

Lab6

Hello Minikube

Installation

1. Download minikube installer and run it:

Click on the buttons that describe your target platform. For other architectures, see [the release page](#) for a complete list of minikube binaries.

Operating system	<input type="button" value="Linux"/>	<input type="button" value="macOS"/>	<input type="button" value="Windows"/>
Architecture	<input type="button" value="x86-64"/>		
Release type	<input type="button" value="Stable"/>		
Installer type	<input type="button" value=".exe download"/>	<input type="button" value="Windows Package Manager"/>	<input type="button" value="Chocolatey"/>

To install the latest minikube **stable** release on **x86-64 Windows** using **.exe download**:

1. Download and run the installer for the [latest release](#).



2. Download kicbase.tar and load it into docker:

The screenshot shows a GitHub release page for 'minikube' with the URL github.com/kubernetes/minikube/releases. The 'Assets' section lists several files, including 'kicbase-v0.0.48-amd64.tar', which is highlighted with a red box.

Asset	SHA256	Size	Last Updated
dockermachine-driver-hyperkit	sha256:f73f6b8e81137faf...	22.5 MB	3 weeks ago
dockermachine-driver-hyperkit.sha256	sha256:441f460f2c86f471d...	65 Bytes	3 weeks ago
dockermachine-driver-kvm2	sha256:5435e1867bf3c169c...	15.2 MB	2 weeks ago
dockermachine-driver-kvm2-1.37.0-0-x86_64.rpm	sha256:9f1b7d88ea34e6455...	4.37 MB	2 weeks ago
dockermachine-driver-kvm2-amd64	sha256:5435e1867bf3c169c...	15.2 MB	2 weeks ago
dockermachine-driver-kvm2-amd64.sha256	sha256:e40cd9deb59aa9ad...	65 Bytes	2 weeks ago
dockermachine-driver-kvm2-x86_64	sha256:5435e1867bf3c169c...	15.2 MB	2 weeks ago
dockermachine-driver-kvm2.sha256	sha256:e40cd9deb59aa9ad...	65 Bytes	2 weeks ago
dockermachine-driver-kvm2_1.37.0-0_amd64.deb	sha256:aae1fd3b6fc2c20...	6.97 MB	2 weeks ago
kicbase-v0.0.48-amd64.tar	sha256:c4048676fa3f37de9...	1.22 GB	3 weeks ago
Source code (zip)			3 weeks ago
Source code (tar.gz)			3 weeks ago

```
PS C:\study\云计算导论\Labs\Lab6> docker load -i ./kicbase.tar
Loaded image: kicbase/stable:v0.0.48
```

We can find it in docker desktop:



3. Start minikube with cn mirror:

```
minikube start --driver=docker --image-mirror-country cn
```

```
PS C:\Windows\system32> minikube start --driver=docker --image-mirror-country cn
* Microsoft Windows 11 Home China 10.0.26100.6584 Build 26100.6584 上的 minikube v1.37.0
* 根据用户配置使用 docker 驱动程序
* 正在使用镜像存储库 registry.cn-hangzhou.aliyuncs.com/google_containers
* 使用具有 root 权限的 Docker Desktop 驱动程序
* 在集群中 "minikube" 启动节点 "minikube" primary control-plane
* 正在拉取基础镜像 v0.0.48 ...
! minikube was unable to download registry.cn-hangzhou.aliyuncs.com/google_containers/kicbase:v0.0.48, but successfully downloaded docker.io/kicbase/stable:v0.0.48 as a fallback image
* 创建 docker container (CPU=2, 内存=8000MB) ...
! 从 Minikube 的 container 内部连接到 https://registry.cn-hangzhou.aliyuncs.com/google_containers/ 失败
* 要获取新的外部镜像, 可能需要配置代理: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* 正在 Docker 28.4.0 中准备 Kubernetes v1.34.0...
* 配置 bridge CNI (Container Networking Interface) ...
* 正在验证 Kubernetes 组件...
- 正在使用镜像 registry.cn-hangzhou.aliyuncs.com/google_containers/storage-provisioner:v5
* 启用插件: storage-provisioner, default-storageclass

! C:\Program Files\Docker\ Docker\resources\bin\kubectl.exe 的版本为 1.32.2, 可能与 Kubernetes 1.34.0 不兼容。
- 想要使用 kubectl v1.34.0 吗? 尝试使用 'minikube kubectl -- get pods -A' 命令
* 完成! kubectl 现在已配置, 默认使用"minikube"集群和"default"命名空间
```



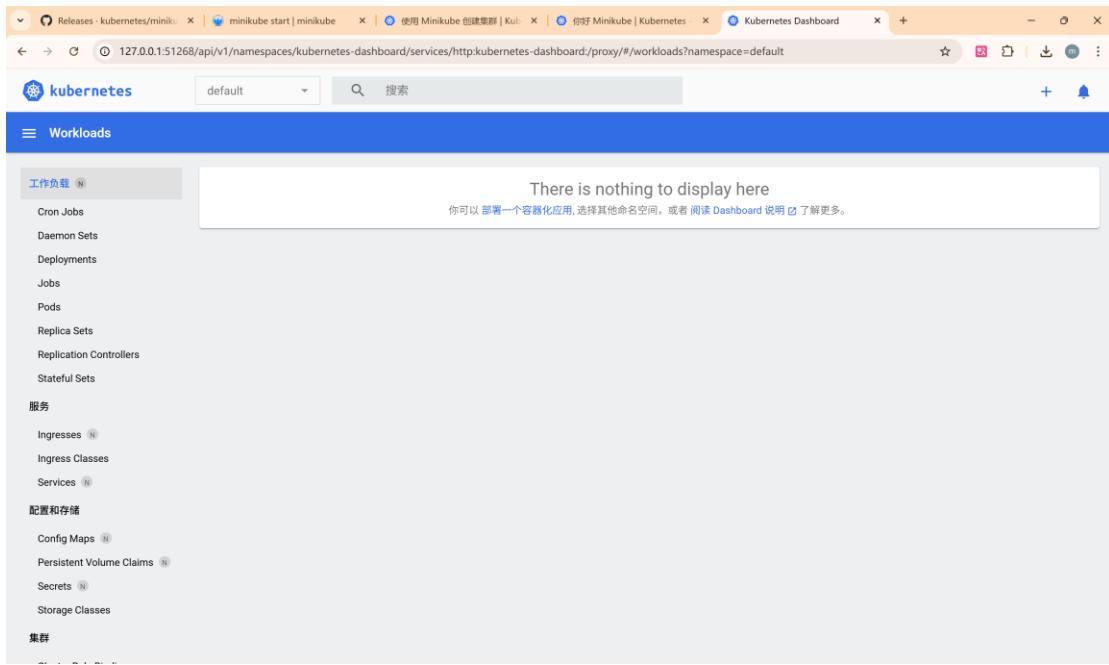
4. Open the dashboard:

```
minikube dashboard
```

```
PS C:\Windows\system32> minikube dashboard
* 正在开启 dashboard ...
- 正在使用镜像 docker.io/kubernetesui/dashboard:v2.7.0
- 正在使用镜像 docker.io/kubernetesui/metrics-scraper:v1.0.8
* 某些仪表板功能需要 metrics-server 插件。要启用所有功能, 请运行:

        minikube addons enable metrics-server

* 正在验证 dashboard 运行情况 ...
* 正在启动代理...
* 正在验证 proxy 运行状况 ...
* 正在使用默认浏览器打开 http://127.0.0.1:51268/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard-proxy/ ...
```



