernetes-bootcamp
                                      pod-template-hash=7fb78bdcf8
Annotations:                          <none>
Status:                              Running
IP:                                  10.244.0.14
IPs:
  IP:                                10.244.0.14
Controlled By:                       ReplicaSet/kubernetes-bootcamp-7fb78bdcf8
Containers:
  kubernetes-bootcamp:
    Container ID:                     docker://01252a4202d5916c0445a703a7f79210dd5281ed0a1d63aa3e84450430593
    Image:                            gcr.io/google-samples/kubernetes-bootcamp:v1
    Image ID:                         docker-pullable://jocatalin/kubernetes-bootcamp@sha256:0d6b8ee63bb575b6156f446b3bc3b3c143d233037f3a2f00e279c8fcc64af
    Port:                             8080/TCP
```

See details about the Pod's container

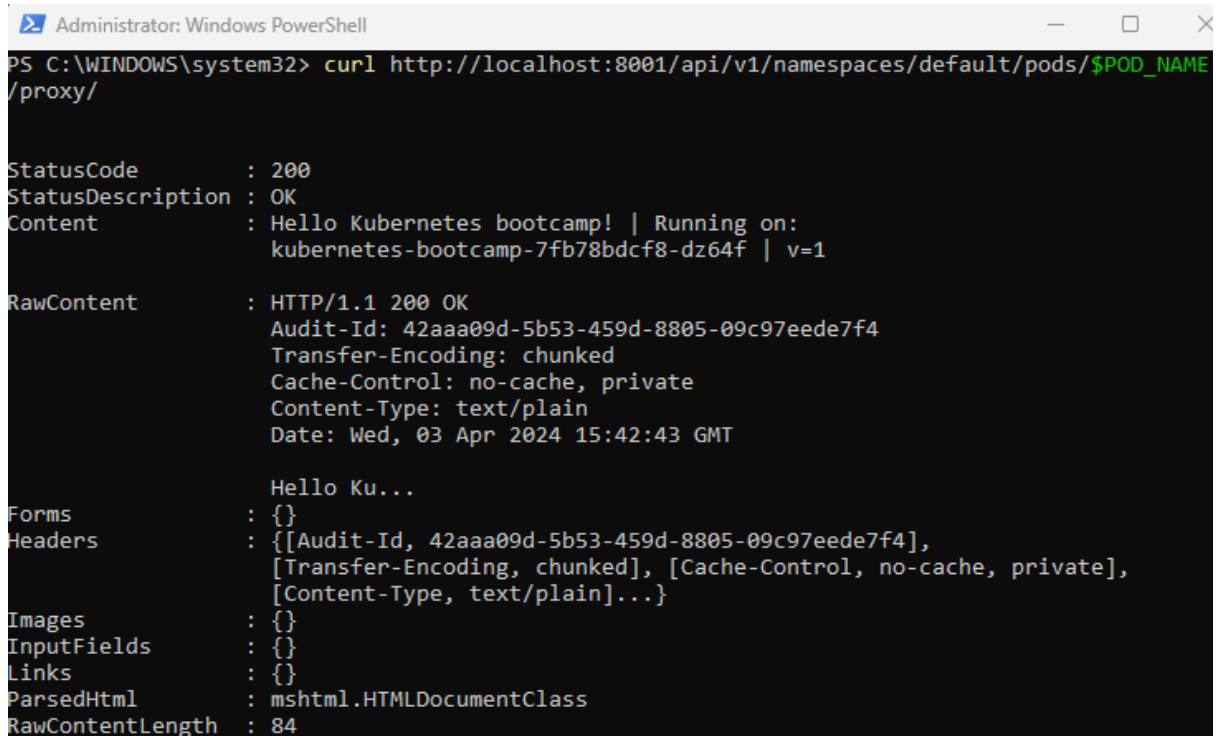
- Create a proxy that will forward communications into the cluster-wide private network  
kubectl proxy



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

Create a proxy

- Show your app in the terminal (or a web browser)  
curl  
[http://localhost:8001/api/v1/namespaces/default/pods/\\$POD\\_NAME:8080/proxy/](http://localhost:8001/api/v1/namespaces/default/pods/$POD_NAME:8080/proxy/)



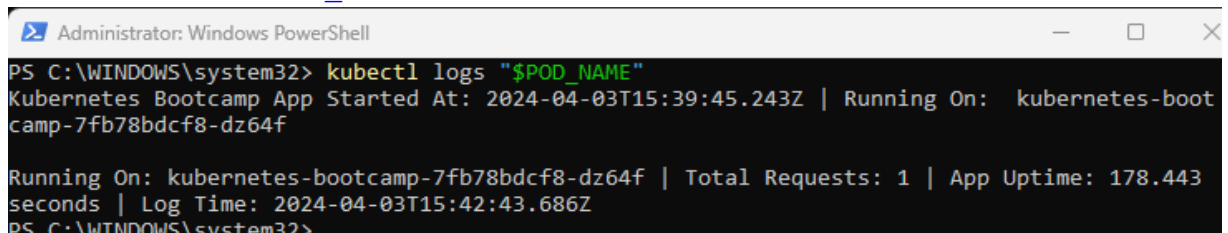
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> curl http://localhost:8001/api/v1/namespaces/default/pods/$POD_NAME:8080/proxy/

StatusCode      : 200
StatusDescription : OK
Content         : Hello Kubernetes bootcamp! | Running on:
                  kubernetes-bootcamp-7fb78bdcf8-dz64f | v=1
RawContent      : HTTP/1.1 200 OK
                  Audit-Id: 42aaa09d-5b53-459d-8805-09c97eede7f4
                  Transfer-Encoding: chunked
                  Cache-Control: no-cache, private
                  Content-Type: text/plain
                  Date: Wed, 03 Apr 2024 15:42:43 GMT

                  Hello Ku...
Forms           : {}
Headers         : [[Audit-Id, 42aaa09d-5b53-459d-8805-09c97eede7f4],
                  [Transfer-Encoding, chunked], [Cache-Control, no-cache, private],
                  [Content-Type, text/plain]...]
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 84
```

Display in the terminal

- View the container logs  
kubectl logs "\$POD\_NAME"



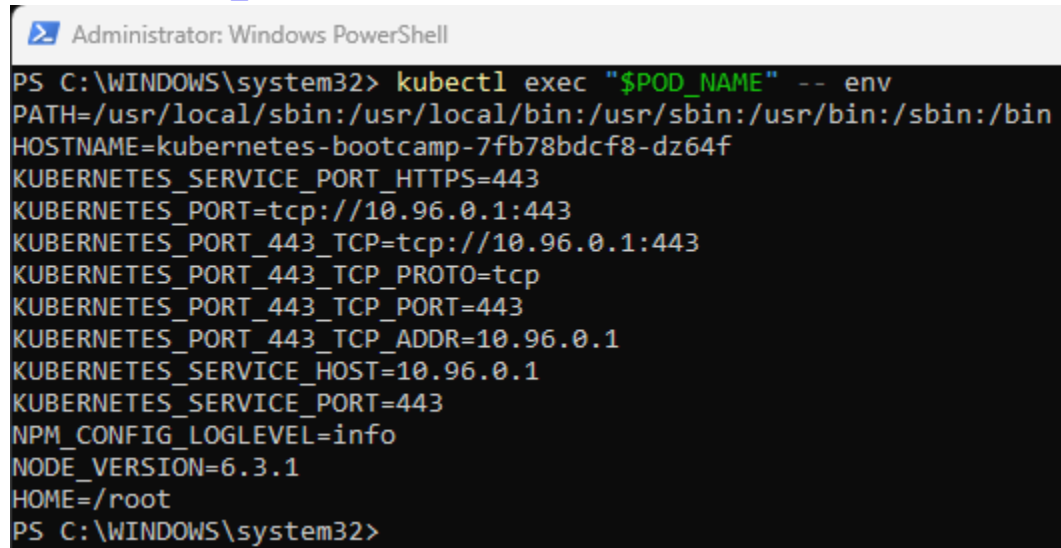
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl logs "$POD_NAME"
Kubernetes Bootcamp App Started At: 2024-04-03T15:39:45.243Z | Running On: kubernetes-boot
camp-7fb78bdcf8-dz64f