Create a deployment

1. Create a deployment that manages a pod:

```
kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.53 -- /agnhost netexec --http-port=8080
```

```
PS C:\Windows\system32> kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.53 -- /agnhost netexec --http-port=8080
deployment.apps/hello-node created
```

2. View it in the dashboard:

名称	镜像	标签	Pods	创建时间
hello-node	registry.k8s.io/e2e-test-images/agnhost:2.53	app: hello-node	1 / 1	11 minutes ago

名称	镜像	标签	节点	状态	重启	CPU 使用率 (cores)	内存使用 (bytes)	创建时间
hello-node	registry.k8s.io/e2e-test-images/agnhost:2.53	app: hello-node		Running	0	0.00	0	11 minutes

3. Use get command to view information:

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

```
PS C:\Windows\system32> kubectl get deployments
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
hello-node 1/1     1           1           72s
PS C:\Windows\system32> kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
hello-node-6c9b5f4b59-rpwg 1/1     Running   0          4m
PS C:\Windows\system32> kubectl get events
LAST SEEN   TYPE    REASON          OBJECT        MESSAGE
4m8s       Normal   Scheduled       pod/hello-node-6c9b5f4b59-rpwg  Successfully assigned default/hello-node-6c9b5f4b59-rpwg to minikube
4m7s       Normal   Pulling        pod/hello-node-6c9b5f4b59-rpwg  Pulling image "registry.k8s.io/e2e-test-images/agnhost:2.53"
3m56s      Normal   Pulled        pod/hello-node-6c9b5f4b59-rpwg  Successfully pulled image "registry.k8s.io/e2e-test-images/agnhost:2.53" in 10.215s (10.216s including waiting). Image size: 139374622 bytes.
3m56s      Normal   Created        pod/hello-node-6c9b5f4b59-rpwg  Created container: agnhost
3m56s      Normal   Started        pod/hello-node-6c9b5f4b59-rpwg  Started container agnhost
4m8s       Normal   SuccessfulCreate replicaset/hello-node-6c9b5f4b59  Created pod: hello-node-6c9b5f4b59-rpwg
4m8s       Normal   ScalingReplicaSet deployment/hello-node-6c9b5f4b59 from 0 to 1  Scaled up replica set hello-node-6c9b5f4b59
37m        Normal   NodeHasSufficientMemory node/minikube  Node minikube status is now: NodeHasSufficientMemory
37m        Normal   NodeHasNoDiskPressure   node/minikube  Node minikube status is now: NodeHasNoDiskPressure
37m        Normal   NodeHasSufficientPID   node/minikube  Node minikube status is now: NodeHasSufficientPID
37m        Normal   NodeAllocatableEnforced node/minikube  Updated Node Allocatable limit across pods
37m        Normal   Starting        node/minikube  Starting kubelet.
```

4. View config and logs:

```
kubectl config view
kubectl logs hello-node-5f76cf6ccf-br9b5
```

```
PS C:\Windows\system32> kubectl logs hello-node-6c9b5f4b59-rpwg
I0928 08:56:23.484885      1 log.go:245] Started HTTP server on port 8080
I0928 08:56:23.508990      1 log.go:245] Started UDP server on port 8081
```

Create a Service

1. Expose the Pod to the public internet using the kubectl expose command:

```
kubectl expose deployment hello-node --type=LoadBalancer --port=8080
```

```
PS C:\Windows\system32> kubectl expose deployment hello-node --type=LoadBalancer --port=8080
service/hello-node exposed
```

2. View the Service you created:

```
kubectl get services
```

```
PS C:\Windows\system32> kubectl get services
NAME      TYPE      CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
hello-node LoadBalancer  10.97.241.151 <pending>   8080:32635/TCP  2m35s
kubernetes ClusterIP  10.96.0.1    <none>      443/TCP   42m
```

3. Opens up a browser window that serves your app and shows the app's response:

```
minikube service hello-node
```

```
PS C:\Windows\system32> minikube service hello-node
NAMESPACE | NAME | TARGET PORT | URL
default   | hello-node | 8080 | http://192.168.49.2:32635 |

* 为服务 hello-node 启动隧道。/
NAMESPACE | NAME | TARGET PORT | URL
default   | hello-node |          | http://127.0.0.1:65423 |

* 为服务 hello-node 启动隧道。-
* 正通过默认浏览器打开服务 default/hello-node...
! 因为你正在使用 windows 上的 Docker 驱动程序，所以需要打开终端才能运行它。
```



Enable addons

1. List the currently supported addons:

```
minikube addons list
```

ADDON NAME	PROFILE	STATUS	MANTAINER
ambassador	minikube	disabled	3rd party (Ambassador)
amd-gpu-device-plugin	minikube	disabled	3rd party (AMD)
auto-pause	minikube	disabled	minikube
cloud-spanner	minikube	disabled	Google
csi-hostpath-driver	minikube	disabled	Kubernetes
dashboard	minikube	enabled <input checked="" type="checkbox"/>	Kubernetes
default-storageclass	minikube	enabled <input checked="" type="checkbox"/>	Kubernetes
efk	minikube	disabled	3rd party (Elastic)
freshpod	minikube	disabled	Google
gcp-auth	minikube	disabled	Google
gvvisor	minikube	disabled	minikube
headlamp	minikube	disabled	3rd party (kinvolk.io)
inaccel	minikube	disabled	3rd party (InAccel [info@inaccel.com])
ingress	minikube	disabled	Kubernetes
ingress-dns	minikube	disabled	minikube
inspektor-gadget	minikube	disabled	3rd party (inspektor-gadget.io)
istio	minikube	disabled	3rd party (Istio)
istio-provisioner	minikube	disabled	3rd party (Istio)
kong	minikube	disabled	3rd party (Kong HQ)
kubeflow	minikube	disabled	3rd party
kubetaint	minikube	disabled	3rd party (kubetaint.com)
kubevirt	minikube	disabled	3rd party (KubeVirt)
logviewer	minikube	disabled	3rd party (unknown)
metallb	minikube	disabled	3rd party (MetallLB)
metrics-server	minikube	disabled	Kubernetes
nvidia-device-plugin	minikube	disabled	3rd party (NVIDIA)
nvidia-driver-installer	minikube	disabled	3rd party (NVIDIA)
nvidia-gpu-device-plugin	minikube	disabled	3rd party (NVIDIA)
olm	minikube	disabled	3rd party (Operator Framework)
pod-security-policy	minikube	disabled	3rd party (unknown)
portainer	minikube	disabled	3rd party (Portainer.io)
registry	minikube	disabled	minikube
registry-aliases	minikube	disabled	3rd party (unknown)
registry-creds	minikube	disabled	3rd party (UPMC Enterprises)
storage-provisioner	minikube	enabled <input checked="" type="checkbox"/>	minikube
storage-provisioner-gluster	minikube	disabled	3rd party (Gluster)
storage-provisioner-rancher	minikube	disabled	3rd party (Rancher)
volcano	minikube	disabled	third-party (volcano)
volumesnapshots	minikube	disabled	Kubernetes
yakd	minikube	disabled	3rd party (marcnuri.com)