Running On: kubernetes-bootcamp-7fb78bdcf8-dz64f | Total Requests: 1 | App Uptime: 178.443
seconds | Log Time: 2024-04-03T15:42:43.686Z
PS C:\WINDOWS\system32>
```

View the container logs

- Execute command on the container

kubectl exec "\$POD\_NAME" -- env

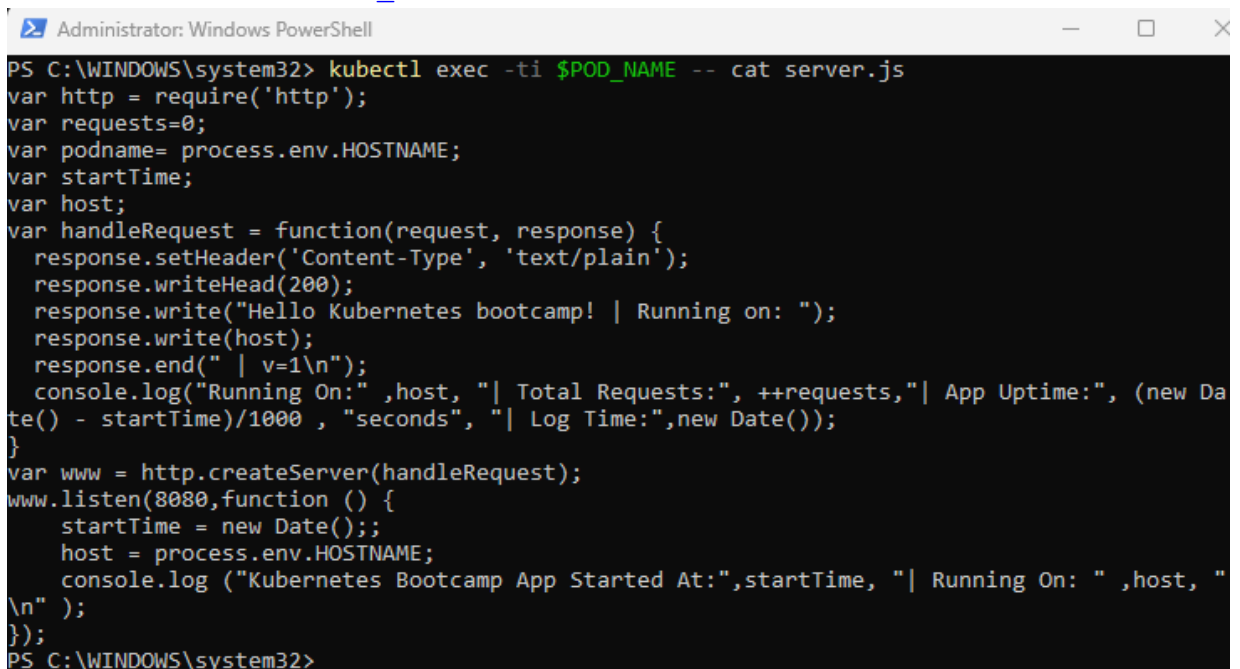


```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl exec "$POD_NAME" -- env
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=kubernetes-bootcamp-7fb78bdcf8-dz64f
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
KUBERNETES_PORT_443_TCP_PROTO=tcp
KUBERNETES_PORT_443_TCP_PORT=443
KUBERNETES_PORT_443_TCP_ADDR=10.96.0.1
KUBERNETES_SERVICE_HOST=10.96.0.1
KUBERNETES_SERVICE_PORT=443
NPM_CONFIG_LOGLEVEL=info
NODE_VERSION=6.3.1
HOME=/root
PS C:\WINDOWS\system32>
```

Execute command on the container

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

kubectl exec -ti \$POD\_NAME -- cat server.js



```
Administrator: Windows PowerShell
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 Time:",new Date());
}
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 source code of the app is in the server.js file

- View the application status

kubectl exec -ti \$POD\_NAME -- curl <http://localhost:8080>  
(take a screenshot)



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl exec -ti $POD_NAME -- curl http://localhost:8080
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7fb78bdcf8-dz64f | v=1
PS C:\WINDOWS\system32>
```

View the application status

### 2.3. Expose Your Application

- List the current services from our cluster:

```
kubectl get services
```

```
Administrator: Windows PowerShell
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    26h
PS C:\WINDOWS\system32>
```

List the current services from our cluster

- Create a new service and expose it to external traffic

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

Note: see [types of services](#) in Kubernetes

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

Create a new service and expose it to external traffic

- List the current services from our cluster again:

```
kubectl get services
```

(take a screenshot)

```
Administrator: Windows PowerShell
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    26h
kubernetes-bootcamp NodePort      10.96.55.122  <none>         8080:31526/TCP 41s
PS C:\WINDOWS\system32>
```

List the current services from our cluster again

- To see the service URL

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube service kubernetes-bootcamp --url
W0403 22:49:10.049131 10508 main.go:291] Unable to resolve the current Docker CLI context
"default": context "default": context not found: open C:\Users\lambo\.docker\contexts\meta
\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: The system can
not find the path specified.
http://192.168.59.101:31526
```

Display the service URL

- Access your service in the terminal (or a web browser)  
curl <Service URL>

(take a screenshot)

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> curl http://192.168.59.101:31526

StatusCode      : 200
StatusDescription : OK
Content         : Hello Kubernetes bootcamp! | Running on:
                  kubernetes-bootcamp-7fb78bdcf8-dz64f | v=1
RawContent      : HTTP/1.1 200 OK
                  Connection: keep-alive
                  Transfer-Encoding: chunked
                  Content-Type: text/plain
                  Date: Wed, 03 Apr 2024 15:50:07 GMT

                  Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7fb78...
Forms           : {}
Headers         : {[Connection, keep-alive], [Transfer-Encoding, chunked],
                  [Content-Type, text/plain], [Date, Wed, 03 Apr 2024 15:50:07 GMT]}
Images          : {}
InputFields     : {}
Links           : {}
ParsedHtml      : mshtml.HTMLDocumentClass
RawContentLength : 84
```

Access the service via the terminal

## 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           12h
PS C:\WINDOWS\system32> kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-7fb78bdcf8-dz64f 1/1     Running   1 (11h ago) 12h
PS C:\WINDOWS\system32>
```

List resources

(take a screenshot)

- To see the ReplicaSet created by the Deployment

kubectl get rs

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get rs
NAME                                DESIRED    CURRENT    READY    AGE
kubernetes-bootcamp-7fb78bdcf8      1          1          1        12h
PS C:\WINDOWS\system32>
```

Display the ReplicaSet created by the Deployment

- Scale the Deployment to 4 replicas

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

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

Scale the Deployment to 4 replicas

- View your Deployments once again

kubectl get deployments

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get deployments
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
kubernetes-bootcamp 4/4      4              4            12h
PS C:\WINDOWS\system32>
```

Now it has 4 replicas

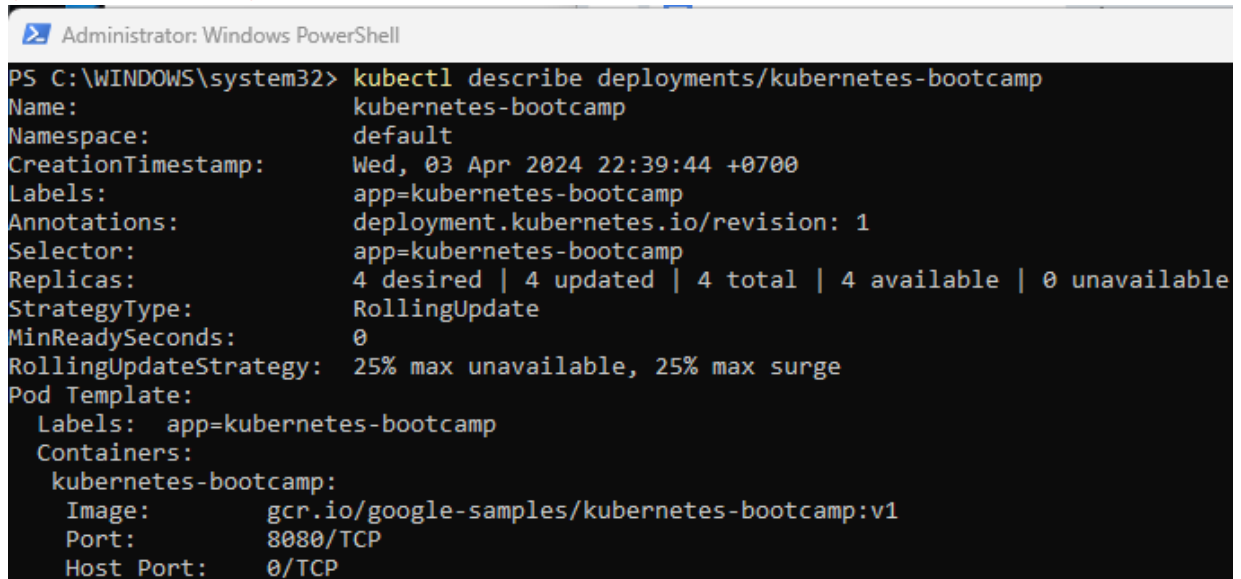
kubectl get pods -o wide

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get pods -o wide
NAME                                READY    STATUS    RESTARTS   AGE    IP            NODE     NOMINATED NODE
kubernetes-bootcamp-7fb78bdcf8-4blxc 1/1      Running   0           2m15s  10.244.0.19   minikube <none>
kubernetes-bootcamp-7fb78bdcf8-dqnsz 1/1      Running   0           2m15s  10.244.0.21   minikube <none>
kubernetes-bootcamp-7fb78bdcf8-dz64f 1/1      Running   1 (12h ago) 12h     10.244.0.16   minikube <none>
kubernetes-bootcamp-7fb78bdcf8-j76zm 1/1      Running   0           2m15s  10.244.0.20   minikube <none>
PS C:\WINDOWS\system32>
```

The same as

(take a screenshot)

kubectl describe deployments/kubernetes-bootcamp  
(take a screenshot)

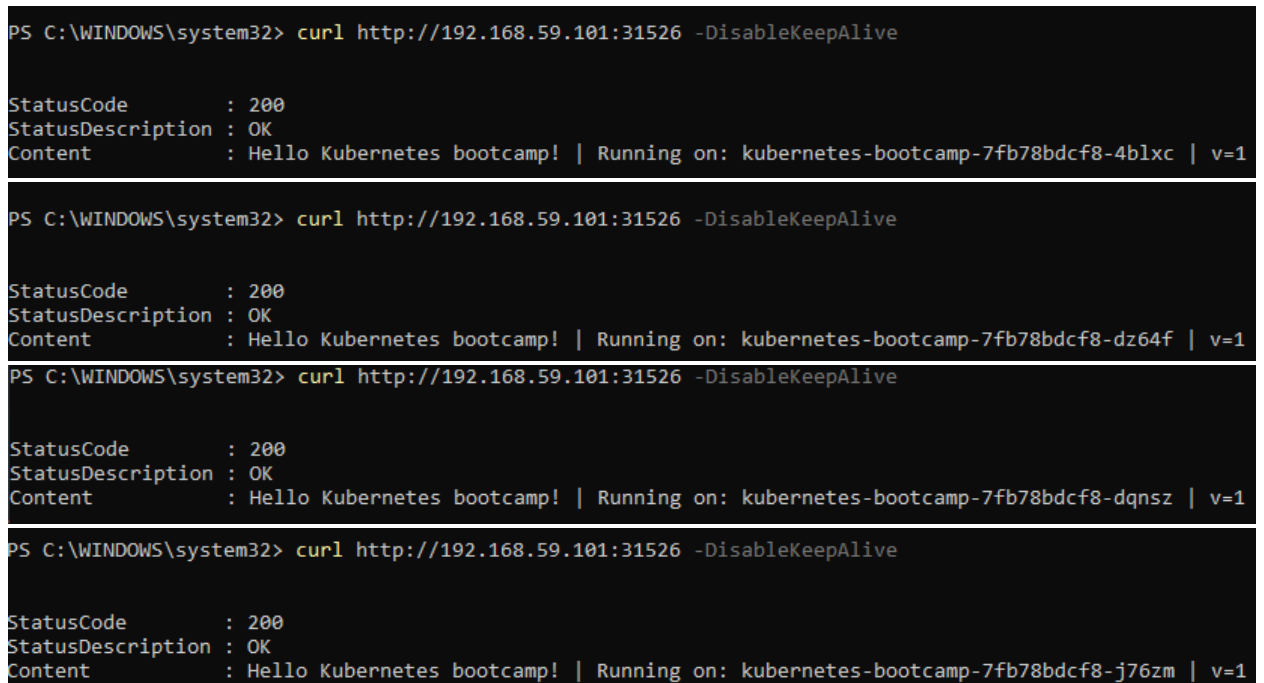


```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl describe deployments/kubernetes-bootcamp
Name: kubernetes-bootcamp
Namespace: default
CreationTimestamp: Wed, 03 Apr 2024 22:39:44 +0700
Labels: app=kubernetes-bootcamp
Annotations: deployment.kubernetes.io/revision: 1
Selector: app=kubernetes-bootcamp
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: 8080/TCP
      Host Port: 0/TCP
```

Display the description again

- 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://192.168.59.101:31526 -DisableKeepAlive

StatusCode      : 200
StatusDescription : OK
Content         : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7fb78bdcf8-4blxc | v=1

PS C:\WINDOWS\system32> curl http://192.168.59.101:31526 -DisableKeepAlive

StatusCode      : 200
StatusDescription : OK
Content         : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7fb78bdcf8-dz64f | v=1

PS C:\WINDOWS\system32> curl http://192.168.59.101:31526 -DisableKeepAlive

StatusCode      : 200
StatusDescription : OK
Content         : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7fb78bdcf8-dqnsz | v=1

PS C:\WINDOWS\system32> curl http://192.168.59.101:31526 -DisableKeepAlive

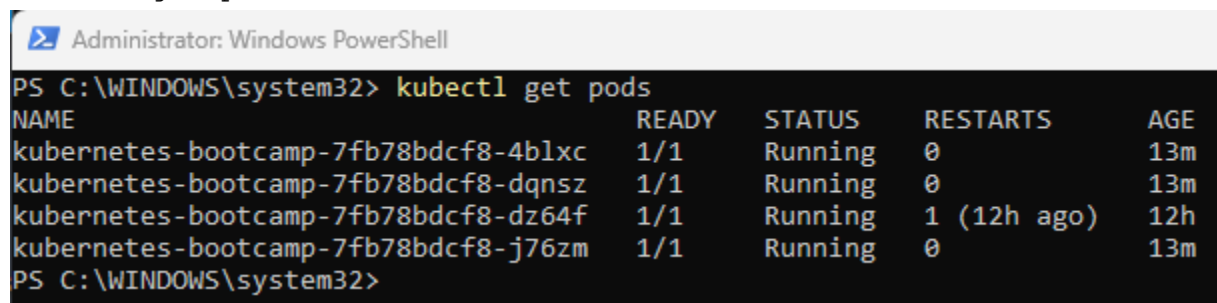
StatusCode      : 200
StatusDescription : OK
Content         : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-7fb78bdcf8-j76zm | v=1
```