2. Enable an addon:

```
minikube addons enable kong
```

```
PS C:\Windows\system32> minikube addons enable kong
! kong 是第三方插件，不由 minikube 维护者进行维护或验证，启用需自担风险。
* kong 由 3rd party (Kong HQ) 维护，如有任何问题，请在 GitHub 上联系 @gAmUss
  - 正在使用镜像 docker.io/kong:3.9.1
  - 正在使用镜像 docker.io/kong/kubernetes-ingress-controller:3.5.1
* 启动 'kong' 插件
```

3. View the Pod and Service you created by installing that addon:

```
kubectl get pod,svc -n kube-system
```

PS C:\Windows\system32> kubectl get pod,svc -n kube-system					
NAME	READY	STATUS	RESTARTS	AGE	
pod/coredns-7ddb67b59b-c52ms	1/1	Running	0	61m	
pod/etc-d-minikube	1/1	Running	0	62m	
pod/kube-apiserver-minikube	1/1	Running	0	62m	
pod/kube-controller-manager-minikube	1/1	Running	0	62m	
pod/kube-proxy-jn8nl	1/1	Running	0	61m	
pod/kube-scheduler-minikube	1/1	Running	0	62m	
pod/storage-provisioner	1/1	Running	0	62m	
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/kube-dns	ClusterIP	10.96.0.10	<none>	53/UDP, 53/TCP, 9153/TCP	62m
service/metrics-server	ClusterIP	10.109.20.187	<none>	443/TCP	7m14s

4. Disable kong:

```
minikube addons disable kong
```

```
PS C:\Windows\system32> minikube addons disable kong
* 'kong' 插件已被禁用
```

Using kubectl to Create a Deployment

1. Deploy an app:

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

```
PS C:\Windows\System32> kubectl create deployment kubernetes-bootcamp --image=gcr.io/google-samples/kubernetes-bootcamp:v1
deployment.apps/kubernetes-bootcamp created
```

2. To list your deployments use the kubectl get deployments command:

```
PS C:\Windows\System32> kubectl get deployments
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
bookinfo-gateway-istio   1/1     1           1          34h
details-v1       1/1     1           1          4d16h
kubernetes-bootcamp   1/1     1           1          2m1s
productpage-v1      1/1     1           1          4d16h
ratings-v1         1/1     1           1          4d16h
reviews-v1          1/1     1           1          4d16h
reviews-v2          1/1     1           1          4d16h
reviews-v3          1/1     1           1          4d16h
```

3. The kubectl proxy command can create a proxy that will forward communications into the cluster-wide, private network. The proxy can be terminated by pressing control-C and won't show any output while it's running.

```
PS C:\Windows\System32> kubectl proxy
Starting to serve on 127.0.0.1:8001
```

4. You can see all those APIs hosted through the proxy endpoint. For example, we can query the version directly through the API using the curl command:

```
curl http://localhost:8001/version
```

```
PS C:\Windows\System32> curl http://localhost:8001/version

StatusCode : 200
StatusDescription : OK
Content : {
    "major": "1",
    "minor": "34",
    "emulationMajor": "1",
    "emulationMinor": "34",
    "minCompatibilityMajor": "1",
    "minCompatibilityMinor": "33",
    "gitVersion": "v1.34.0",
    "gitCommit": "f28b4c..."
}
RawContent : HTTP/1.1 200 OK
Audit-Id: 5b0642e0-befc-4860-8f7c-bf27d1de446c
X-Kubernetes-Pf-Flowschema-Uid: 53a1139d-7ac1-477a-a8c1-1573c380966d
X-Kubernetes-Pf-Prioritylevel-Uid: 933ce1d9-bec3-4054-8118-481ebb...
Forms : {}
Headers : {[Audit-Id, 5b0642e0-befc-4860-8f7c-bf27d1de446c], [X-Kubernetes-Pf-Flowschema-Uid, 53a1139d-7ac1-477a-a8c1-1573c380966d], [X-Kubernetes-Pf-Prioritylevel-Uid, 933ce1d9-bec3-4054-8118-481ebb9fb70c], [Content-Length, 379]...}
Images : {}
InputFields : {}
Links : {}
ParsedHtml : mshtml.HTMLDocumentClass
RawContentLength : 379
```

- The API server will automatically create an endpoint for each pod, based on the pod name, that is also accessible through the proxy.

First we need to get the Pod name, and we'll store it in the environment variable POD_NAME:

```
$names = (kubectl get pods -o jsonpath='{.items[*].metadata.name}') -split '\s+'
$POD_NAME = ($names -join "`n")
```

```
Write-Host "Name of the Pod:`n$POD_NAME"
```

```
PS C:\Windows\System32> $names = (kubectl get pods -o jsonpath='{.items[*].metadata.name}') -split '\s+
PS C:\Windows\System32> $POD_NAME = ($names -join "`n")
PS C:\Windows\System32> Write-Host "Name of the Pod:`n$POD_NAME"
Name of the Pod:
bookinfo-gateway-istio-6bbf6978f9-9kcv5
details-v1-77d6bd5675-n4zrp
kubernetes-bootcamp-658f6cbd58-6x8wj
productpage-v1-bb87ff47b-r75r5
ratings-v1-8589f64b4c-sntz4
reviews-v1-8cf7b9cc5-scgpl
reviews-v2-67d565655f-4c6kt
reviews-v3-d587fc9d7-w92jc
```

- You can access the Pod through the proxied API, by running:

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/)

Viewing Pods and Nodes

Check application configuration

- Let's verify that the application we deployed in the previous scenario is running.

We'll use the kubectl get command and look for existing Pods:

kubectl get pods

```
PS C:\Windows\System32> kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
bookinfo-gateway-istio-6bbf6978f9-9kcv5   1/1     Running   3 (11m ago)   34h
details-v1-77d6bd5675-n4zrp                 2/2     Running   4 (11m ago)   4d16h
kubernetes-bootcamp-658f6cbd58-6x8wj       2/2     Running   0          10m
productpage-v1-bb87ff47b-r75r5                2/2     Running   4 (11m ago)   4d16h
ratings-v1-8589f64b4c-sntz4                  2/2     Running   4 (11m ago)   4d16h
reviews-v1-8cf7b9cc5-scgpl                  2/2     Running   4 (11m ago)   4d16h
reviews-v2-67d565655f-4c6kt                  2/2     Running   4 (11m ago)   4d16h
reviews-v3-d587fc9d7-w92jc                  2/2     Running   4 (11m ago)   4d16h
```

- Next, to view what containers are inside that Pod and what images are used to build those containers we run the kubectl describe pods command:

kubectl describe pods

```
Windows PowerShell -> istio-ca-crl
Optional: false
istio-ca-crl:
  Type: ConfigMap (a volume populated by a ConfigMap)
  Name: istio-ca-crl
  Optional: true
wlp-output:
  Type: EmptyDir (a temporary directory that shares a pod's lifetime)
  Medium:
  SizeLimit: <unset>
tmp:
  Type: EmptyDir (a temporary directory that shares a pod's lifetime)
  Medium:
  SizeLimit: <unset>
kube-api-access-vfnsn:
  Type: Projected (a volume that contains injected data from multiple sources)
  TokenExpirationSeconds: 3607
  ConfigMapName: kube-root-ca.crt
  ConfigMapOptional: <nil>
  DownwardAPI: true
  QoS Class: Burstable
  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 SandboxChanged 11m kubelet Pod sandbox changed, it will be killed and re-created.
  Normal Pulled 11m kubelet Container image "docker.io/istio/proxyv2:1.27.1" already present
on machine
  Normal Created 11m kubelet Created container: istio-init
  Normal Started 11m kubelet Started container istio-init
  Normal Pulled 11m kubelet Container image "docker.io/istio/proxyv2:1.27.1" already present
on machine
  Normal Created 11m kubelet Created container: istio-proxy
  Normal Started 11m kubelet Started container istio-proxy
  Warning Unhealthy 10m (x16 over 11m) kubelet Startup probe failed: Get "http://10.244.0.142:15021/healthz/readiness": dial tcp 10.244.0.142:15021: connect: connection refused
  Normal Pulled 10m kubelet Container image "docker.io/istio/examples-bookinfo-reviews-v3:1.20.3" already present on machine
  Normal Created 10m kubelet Created container: reviews
  Normal Started 10m kubelet Started container reviews
PS C:\Windows\System32> |
```

Show the app in the terminal

- Recall that Pods are running in an isolated, private network - so we need to proxy access to them so we can debug and interact with them. To do this, we'll use the kubectl proxy command to run a proxy in a **second terminal**. Open a new terminal window, and in that new terminal, run:

```
kubectl proxy
```

```
PS C:\Windows\System32> kubectl proxy
Starting to serve on 127.0.0.1:8001
```

- To see the output of our application, run a curl request:
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/)

```
PS C:\Windows\System32> $POD_NAME
bookinfo-gateway-istio-6bbf6978f9-9kcv5
details-v1-77d6bd5675-n4zrp
kubernetes-bootcamp-658f6cbd58-6x8wj
productpage-v1-bb87ff47b-r75r5
ratings-v1-8589f64b4c-sntz4
reviews-v1-8cf7b9cc5-scopl
reviews-v2-67d565655f-4c6kt
reviews-v3-d587fc9d7-w92jc
PS C:\Windows\System32> curl http://localhost:8001/api/v1/namespaces/default/pods/kubernetes-bootcamp-658f6cbd58-6x8wj:8080/proxy/
{
  StatusCode : 200
  StatusDescription : OK
  Content : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-658f6cbd58-6x8wj | v=1
  RawContent : HTTP/1.1 200 OK
                Audit-Id: 48a29e62-5060-4876-881a-8c20212ce3d1
                X-Envoy-Decorator-Operation: :0/*
                X-Envoy-Upstream-Service-Time: 15
                Transfer-Encoding: chunked
                Cache-Control: no-cache, private
                Con...
  Forms : {}
  Headers : {[Audit-Id, 48a29e62-5060-4876-881a-8c20212ce3d1], [X-Envoy-Decorator-Operation, :0/*], [X-Envoy-Upstream-Service-Time, 15], [Transfer-Encoding, chunked]...}
  Images : {}
  InputFields : {}
  Links : {}
  ParsedHtml : mshtml.HTMLDocumentClass
  RawContentLength : 84
}
```

Executing commands on the container

1. We can execute commands directly on the container once the Pod is up and running. For this, we use the exec subcommand and use the name of the Pod as a parameter. Let's list the environment variables:

```
kubectl exec "$POD_NAME" – env
```

```
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-658f6cbd58-6x8wj
BOOKINFO_GATEWAY_ISTIO_PORT_80_TCP=tcp://10.96.103.189:80
BOOKINFO_GATEWAY_ISTIO_PORT_80_TCP_PROTO=tcp
HELLO_NODE_SERVICE_PORT=8080
DETAILS_PORT_9080_TCP_ADDR=10.111.234.8
HELLO_NODE_PORT_8080_TCP=tcp://10.97.241.151:8080
REVIEWS_SERVICE_PORT_HTTP=9080
PRODUCTPAGE_SERVICE_PORT_HTTP=9080
KUBERNETES_PORT=tcp://10.96.0.1:443
REVIEWS_PORT_9080_TCP=tcp://10.105.253.186:9080
BOOKINFO_GATEWAY_ISTIO_SERVICE_HOST=10.96.103.189
DETAILS_SERVICE_PORT=9080
KUBERNETES_SERVICE_HOST=10.96.0.1
BOOKINFO_GATEWAY_ISTIO_SERVICE_PORT_STATUS_PORT=15021
DETAILS_SERVICE_PORT_HTTP=9080
KUBERNETES_PORT_443_TCP=tcp://10.96.0.1:443
RATINGS_SERVICE_HOST=10.106.84.3
HELLO_NODE_PORT_8080_TCP_ADDR=10.97.241.151
RATINGS_PORT=tcp://10.106.84.3:9080
DETAILS_SERVICE_HOST=10.111.234.8
HELLO_NODE_PORT_8080_TCP_PORT=8080
BOOKINFO_GATEWAY_ISTIO_PORT_15021_TCP_PORT=15021
BOOKINFO_GATEWAY_ISTIO_PORT_80_TCP_PORT=80
KUBERNETES_SERVICE_PORT_HTTPS=443
KUBERNETES_PORT_443_TCP_PROTO=tcp
BOOKINFO_GATEWAY_ISTIO_PORT=tcp://10.96.103.189:15021
BOOKINFO_GATEWAY_ISTIO_PORT_15021_TCP_ADDR=10.96.103.189
PRODUCTPAGE_PORT_9080_TCP=tcp://10.98.235.106:9080
PRODUCTPAGE_PORT_9080_TCP_ADDR=10.98.235.106
RATINGS_PORT_9080_TCP=tcp://10.106.84.3:9080
REVIEWS_PORT=tcp://10.105.253.186:9080
RATINGS_SERVICE_PORT_HTTP=9080
PRODUCTPAGE_PORT=tcp://10.98.235.106:9080
HELLO_NODE_PORT_8080_TCP_PROTO=tcp
RATINGS_PORT_9080_TCP_PROTO=tcp
RATINGS_PORT_9080_TCP_PORT=9080
DETAILS_PORT=tcp://10.111.234.8:9080
HELLO_NODE_PORT=tcp://10.97.241.151:8080
KUBERNETES_PORT_443_TCP_PORT=443
BOOKINFO_GATEWAY_ISTIO_SERVICE_PORT_HTTP=80
PRODUCTPAGE_SERVICE_HOST=10.98.235.106
```

2. Next let's start a bash session in the Pod's container:
kubectl exec -ti \$POD_NAME -- bash
3. The source code of the app is in the server.js file:
cat server.js

```

PS C:\Windows\System32> kubectl exec -ti $POD_NAME -- bash
root@kubernetes-bootcamp-658f6cbd58-6x8wj:/# 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 , " onds, " | 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");
});

```

4. You can check that the application is up by running a curl command:

`curl http://localhost:8080`

```

root@kubernetes-bootcamp-658f6cbd58-6x8wj:/# curl http://localhost:8080
Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-658f6cbd58-6x8wj | v=1

```

Using a Service to Expose Your App

Step 1: Creating a new Service

1. We'll use the kubectl get command and look for existing Pods:

`kubectl get pods`

```

PS C:\Windows\System32> kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
bookinfo-gateway-istio-6bbf6978f9-9kcv5   1/1     Running   3 (31m ago)  35h
details-v1-77d6bd5675-n4zrp                2/2     Running   4 (31m ago)  4d16h
kubernetes-bootcamp-658f6cbd58-6x8wj      2/2     Running   0          30m
productpage-v1-bb87ff47b-r75r5              2/2     Running   4 (31m ago)  4d16h
ratings-v1-8589f64b4c-sntz4                2/2     Running   4 (31m ago)  4d16h
reviews-v1-8cf7b9cc5-scgpl                 2/2     Running   4 (31m ago)  4d16h
reviews-v2-67d565655f-4c6kt                 2/2     Running   4 (31m ago)  4d16h
reviews-v3-d587fc9d7-w92jc                 2/2     Running   4 (31m ago)  4d16h