We hit a different Pod with every request

## 2.5. Update your application

- To view the current image version of the app

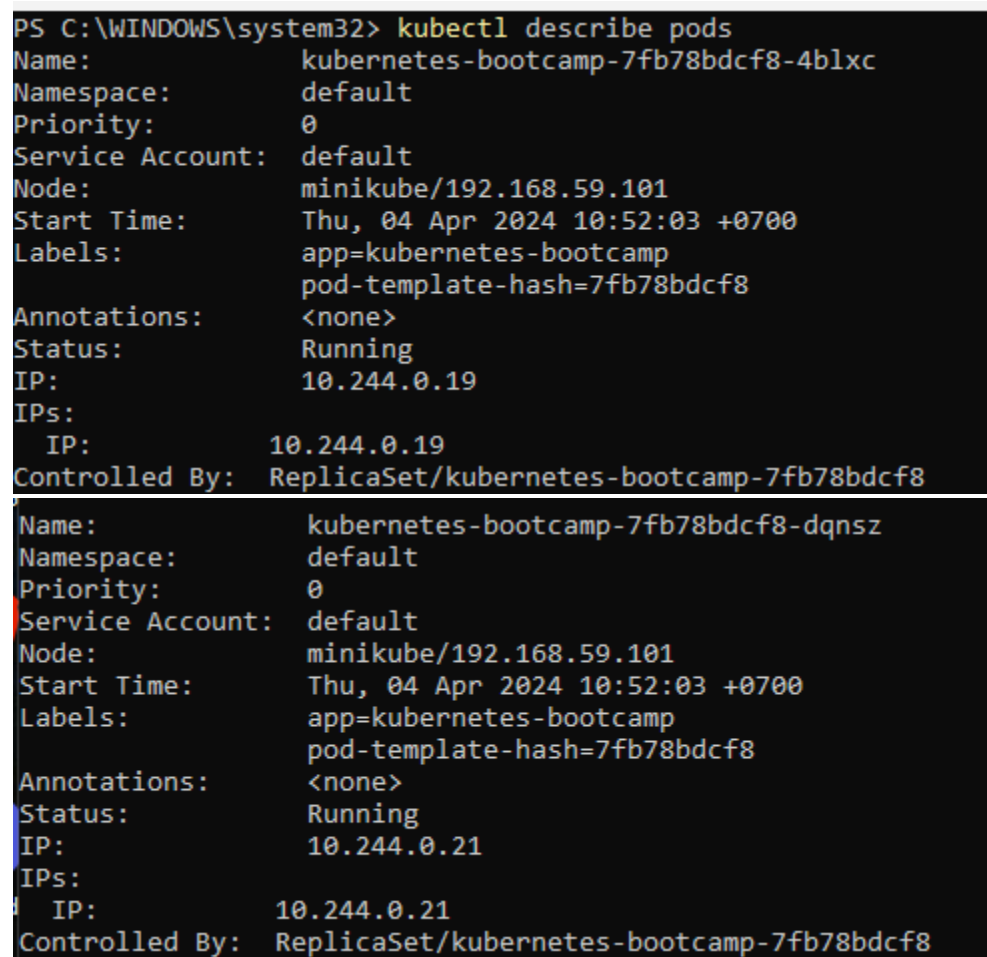
```
kubectl get pods
```



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-7fb78bdcf8-4blxc 1/1     Running   0           13m
kubernetes-bootcamp-7fb78bdcf8-dqnsz 1/1     Running   0           13m
kubernetes-bootcamp-7fb78bdcf8-dz64f 1/1     Running   1 (12h ago) 12h
kubernetes-bootcamp-7fb78bdcf8-j76zm 1/1     Running   0           13m
PS C:\WINDOWS\system32>
```

Display pods

```
kubectl describe pods
```



```
PS C:\WINDOWS\system32> kubectl describe pods
Name:                                kubernetes-bootcamp-7fb78bdcf8-4blxc
Namespace:                          default
Priority:                            0
Service Account:                    default
Node:                               minikube/192.168.59.101
Start Time:                         Thu, 04 Apr 2024 10:52:03 +0700
Labels:                             app=kubernetes-bootcamp
                                     pod-template-hash=7fb78bdcf8
Annotations:                         <none>
Status:                             Running
IP:                                 10.244.0.19
IPs:
  IP:                                10.244.0.19
Controlled By:                      ReplicaSet/kubernetes-bootcamp-7fb78bdcf8

Name:                                kubernetes-bootcamp-7fb78bdcf8-dqnsz
Namespace:                          default
Priority:                            0
Service Account:                    default
Node:                               minikube/192.168.59.101
Start Time:                         Thu, 04 Apr 2024 10:52:03 +0700
Labels:                             app=kubernetes-bootcamp
                                     pod-template-hash=7fb78bdcf8
Annotations:                         <none>
Status:                             Running
IP:                                 10.244.0.21
IPs:
  IP:                                10.244.0.21
Controlled By:                      ReplicaSet/kubernetes-bootcamp-7fb78bdcf8
```

Display the description of pods again

- Update the image of the application to version 2

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=
docker.io/jocatalin/kubernetes-bootcamp:v2
deployment.apps/kubernetes-bootcamp image updated
PS C:\WINDOWS\system32>
```

kubernetes-bootcamp image updated

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-5f9578c84f-d9hq1 1/1     Running   0          3m11s
kubernetes-bootcamp-5f9578c84f-m56vm 1/1     Running   0          3m12s
kubernetes-bootcamp-5f9578c84f-rf22j 1/1     Running   0          3m17s
kubernetes-bootcamp-5f9578c84f-sw8k6 1/1     Running   0          3m17s
PS C:\WINDOWS\system32>
```

Check the status of the new Pods

- Roll back the deployment to your last working version  
kubectl rollout undo deployments/kubernetes-bootcamp

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

Roll back

- To view the current image version of the app  
kubectl get pods

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-7fb78bdcf8-bwkqh 1/1     Running   0          70s
kubernetes-bootcamp-7fb78bdcf8-j9mlj 1/1     Running   0          69s
kubernetes-bootcamp-7fb78bdcf8-khx9q 1/1     Running   0          70s
kubernetes-bootcamp-7fb78bdcf8-mjqcf 1/1     Running   0          69s
PS C:\WINDOWS\system32>
```

View the current image version of the app

kubectl describe pods  
(take a screenshot)

```
PS C:\WINDOWS\system32> kubectl describe pods
Name:      kubernetes-bootcamp-7fb78bdcf8-bwkqh
Namespace: default
Priority:   0
Service Account: default
Node:      minikube/192.168.59.101
Start Time: Thu, 04 Apr 2024 11:12:47 +0700
Labels:    app=kubernetes-bootcamp
           pod-template-hash=7fb78bdcf8
Annotations: <none>
Status:     Running
IP:         10.244.0.26
IPs:
  IP:      10.244.0.26
Controlled By: ReplicaSet/kubernetes-bootcamp-7fb78bdcf8
Containers:
  kubernetes-bootcamp:
    Container ID:  docker://457b5e40401cc012b900e159ed8c45d52a75fb209aa6d8db11f2122487dfcb0d

Name:      kubernetes-bootcamp-7fb78bdcf8-j9m1j
Namespace: default
Priority:   0
Service Account: default
Node:      minikube/192.168.59.101
Start Time: Thu, 04 Apr 2024 11:12:48 +0700
Labels:    app=kubernetes-bootcamp
           pod-template-hash=7fb78bdcf8
Annotations: <none>
Status:     Running
IP:         10.244.0.28
IPs:
  IP:      10.244.0.28
Controlled By: ReplicaSet/kubernetes-bootcamp-7fb78bdcf8
Containers:
  kubernetes-bootcamp:
    Container ID:  docker://761278bfb94aa9844c22f2d7120a917dd36817933175d186958cbe30779e8e7c
```