```

2. Next, let's 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
bookinfo-gateway-istio  ClusterIP  10.96.103.189 <none>       15021/TCP,80/TCP  35h
details      ClusterIP  10.111.234.8  <none>       9080/TCP   4d16h
hello-node   LoadBalancer 10.97.241.151  <pending>   8080:32635/TCP  17d
kubernetes   ClusterIP  10.96.0.1    <none>       443/TCP    17d
productpage  ClusterIP  10.98.235.106 <none>       9080/TCP   4d16h
ratings      ClusterIP  10.106.84.3   <none>       9080/TCP   4d16h
reviews      ClusterIP  10.105.253.186 <none>       9080/TCP   4d16h

```

3. To expose the deployment to external traffic, we'll use the kubectl expose command with the `--type=NodePort` option:

`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

```

4. To find out what port was opened externally (for the type: NodePort Service) we'll run the describe service subcommand:

```
kubectl describe services/kubernetes-bootcamp
```

```
PS C:\Windows\System32> kubectl describe services/kubernetes-bootcamp
Name:           kubernetes-bootcamp
Namespace:      default
Labels:         app=kubernetes-bootcamp
Annotations:   <none>
Selector:       app=kubernetes-bootcamp
Type:          NodePort
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.108.67.253
IPs:           10.108.67.253
Port:          <unset>  8080/TCP
TargetPort:    8080/TCP
NodePort:      <unset>  32331/TCP
Endpoints:     10.244.0.149:8080
Session Affinity: None
External Traffic Policy: Cluster
Internal Traffic Policy: Cluster
Events:        <none>
```

5. If you're running minikube with Docker Desktop as the container driver, a minikube tunnel is needed. This is because containers inside Docker Desktop are isolated from your host computer.
6. In a separate terminal window, execute:

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

```
PS C:\Windows\System32> minikube service kubernetes-bootcamp --url
http://127.0.0.1:55910
! 因为你正在使用 windows 上的 Docker 驱动程序，所以需要打开终端才能运行它。
```

7. Then use the given URL to access the app:

```
curl http://127.0.0.1:55910
```

```
PS C:\Windows\System32> curl http://127.0.0.1:55910

StatusCode : 200
StatusDescription : OK
Content : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-658f6cbd58-6x8wj | v=1

RawContent : HTTP/1.1 200 OK
             x-envoy-upstream-service-time: 6
             x-envoy-decorator-operation: kubernetes-bootcamp.default.svc.cluster.local:8080/*
             transfer-encoding: chunked
             Content-Type: text/plain
             Date: Wed, 1...

Forms : {}
Headers : {[x-envoy-upstream-service-time, 6], [x-envoy-decorator-operation, kubernetes-bootcamp.default.svc.cluster.local:8080/*], [transfer-encoding, chunked], [Content-Type, text/plain]...}
Images : {}
InputFields : {}
Links : {}
ParsedHtml : mshtml.HTMLDocumentClass
RawContentLength : 84
```

Step 2: Using labels

1. The Deployment created automatically a label for our Pod. With the describe deployment subcommand you can see the name (the *key*) of that label:
`kubectl describe deployment`

```

Name:           kubernetes-bootcamp
Namespace:      default
CreationTimestamp: Thu, 16 Oct 2025 00:45:58 +0800
Labels:          app=kubernetes-bootcamp
Annotations:    deployment.kubernetes.io/revision: 1
Selector:        app=kubernetes-bootcamp
Replicas:       1 desired | 1 updated | 1 total | 1 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>
      Node-Selectors: <none>
      Tolerations:  <none>
  Conditions:
    Type     Status  Reason
    ----  -----
    Available  True    MinimumReplicasAvailable
    Progressing  True    NewReplicaSetAvailable
  OldReplicaSets: <none>
  NewReplicaSet:  kubernetes-bootcamp-658f6cbd58 (1/1 replicas created)
Events:
  Type     Reason            Age   From           Message
  ----  -----  ----  ----
  Normal  ScalingReplicaSet  39m  deployment-controller  Scaled up replica set kubernetes-bootcamp-658f6cbd58 from 0 to 1

```

- Let's use this label to query our list of Pods. We'll use the `kubectl get pods` command with `-l` as a parameter, followed by the label values:

```
kubectl get pods -l app=kubernetes-bootcamp
```

```
PS C:\Windows\System32> kubectl get pods -l app=kubernetes-bootcamp
NAME                  READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-658f6cbd58-6x8wj   2/2     Running   0          40m
```

- You can do the same to list the existing Services:

```
kubectl get services -l app=kubernetes-bootcamp
```

```
PS C:\Windows\System32> kubectl get services -l app=kubernetes-bootcamp
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes-bootcamp   NodePort    10.108.67.253  <none>        8080:32331/TCP  9m29s
```

- Get the name of the Pod and store it in the `POD_NAME` environment variable.
- To apply a new label we use the `label` subcommand followed by the object type, object name and the new label:

```
kubectl label pods "$POD_NAME" version=v1
```

```
PS C:\Windows\System32> kubectl label pods "$POD_NAME" version=v1
pod/kubernetes-bootcamp-658f6cbd58-6x8wj labeled
```

- This will apply a new label to our Pod (we pinned the application version to the Pod), and we can check it with the `describe pod` command:

```
kubectl describe pods "$POD_NAME"
```

```

PS C:\Windows\System32> kubectl describe pods "$POD_NAME"
Name:           kubernetes-bootcamp-658f6cbd58-6x8wj
Namespace:      default
Priority:      0
Service Account: default
Node:          minikube/192.168.49.2
Start Time:    Thu, 16 Oct 2025 00:46:22 +0800
Labels:         app=kubernetes-bootcamp
                pod-template-hash=658f6cbd58
                security.istio.io/tlsMode=istio
                service.istio.io/canonical-name=kubernetes-bootcamp
                service.istio.io/canonical-revision=latest
                version=v1
Annotations:   istio.io/rev: default
                kubectl.kubernetes.io/default-container: kubernetes-bootcamp
                kubectl.kubernetes.io/default-logs-container: kubernetes-bootcamp
                prometheus.io/path: /stats/prometheus
                prometheus.io/port: 15020
                prometheus.io/scrape: true
                sidecar.istio.io/status:
                    {"initContainers":["istio-init","istio-proxy"],"containers":null,"volumes":["workload-socket","crede
ntial-socket","workload-certs","istio-...
Status:        Running
IP:            10.244.0.149
IPs:
    IP:       10.244.0.149
Controlled By: ReplicaSet/kubernetes-bootcamp-658f6cbd58
Init Containers:
    istio-init:
        Container ID: docker://1edebbb4b6b7681188b4fe8c7be7a755c96d3541edeca619e19e01f0a642d38eb
        Image:        docker.io/istio/proxyv2:1.27.1
        Image ID:     docker-pullable://istio/proxyv2@sha256:d902f7f80be93a0521af5e5796d1e989f2107af0725eca8881eb4c9848bd7
bf
        Port:        <none>
        Host Port:   <none>
        Args:
            istio-iptables
            -p
            15001
            -z
            15006

```

- We see here that the label is attached now to our Pod. And we can query now the list of pods using the new label:

```
kubectl get pods -l version=v1
```

NAME	READY	STATUS	RESTARTS	AGE
details-v1-77d6bd5675-n4zrp	2/2	Running	4 (44m ago)	4d17h
kubernetes-bootcamp-658f6cbd58-6x8wj	2/2	Running	0	43m
productpage-v1-bb87ff47b-r75r5	2/2	Running	4 (44m ago)	4d17h
ratings-v1-8589f64b4c-sntz4	2/2	Running	4 (44m ago)	4d17h
reviews-v1-8cf7b9cc5-scgpl	2/2	Running	4 (44m ago)	4d17h

Step 3: Deleting a service

- To delete Services you can use the delete service subcommand. Labels can be used also here:

```
kubectl delete service -l app=kubernetes-bootcamp
```

```

PS C:\Windows\System32> kubectl delete service -l app=kubernetes-bootcamp
service "kubernetes-bootcamp" deleted

```

- Confirm that the Service is gone:

```
kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
bookinfo-gateway-istio	ClusterIP	10.96.103.189	<none>	15021/TCP, 80/TCP	35h
details	ClusterIP	10.111.234.8	<none>	9080/TCP	4d17h
hello-node	LoadBalancer	10.97.241.151	<pending>	8080:32635/TCP	17d
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	17d
productpage	ClusterIP	10.98.235.106	<none>	9080/TCP	4d17h
ratings	ClusterIP	10.106.84.3	<none>	9080/TCP	4d17h
reviews	ClusterIP	10.105.253.186	<none>	9080/TCP	4d17h

- This confirms that our Service was removed. To confirm that route is not exposed anymore you can curl the previously exposed IP and port:

```
curl http://"$(minikube ip)":$NODE_PORT"
```

```
PS C:\Windows\System32> curl http://"$(minikube ip)":$NODE_PORT"
curl : 基础连接已经关闭：连接被意外关闭。
所在位置 行:1 字符: 1
+ curl http://"$(minikube ip)":$NODE_PORT"
+ ~~~~~
+ CategoryInfo          : InvalidOperationException: (System.Net.HttpWebRequest:HttpWebRequest) [Invoke-WebRequest], WebException
+ FullyQualifiedErrorMessage : WebCmdletWebResponseException, Microsoft.PowerShell.Commands.InvokeWebRequestCommand
```

4. This proves that the application is not reachable anymore from outside of the cluster. You can confirm that the app is still running with a curl from inside the pod:
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-658f6cbd58-6x8wj | v=1
```

We see here that the application is up. This is because the Deployment is managing the application. To shut down the application, you would need to delete the Deployment as well.

Running Multiple Instances of Your App

Scaling a Deployment

1. To list your Deployments, use the get deployments subcommand:

```
kubectl get deployments
```

```
PS C:\Windows\System32> kubectl get deployments
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
bookinfo-gateway-istio   1/1     1           1           35h
details-v1      1/1     1           1           4d17h
kubernetes-bootcamp   1/1     1           1           48m
productpage-v1    1/1     1           1           4d17h
ratings-v1       1/1     1           1           4d17h
reviews-v1       1/1     1           1           4d17h
reviews-v2       1/1     1           1           4d17h
reviews-v3       1/1     1           1           4d17h
PS C:\Windows\System32>
```

2. To see the ReplicaSet created by the Deployment, run:

```
kubectl get rs
```

```
PS C:\Windows\System32> kubectl get rs
NAME           DESIRED   CURRENT   READY   AGE
bookinfo-gateway-istio-6bbf6978f9   1         1         1         35h
bookinfo-gateway-istio-79d769c47   0         0         0         35h
details-v1-77d6bd5675   1         1         1         4d17h
kubernetes-bootcamp-658f6cbd58   1         1         1         48m
productpage-v1-bb87ff47b   1         1         1         4d17h
ratings-v1-8589f64b4c   1         1         1         4d17h
reviews-v1-8cf7b9cc5   1         1         1         4d17h
reviews-v2-67d565655f   1         1         1         4d17h
reviews-v3-d587fc9d7   1         1         1         4d17h
PS C:\Windows\System32>
```

3. Next, let's scale the Deployment to 4 replicas. We'll use the kubectl scale command, followed by the Deployment type, name and desired number of instances:

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

```
PS C:\Windows\System32> kubectl scale deployments/kubernetes-bootcamp --replicas=4
deployment.apps/kubernetes-bootcamp scaled
```

4. To list your Deployments once again, use get deployments:

```
kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
bookinfo-gateway-istio	1/1	1	1	35h
details-v1	1/1	1	1	4d17h
kubernetes-bootcamp	4/4	4	4	50m
productpage-v1	1/1	1	1	4d17h
ratings-v1	1/1	1	1	4d17h
reviews-v1	1/1	1	1	4d17h
reviews-v2	1/1	1	1	4d17h
reviews-v3	1/1	1	1	4d17h

5. Next, let's check if the number of Pods changed:

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

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED
DE READINESS GATES							
bookinfo-gateway-istio-6bbf6978f9-9kcv5	1/1	Running	3 (51m ago)	35h	10.244.0.130	minikube	<none>
<none>							
details-v1-77d6bd5675-n4zrp	2/2	Running	4 (51m ago)	4d17h	10.244.0.145	minikube	<none>
<none>							
kubernetes-bootcamp-658f6cbd58-5qk2c	2/2	Running	0	58s	10.244.0.156	minikube	<none>
<none>							
kubernetes-bootcamp-658f6cbd58-6x8wj	2/2	Running	0	50m	10.244.0.149	minikube	<none>
<none>							
kubernetes-bootcamp-658f6cbd58-hpn12	2/2	Running	0	58s	10.244.0.157	minikube	<none>
<none>							
kubernetes-bootcamp-658f6cbd58-t4kg2	2/2	Running	0	58s	10.244.0.155	minikube	<none>
<none>							
productpage-v1-bb87ff47b-r75r5	2/2	Running	4 (51m ago)	4d17h	10.244.0.144	minikube	<none>
<none>							
ratings-v1-8589f64b4c-sntz4	2/2	Running	4 (51m ago)	4d17h	10.244.0.143	minikube	<none>
<none>							
reviews-v1-8cf7b9cc5-scgpl	2/2	Running	4 (51m ago)	4d17h	10.244.0.140	minikube	<none>
<none>							
reviews-v2-67d565655f-4c6kt	2/2	Running	4 (51m ago)	4d17h	10.244.0.141	minikube	<none>
<none>							
reviews-v3-d587fc9d7-w92jc	2/2	Running	4 (51m ago)	4d17h	10.244.0.142	minikube	<none>
<none>							

6. The change was registered in the Deployment events log. To check that, use the describe subcommand:

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

```

<none>
PS C:\Windows\System32> kubectl describe deployments/kubernetes-bootcamp
Name:           kubernetes-bootcamp
Namespace:      default
CreationTimestamp: Thu, 16 Oct 2025 00:45:58 +0800
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>
      Node-Selectors: <none>
      Tolerations: <none>
  Conditions:
    Type     Status  Reason
    ----   ----   -----
    Progressing  True   NewReplicaSetAvailable
    Available   True   MinimumReplicasAvailable
OldReplicaSets: <none>
NewReplicaSet:  kubernetes-bootcamp-658f6cbd58 (4/4 replicas created)
Events:
  Type     Reason          Age   From            Message
  ----   -----          ---   ---            -----
  Normal  ScalingReplicaSet 51m   deployment-controller  Scaled up replica set kubernetes-bootcamp-658f6cbd58 from 0 to 1
  Normal  ScalingReplicaSet 99s   deployment-controller  Scaled up replica set kubernetes-bootcamp-658f6cbd58 from 1 to 4