Display the description of pods again

## 2.6. Cleaning up

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

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

```
Administrator: Windows PowerShell
```

```
PS C:\WINDOWS\system32> kubectl delete service kubernetes-bootcamp
service "kubernetes-bootcamp" deleted
PS C:\WINDOWS\system32>
```

Delete the service

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

```
Administrator: Windows PowerShell
```

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

Delete deployments

- Stop the Minikube cluster

minikube stop

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube stop
W0404 11:19:11.127656 11632 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": context not found: open C:\Users\lambo\.docker\contexts\meta\37a8e6c1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: The system cannot find the path specified.
* Stopping node "minikube" ...
* 1 node stopped.
PS C:\WINDOWS\system32>
```

Stop the Minikube cluster

- Delete the Minikube VM (optional)

minikube delete

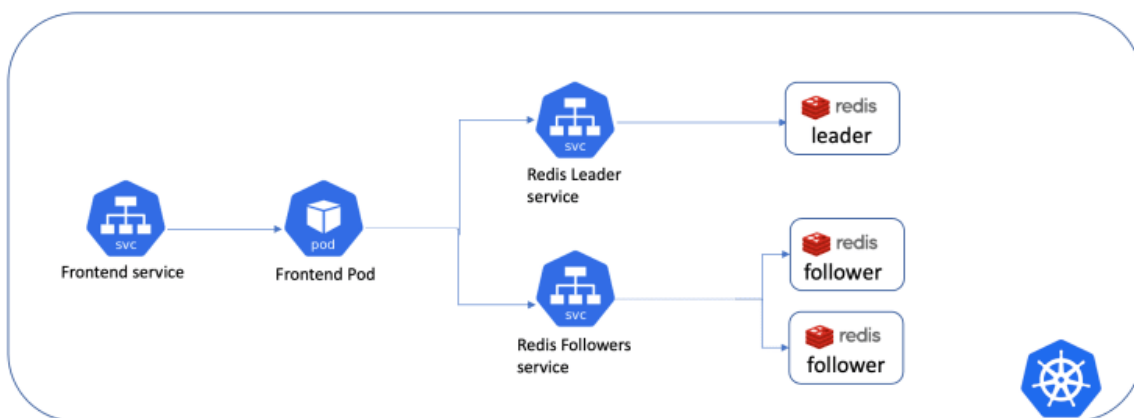
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> minikube delete
W0404 11:21:06.895553 8 main.go:291] Unable to resolve the current Docker CLI context "default": context "default": context not found: open C:\Users\lambo\.docker\contexts\meta\37a8e6c1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a33f0688f\meta.json: The system cannot find the path specified.
* Deleting "minikube" in virtualbox ...
* Removed all traces of the "minikube" cluster.
PS C:\WINDOWS\system32>
```

Delete the Minikube VM

### 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

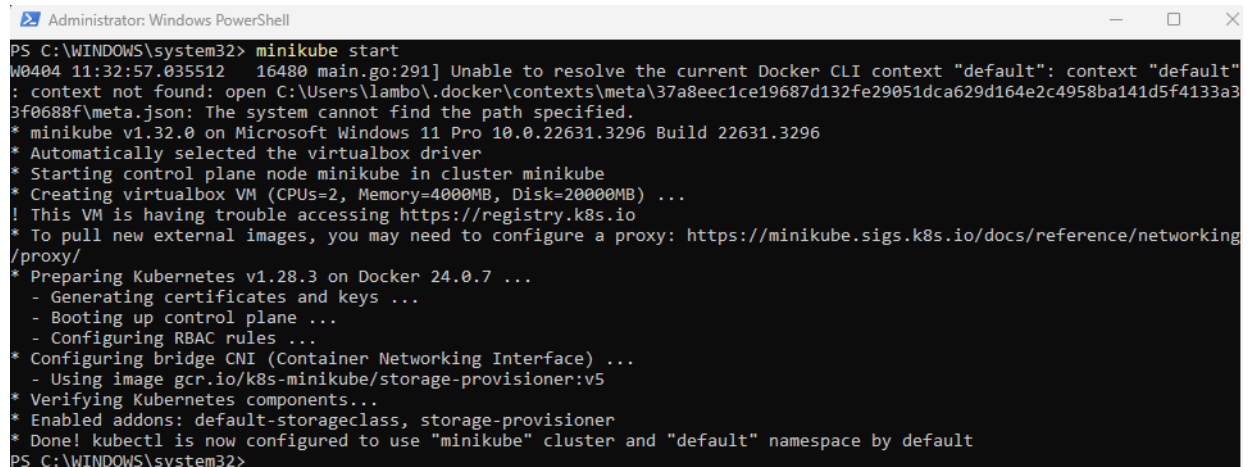




### 3.1. Start up the Redis Database

The guestbook application uses Redis to store its data.

#### a. Creating the Redis Deployment

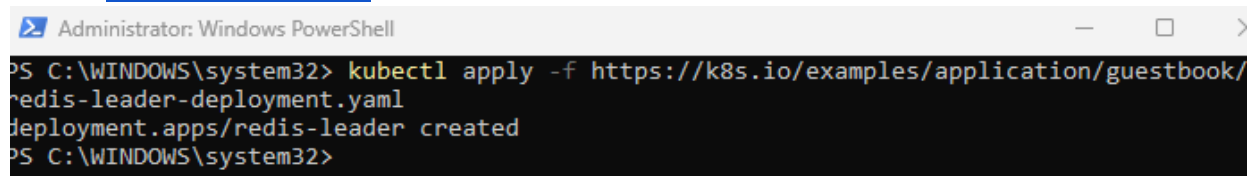


```

PS C:\WINDOWS\system32> minikube start
W0404 11:32:57.035512 16480 main.go:291] Unable to resolve the current Docker CLI context "default": context "default"
: context not found: open C:\Users\lambo\.docker\contexts\meta\37a8eec1ce19687d132fe29051dca629d164e2c4958ba141d5f4133a3
3f0688f\meta.json: The system cannot find the path specified.
* minikube v1.32.0 on Microsoft Windows 11 Pro 10.0.22631.3296 Build 22631.3296
* Automatically selected the virtualbox driver
* Starting control plane node minikube in cluster minikube
* Creating virtualbox VM (CPUs=2, Memory=4000MB, 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/
* Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
  - 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: default-storageclass, storage-provisioner
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
PS C:\WINDOWS\system32>
    
```

Create new cluster

- Apply the Redis Deployment from the redis-leader-deployment.yaml file:  
`kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-deployment.yaml`

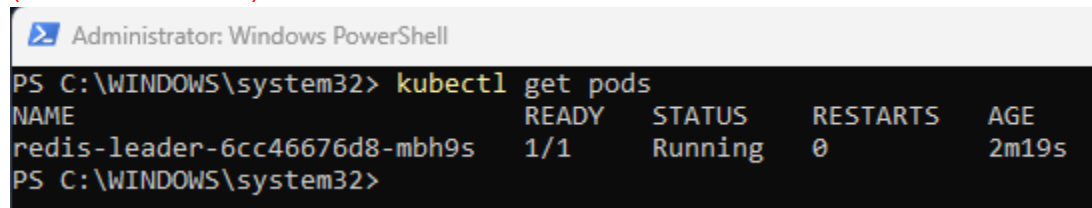


```

PS C:\WINDOWS\system32> kubectl apply -f https://k8s.io/examples/application/guestbook/redis-leader-deployment.yaml
deployment.apps/redis-leader created
PS C:\WINDOWS\system32>
    
```

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

- Query the list of Pods to verify that the Redis Pod is running:  
`kubectl get pods`  
 (take a screenshot)



```

PS C:\WINDOWS\system32> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
redis-leader-6cc46676d8-mbh9s      1/1     Running   0           2m19s
PS C:\WINDOWS\system32>
    
```

Verify that the Redis Pod is running

- Run the following command to view the logs from the Redis leader Pod:  
`kubectl logs -f deployment/redis-leader`

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl logs -f deployment/redis-leader
1:C 04 Apr 2024 04:35:01.109 # o000o000o000o Redis is starting o000o000o000o
1:C 04 Apr 2024 04:35:01.109 # Redis version=6.0.5, bits=64, commit=00000000, modified=
0, pid=1, just started
1:C 04 Apr 2024 04:35:01.109 # Warning: no config file specified, using the default con
fig. In order to specify a config file use redis-server /path/to/redis.conf
1:M 04 Apr 2024 04:35:01.109 * Running mode=standalone, port=6379.
1:M 04 Apr 2024 04:35:01.110 # Server initialized
1:M 04 Apr 2024 04:35:01.110 # WARNING you have Transparent Huge Pages (THP) support en
abled in your kernel. This will create latency and memory usage issues with Redis. To f
ix this issue run 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 a rebo
ot. Redis must be restarted after THP is disabled.
1:M 04 Apr 2024 04:35:01.110 * Ready to accept connections
```

View the logs from the Redis leader Pod

### 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>

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl apply -f https://k8s.io/examples/application/guestbook/
redis-leader-service.yaml
service/redis-leader created
PS C:\WINDOWS\system32>
```

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

- Query the list of Services to verify that the Redis Service is running:  
kubectl get services  
(take a screenshot)

```
Administrator: Windows PowerShell
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          8m37s
redis-leader         ClusterIP   10.108.71.194    <none>            6379/TCP         111s
PS C:\WINDOWS\system32>
```

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

### 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>

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-deployment.yaml
deployment.apps/redis-follower created
PS C:\WINDOWS\system32>
```

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

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

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
redis-follower-7dddf7c979-lpplm     1/1     Running   0           52s
redis-follower-7dddf7c979-nh8wq     1/1     Running   0           52s
redis-leader-6cc46676d8-mbh9s       1/1     Running   0           14m
PS C:\WINDOWS\system32>
```

Verify that the two Redis follower replicas are running by querying the list of Pods  
(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>

```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl apply -f https://k8s.io/examples/application/guestbook/redis-follower-service.yaml
service/redis-follower created
PS C:\WINDOWS\system32>
```

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

- Query the list of Services to verify that the Redis Service is running:  
kubectl get services  
(take a screenshot)

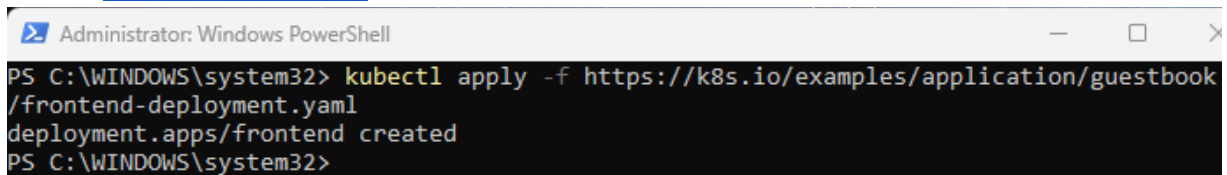
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get services
NAME             TYPE        CLUSTER-IP      EXTERNAL-IP  PORT(S)    AGE
kubernetes       ClusterIP   10.96.0.1       <none>       443/TCP    48m
redis-follower    ClusterIP   10.104.66.28    <none>       6379/TCP   31m
redis-leader      ClusterIP   10.108.71.194   <none>       6379/TCP   41m
PS C:\WINDOWS\system32>
```

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

### 3.2. Set up and Expose the Guestbook Frontend

#### a. Creating the Guestbook Frontend Deployment

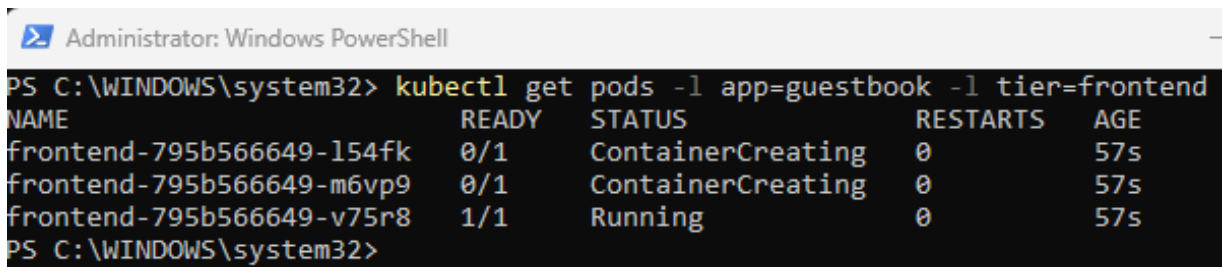
- Apply the frontend Deployment from the frontend-deployment.yaml file:  
kubectl apply -f <https://k8s.io/examples/application/guestbook/frontend-deployment.yaml> -f



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-deployment.yaml
deployment.apps/frontend created
PS C:\WINDOWS\system32>
```

Apply the frontend Deployment from the frontend-deployment.yaml file

- Query the list of Pods to verify that the three frontend replicas are running:  
kubectl get pods -l app=guestbook -l tier=frontend



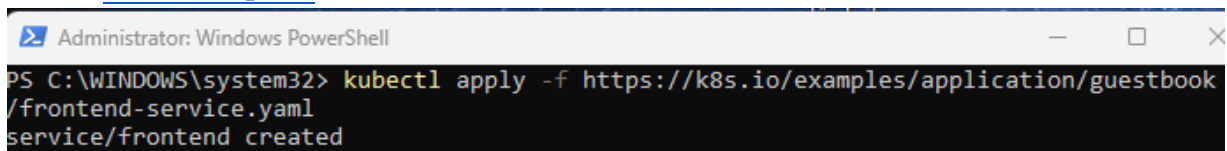
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get pods -l app=guestbook -l tier=frontend
NAME                                READY   STATUS             RESTARTS   AGE
frontend-795b566649-154fk           0/1     ContainerCreating   0           57s
frontend-795b566649-m6vp9           0/1     ContainerCreating   0           57s
frontend-795b566649-v75r8           1/1     Running              0           57s
PS C:\WINDOWS\system32>
```

Query the list of Pods to verify that the three frontend replicas are running

(take a screenshot)

#### b. Creating the Frontend Service

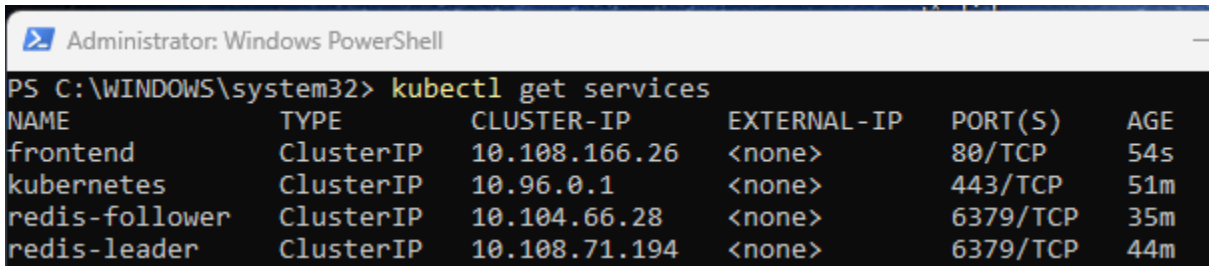
- Apply the frontend Service from the frontend-service.yaml file:  
kubectl apply -f <https://k8s.io/examples/application/guestbook/frontend-service.yaml> -f



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl apply -f https://k8s.io/examples/application/guestbook/frontend-service.yaml
service/frontend created
PS C:\WINDOWS\system32>
```