```

Load Balancing

1. Next, we'll do a curl to the exposed IP address and port. Execute the command multiple times:

```

PS C:\Windows\System32> curl http://127.0.0.1:58168

StatusCode : 200
StatusDescription : OK
Content : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-658f6cbd58-hpn12 | v=1
RawContent : HTTP/1.1 200 OK
             x-envoy-upstream-service-time: 22
             x-envoy-decorator-operation: kubernetes-bootcamp.default.svc.cluster.local:8080/*
             transfer-encoding: chunked
             Content-Type: text/plain
             Date: Wed, ...
Forms : {}
Headers : {[x-envoy-upstream-service-time, 22], [x-envoy-decorator-operation, kubernetes-bootcamp.default.svc.cluster.local:8080/*], [transfer-encoding, chunked], [Content-Type, text/plain]...}
Images : {}
InputFields : {}
Links : {}
ParsedHtml : mshtml.HTMLDocumentClass
RawContentLength : 84


PS C:\Windows\System32> curl http://127.0.0.1:58168

StatusCode : 200
StatusDescription : OK
Content : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-658f6cbd58-hpn12 | v=1
RawContent : HTTP/1.1 200 OK
             x-envoy-upstream-service-time: 3
             x-envoy-decorator-operation: kubernetes-bootcamp.default.svc.cluster.local:8080/*
             transfer-encoding: chunked
             Content-Type: text/plain
             Date: Wed, 1...
Forms : {}
Headers : {[x-envoy-upstream-service-time, 3], [x-envoy-decorator-operation, kubernetes-bootcamp.default.svc.cluster.local:8080/*], [transfer-encoding, chunked], [Content-Type, text/plain]...}
Images : {}
InputFields : {}

```

Scale Down

1. To scale down the Deployment to 2 replicas, run again the scale subcommand:

```
kubectl scale deployments/kubernetes-bootcamp --replicas=2
```

```
PS C:\Windows\System32> kubectl scale deployments/kubernetes-bootcamp --replicas=2
deployment.apps/kubernetes-bootcamp scaled
```

2. List the Deployments to check if the change was applied with the get deployments subcommand:

```
PS C:\Windows\System32> kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
bookinfo-gateway-istio  1/1     1           1           35h
details-v1      1/1     1           1           4d17h
kubernetes-bootcamp  2/2     2           2           60m
productpage-v1    1/1     1           1           4d17h
ratings-v1       1/1     1           1           4d17h
reviews-v1       1/1     1           1           4d17h
reviews-v2       1/1     1           1           4d17h
reviews-v3       1/1     1           1           4d17h
PS C:\Windows\System32>
```

Performing a Rolling Update

Update the version of the app

1. To view the current image version of the app, run the describe pods subcommand and look for the Image field:

```
kubectl describe pods
```

```
Name:           kubernetes-bootcamp-658f6cbd58-6x8wj
Namespace:      default
Priority:       0
Service Account: default
Node:           minikube/192.168.49.2
Start Time:    Thu, 16 Oct 2025 00:46:22 +0800
Labels:         app=kubernetes-bootcamp
                pod-template-hash=658f6cbd58
                security.istio.io/tlsMode=istio
                service.istio.io/canonical-name=kubernetes-bootcamp
                service.istio.io/canonical-revision=latest
                version=v1
Annotations:   istio.io/rev: default
                kubectl.kubernetes.io/default-container: kubernetes-bootcamp
                kubectl.kubernetes.io/default-logs-container: kubernetes-bootcamp
                prometheus.io/path: /stats/prometheus
                prometheus.io/port: 15020
                prometheus.io/scrape: true
                sidecar.istio.io/status:
                  {"initContainers":["istio-init","istio-proxy"],"containers":null,"volumes":["workload-socket","potential-socket","workload-certs","istio-..."]}
                Status: Running
                IP:   10.244.0.149
IPs:
  IP:   10.244.0.149
Controlled By: ReplicaSet/kubernetes-bootcamp-658f6cbd58
Init Containers:
  istio-init:
    Container ID: docker://1edebb4b6b7681188b4fe8c7be7a755c96d3541edeca619e19e01f0a642d38eb
    Image:        docker.io/istio/proxyv2:1.27.1
    Image ID:     docker-pullable://istio/proxyv2@sha256:d902f7f80be93a0521af5e5796d1e989f2107af0725eca8881eb4c984
bf
    Port:        <none>
    Host Port:   <none>
    Args:
      istio-iptables
      -p
      15001
      -z
      15006
      -u
      1337

```

2. To update the image of the application to version 2, use the set

image subcommand, followed by the deployment name and the new image version:

```
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:v2
deployment.apps/kubernetes-bootcamp image updated
```

3. The command notified the Deployment to use a different image for your app and initiated a rolling update. Check the status of the new Pods, and view the old one terminating with the get pods subcommand:

```
kubectl get pods
```

```
PS C:\Windows\System32> kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
bookinfo-gateway-istio-6bbf6978f9-9kcv5   1/1     Running   3 (64m ago)   35h
details-v1-77d6bd5675-n4zrp                2/2     Running   4 (64m ago)   4d17h
kubernetes-bootcamp-57cc954bb9-587fl      2/2     Running   0           16s
kubernetes-bootcamp-57cc954bb9-d824c      2/2     Running   0           30s
kubernetes-bootcamp-658f6cbd58-5qk2c      2/2     Terminating   0           13m
kubernetes-bootcamp-658f6cbd58-6x8wj      2/2     Terminating   0           62m
productpage-v1-bb87ff47b-r75r5              2/2     Running   4 (64m ago)   4d17h
ratings-v1-8589f64b4c-sntz4                2/2     Running   4 (64m ago)   4d17h
reviews-v1-8cf7b9cc5-scgpl                 2/2     Running   4 (64m ago)   4d17h
reviews-v2-67d565655f-4c6kt                2/2     Running   4 (64m ago)   4d17h
reviews-v3-d587fc9d7-w92jc                2/2     Running   4 (64m ago)   4d17h
PS C:\Windows\System32> kubectl get pods
NAME                               READY   STATUS    RESTARTS   AGE
bookinfo-gateway-istio-6bbf6978f9-9kcv5   1/1     Running   3 (64m ago)   35h
details-v1-77d6bd5675-n4zrp                2/2     Running   4 (64m ago)   4d17h
kubernetes-bootcamp-57cc954bb9-587fl      2/2     Running   0           47s
kubernetes-bootcamp-57cc954bb9-d824c      2/2     Running   0           61s
productpage-v1-bb87ff47b-r75r5              2/2     Running   4 (64m ago)   4d17h
ratings-v1-8589f64b4c-sntz4                2/2     Running   4 (64m ago)   4d17h
reviews-v1-8cf7b9cc5-scgpl                 2/2     Running   4 (64m ago)   4d17h
reviews-v2-67d565655f-4c6kt                2/2     Running   4 (64m ago)   4d17h
reviews-v3-d587fc9d7-w92jc                2/2     Running   4 (64m ago)   4d17h
```