Apply the frontend Service from the frontend-service.yaml file

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



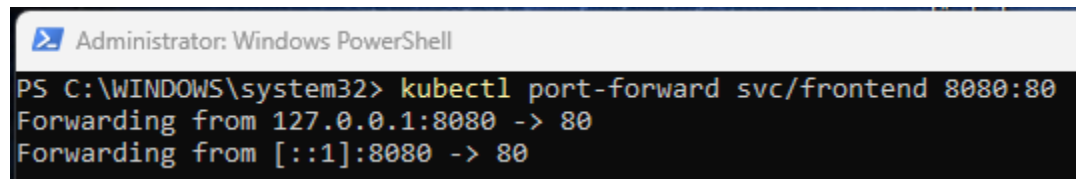
```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl get services
NAME           TYPE        CLUSTER-IP    EXTERNAL-IP  PORT(S)    AGE
frontend       ClusterIP   10.108.166.26 <none>       80/TCP     54s
kubernetes     ClusterIP   10.96.0.1     <none>       443/TCP    51m
redis-follower  ClusterIP   10.104.66.28  <none>       6379/TCP   35m
redis-leader    ClusterIP   10.108.71.194 <none>       6379/TCP   44m
```

Query the list of Services to verify that the frontend Service is running  
(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.

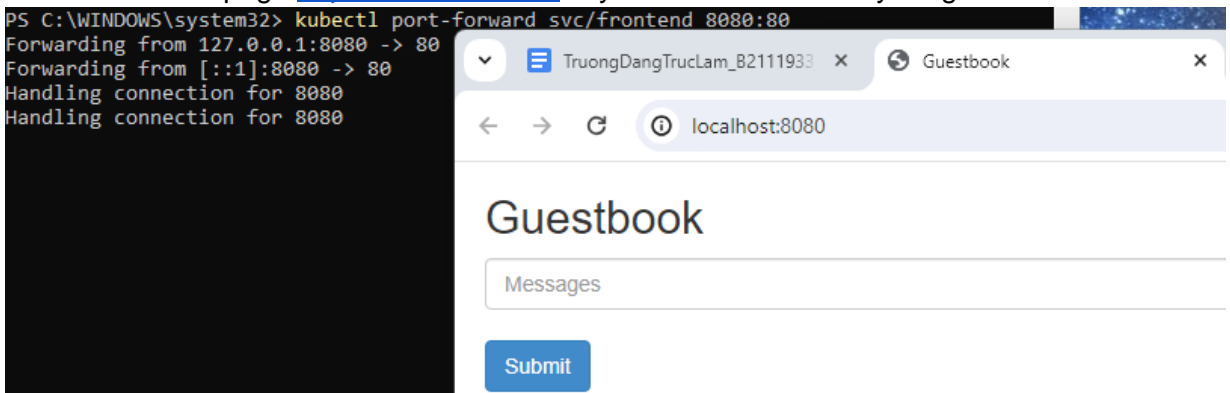
`kubectl port-forward svc/frontend 8080:80`



```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl port-forward svc/frontend 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

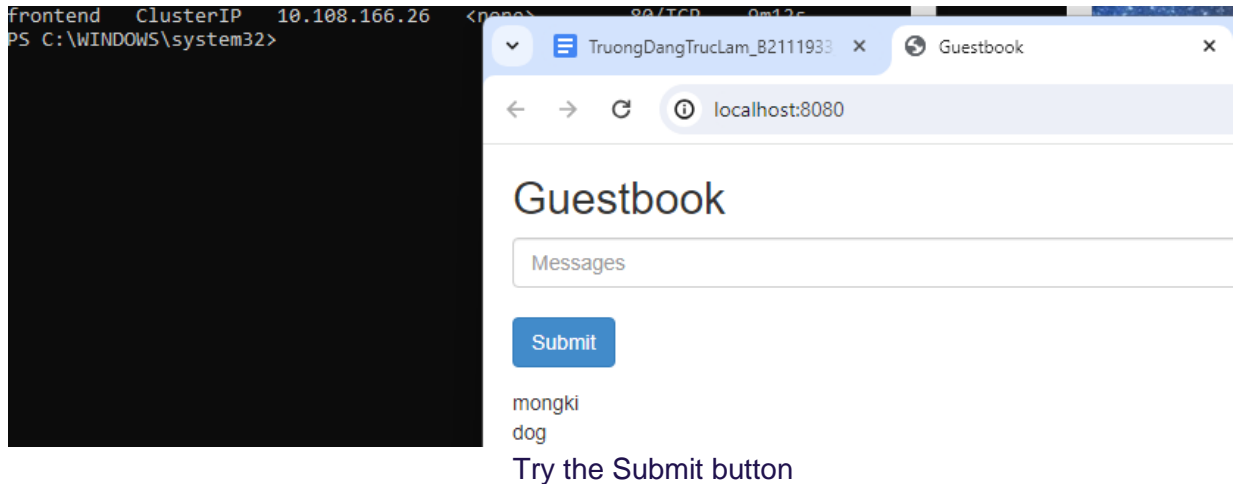
Forward port 8080 on your local machine to port 80 on the service

- Load the page <http://localhost:8080> in your browser to view your guestbook.



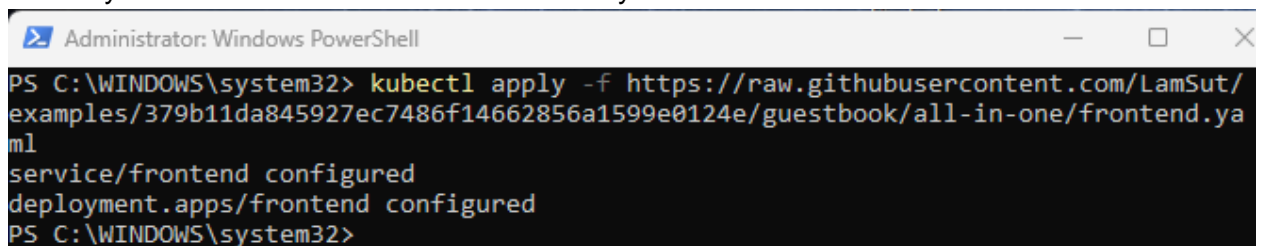
Display the guestbook

(take a screenshot)

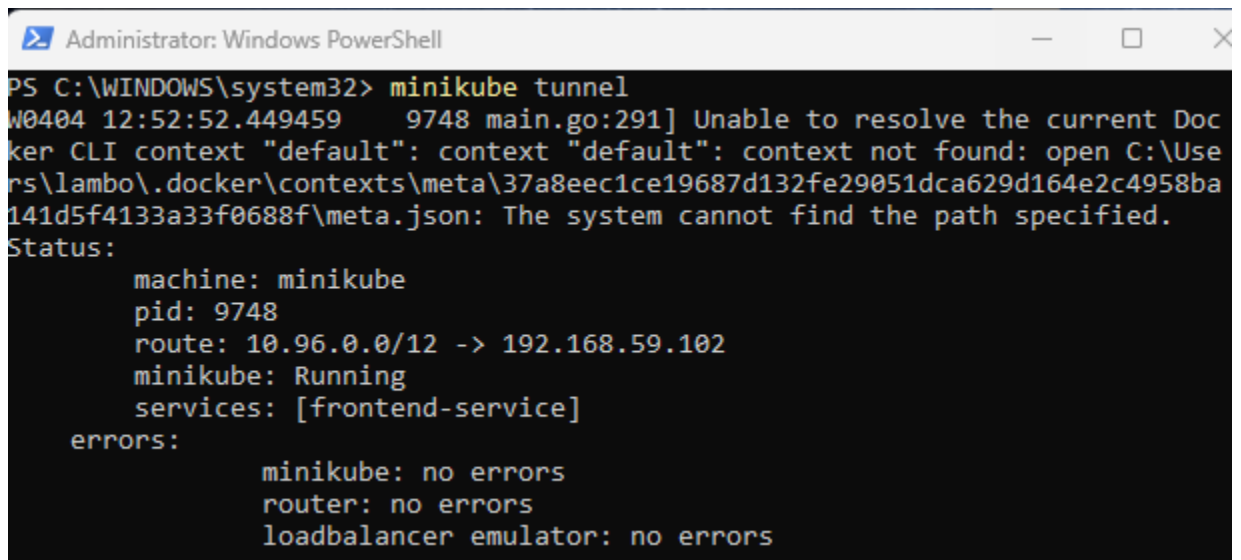


d. Viewing the Frontend Service via LoadBalancer

- If you deployed the `frontend-service.yaml` manifest with type: LoadBalancer you need to find the IP address to view your Guestbook.



This is the [link](#) of `frontend-service.yaml` manifest with type: LoadBalancer



Expose the external IP directly to any program running on the host operating system

- Copy the external IP address, and load the page in your browser to view your guestbook.

The terminal window shows the output of the command `kubectl get services`:

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
frontend	LoadBalancer	10.108.166.26	10.108.166.26	80:32537/TCP	54m
frontend-service	LoadBalancer	10.100.210.10	10.100.210.10	80:31362/TCP	1m
kubernetes	ClusterIP				84m
redis-follower	ClusterIP				8m
redis-leader	ClusterIP				8m

The browser window shows the 'Guestbook' application at the external IP address 10.108.166.26. The page has a title 'Guestbook', a text input field labeled 'Messages', a 'Submit' button, and two lines of text: 'mongki' and 'dog'.

Display Guestbook via LoadBalancer

### 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`

The terminal window shows the execution of the following commands:

```
PS C:\WINDOWS\system32> kubectl scale deployment frontend --replicas=5
deployment.apps/frontend scaled
PS C:\WINDOWS\system32> kubectl get pods
```

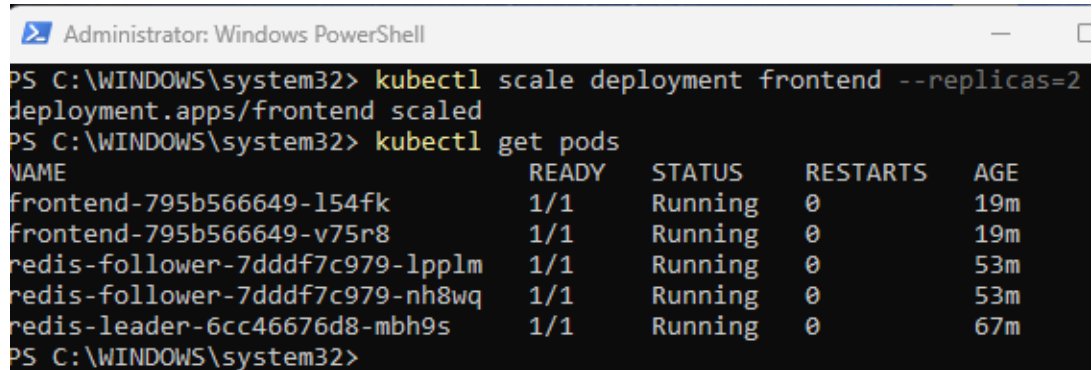
The output of `kubectl get pods` is as follows:

NAME	READY	STATUS	RESTARTS	AGE
frontend-795b566649-2lc8m	1/1	Running	0	13s
frontend-795b566649-4j8ss	1/1	Running	0	13s
frontend-795b566649-l54fk	1/1	Running	0	18m
frontend-795b566649-m6vp9	1/1	Running	0	18m
frontend-795b566649-v75r8	1/1	Running	0	18m
redis-follower-7dddf7c979-lpplm	1/1	Running	0	52m
redis-follower-7dddf7c979-nh8wq	1/1	Running	0	52m
redis-leader-6cc46676d8-mbh9s	1/1	Running	0	66m

Scale up the number of frontend Pods

- Run the following command to scale down the number of frontend Pods:  
`kubectl scale deployment frontend --replicas=2`  
`kubectl get pods`





```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl scale deployment frontend --replicas=2
deployment.apps/frontend scaled
PS C:\WINDOWS\system32> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
frontend-795b566649-154fk           1/1     Running   0           19m
frontend-795b566649-v75r8           1/1     Running   0           19m
redis-follower-7dddf7c979-lpplm     1/1     Running   0           53m
redis-follower-7dddf7c979-nh8wq     1/1     Running   0           53m
redis-leader-6cc46676d8-mbh9s       1/1     Running   0           67m
PS C:\WINDOWS\system32>
```

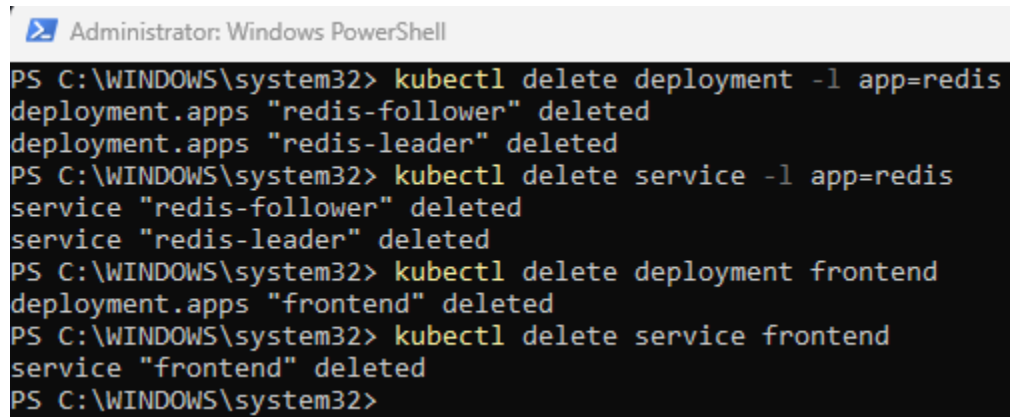
Scale down the number of frontend Pods

(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
```

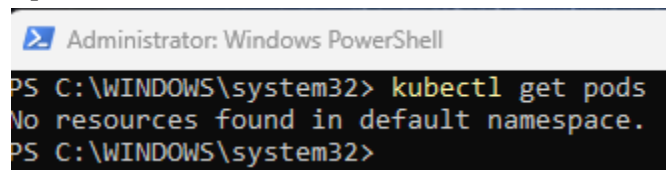


```
Administrator: Windows PowerShell
PS C:\WINDOWS\system32> kubectl delete deployment -l app=redis
deployment.apps "redis-follower" deleted
deployment.apps "redis-leader" deleted
PS C:\WINDOWS\system32> kubectl delete service -l app=redis
service "redis-follower" deleted
service "redis-leader" deleted
PS C:\WINDOWS\system32> kubectl delete deployment frontend
deployment.apps "frontend" deleted
PS C:\WINDOWS\system32> kubectl delete service frontend
service "frontend" deleted
PS C:\WINDOWS\system32>
```

Delete all Pods, Deployments, and Services

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

```
kubectl get pods
```



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

There is no Pod left

(take a screenshot)

---END---