Verify an update

1. First, check that the service is running, as you might have deleted it in previous tutorial step, run:

```
describe services/kubernetes-bootcamp.
```

```
PS C:\Windows\System32> kubectl describe services/kubernetes-bootcamp
Name:           kubernetes-bootcamp
Namespace:      default
Labels:         app=kubernetes-bootcamp
Annotations:   <none>
Selector:       app=kubernetes-bootcamp
Type:          NodePort
IP Family Policy: SingleStack
IP Families:   IPv4
IP:            10.105.186.187
IPs:           10.105.186.187
Port:          <unset>  8080/TCP
TargetPort:    8080/TCP
NodePort:      <unset>  32003/TCP
Endpoints:    10.244.0.158:8080,10.244.0.159:8080
Session Affinity: None
External Traffic Policy: Cluster
Internal Traffic Policy: Cluster
Events:        <none>
```

2. If it's missing, you can create it again with:

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

3. Next, do a curl to the exposed IP and port:

```
PS C:\Windows\System32> $NODE_PORT="32003"
PS C:\Windows\System32> curl http://127.0.0.1:$NODE_PORT

StatusCode : 200
StatusDescription : OK
Content : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-57cc954bb9-587f1 | v=2
RawContent : HTTP/1.1 200 OK
x-envoy-upstream-service-time: 9
x-envoy-decorator-operation: kubernetes-bootcamp.default.svc.cluster.local:8080/*
transfer-encoding: chunked
Content-Type: text/plain
Date: Wed, 1...
Forms : {}
Headers : {[x-envoy-upstream-service-time, 9], [x-envoy-decorator-operation, kubernetes-bootcamp.default.svc.cluster.local:8080/*], [transfer-encoding, chunked], [Content-Type, text/plain]...}
Images : {}
InputFields : {}
Links : {}
ParsedHtml : mshtml.HTMLDocumentClass
RawContentLength : 84
```

Notice that all Pods are now running the latest version (v2).

4. You can also confirm the update by running the rollout status subcommand:

```
kubectl rollout status deployments/kubernetes-bootcamp
```

```
PS C:\Windows\System32> kubectl rollout status deployments/kubernetes-bootcamp
deployment "kubernetes-bootcamp" successfully rolled out
```

Roll back an update

1. Let's perform another update, and try to deploy an image tagged with v10:

```
kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=gcr.io/google-samples/kubernetes-bootcamp:v10
```

```
PS C:\Windows\System32> kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=gcr.io/google-samples/kubernetes-bootcamp:v10
deployment.apps/kubernetes-bootcamp image updated
```

2. Use get deployments to see the status of the deployment:

```
kubectl get deployments
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
bookinfo-gateway-istio	1/1	1	1	35h
details-v1	1/1	1	1	4d17h
kubernetes-bootcamp	2/2	1	2	69m
productpage-v1	1/1	1	1	4d17h
ratings-v1	1/1	1	1	4d17h
reviews-v1	1/1	1	1	4d17h
reviews-v2	1/1	1	1	4d17h
reviews-v3	1/1	1	1	4d17h

3. Notice that the output doesn't list the desired number of available Pods. Run the get pods subcommand to list all Pods:

```
kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
bookinfo-gateway-istio-6bbf6978f9-9kcv5	1/1	Running	3 (70m ago)	35h
details-v1-77d6bd5675-n4zrp	2/2	Running	4 (70m ago)	4d17h
kubernetes-bootcamp-57cc954bb9-587fl	2/2	Running	0	7m5s
kubernetes-bootcamp-57cc954bb9-d824c	2/2	Running	0	7m19s
kubernetes-bootcamp-677ff875c4-kqjc4	1/2	ErrImagePull	0	52s
productpage-v1-bb87ff47b-r75r5	2/2	Running	4 (70m ago)	4d17h
ratings-v1-8589f64b4c-sntz4	2/2	Running	4 (70m ago)	4d17h
reviews-v1-8cf7b9cc5-scgpl	2/2	Running	4 (70m ago)	4d17h
reviews-v2-67d565655f-4c6kt	2/2	Running	4 (70m ago)	4d17h
reviews-v3-d587fc9d7-w92jc	2/2	Running	4 (70m ago)	4d17h

Notice that some of the Pods have a status of ImagePullBackOff.

4. To get more insight into the problem, run the describe pods subcommand:

```
kubectl describe pods
```

Events:				
Type	Reason	Age	From	Message
Normal	Scheduled	2m28s	default-scheduler	Successfully assigned default/kubernetes-bootcamp-677ff875c4-kqjc4 to minikube
Normal	Pulled	2m27s	kubelet	Container image "docker.io/istio/proxyv2:1.27.1" already present on machine
Normal	Created	2m27s	kubelet	Created container: istio-init
Normal	Started	2m27s	kubelet	Started container istio-init
Normal	Pulled	2m26s	kubelet	Container image "docker.io/istio/proxyv2:1.27.1" already present on machine
Normal	Created	2m26s	kubelet	Created container: istio-proxy
Normal	Started	2m25s	kubelet	Started container istio-proxy
Warning	Unhealthy	2m25s	kubelet	Startup probe failed: Get "http://10.244.0.160:15021/healthz/ready": dial tcp 10.244.0.160:15021: connect: connection refused
Normal	Pulling	35s (x4 over 2m23s)	kubelet	Pulling image "gcr.io/google-samples/kubernetes-bootcamp:v10"
Warning	Failed	30s (x4 over 2m17s)	kubelet	Failed to pull image "gcr.io/google-samples/kubernetes-bootcamp:v10": Error response from daemon: manifest for gcr.io/google-samples/kubernetes-bootcamp:v10 not found: manifest unknown: Failed to fetch "v10"
Warning	Failed	30s (x4 over 2m17s)	kubelet	Error: ErrImagePull
Normal	BackOff	3s (x7 over 2m17s)	kubelet	Back-off pulling image "gcr.io/google-samples/kubernetes-bootcamp:v10"
Warning	Failed	3s (x7 over 2m17s)	kubelet	Error: ImagePullBackOff

5. To roll back the deployment to your last working version, use the rollout undo subcommand:

```
kubectl rollout undo deployments/kubernetes-bootcamp
```

PS C:\Windows\System32> kubectl rollout undo deployments/kubernetes-bootcamp
deployment.apps/kubernetes-bootcamp rolled back

6. Use the get pods subcommand to list the Pods again:

```
kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
bookinfo-gateway-istio-6bbf6978f9-9kcv5	1/1	Running	3 (75m ago)	35h
details-v1-77d6bd5675-n4zrp	2/2	Running	4 (75m ago)	4d17h
kubernetes-bootcamp-57cc954bb9-587fl	2/2	Running	0	11m
kubernetes-bootcamp-57cc954bb9-d824c	2/2	Running	0	12m
productpage-v1-bb87ff47b-r75r5	2/2	Running	4 (75m ago)	4d17h
ratings-v1-8589f64b4c-sntz4	2/2	Running	4 (75m ago)	4d17h
reviews-v1-8cf7b9cc5-scgpl	2/2	Running	4 (75m ago)	4d17h
reviews-v2-67d565655f-4c6kt	2/2	Running	4 (75m ago)	4d17h
reviews-v3-d587fc9d7-w92jc	2/2	Running	4 (75m ago)	4d17h

7. Remember to clean up your local cluster.

```
kubectl delete deployments/kubernetes-bootcamp services/kubernetes-bootcamp
```

PS C:\Windows\System32> kubectl delete deployments/kubernetes-bootcamp services/kubernetes-bootcamp deployment.apps "kubernetes-bootcamp" deleted
service "kubernetes-bootcamp" deleted