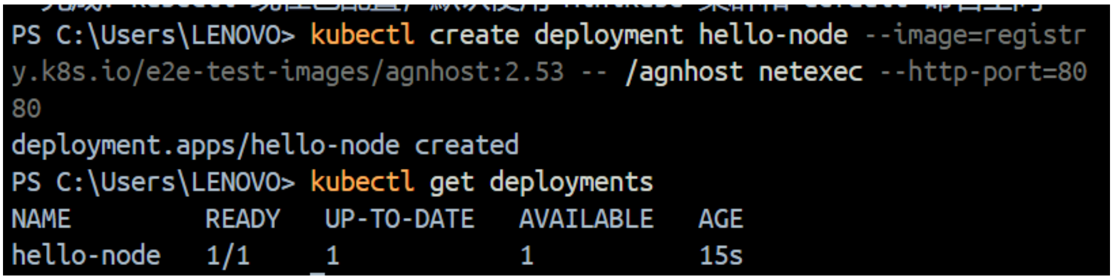
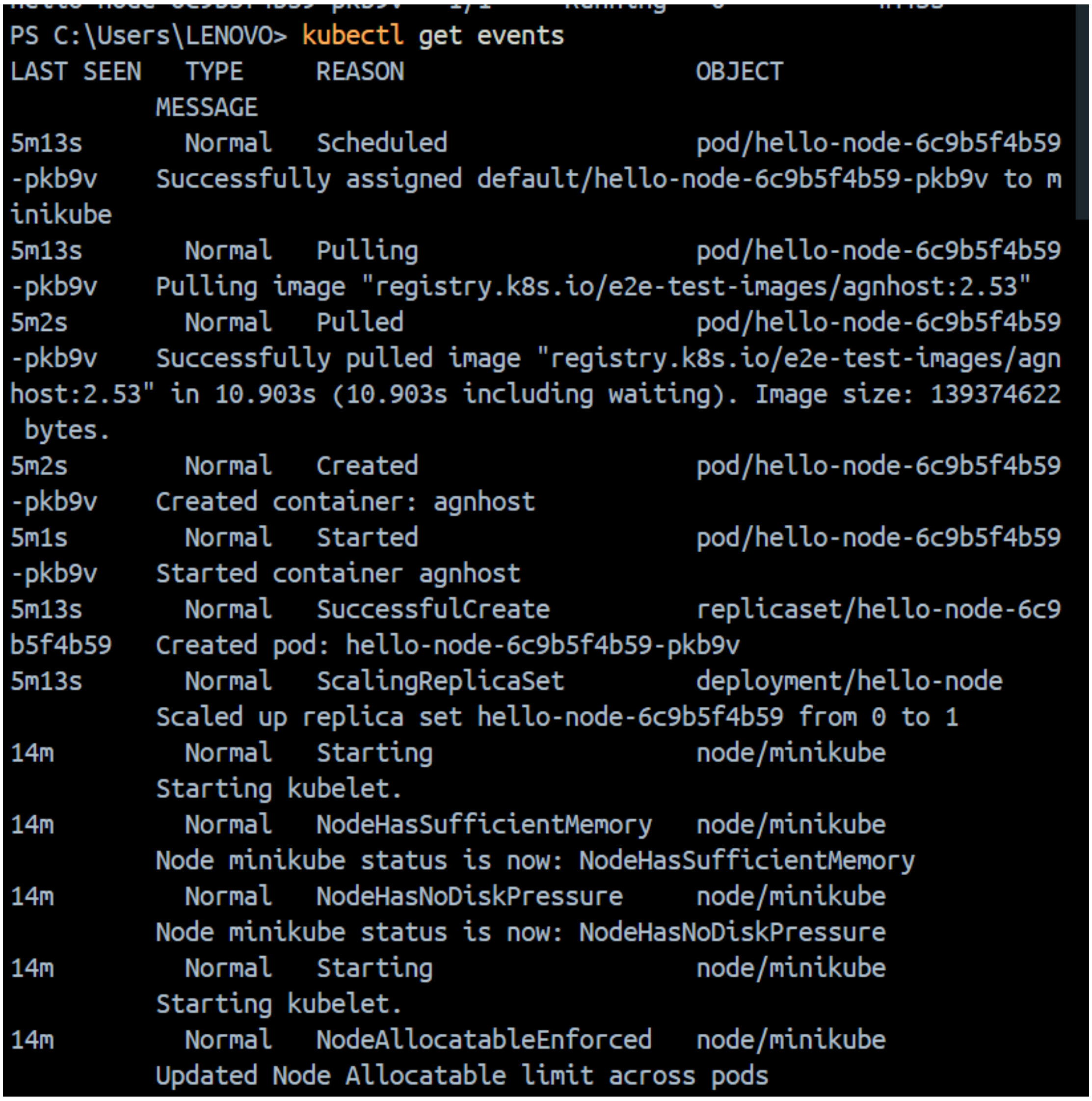
Use the kubectl create command to create a Deployment that manages a Pod. The Pod runs a Container based on the provided Docker image.

View the Deployment



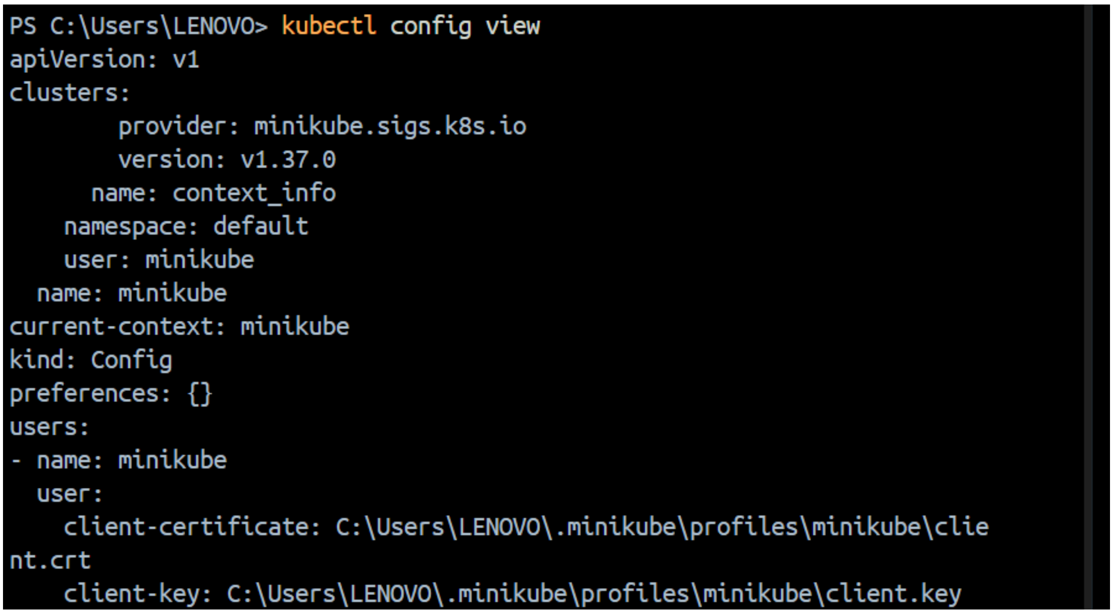
```
PS C:\Users\LENOVO> kubectl create deployment hello-node --image=registr
y.k8s.io/e2e-test-images/agnhost:2.53 -- /agnhost netexec --http-port=80
80
deployment.apps/hello-node created
PS C:\Users\LENOVO> kubectl get deployments
NAME         READY   UP-TO-DATE   AVAILABLE   AGE
hello-node   1/1     1            1           15s
```

View cluster events

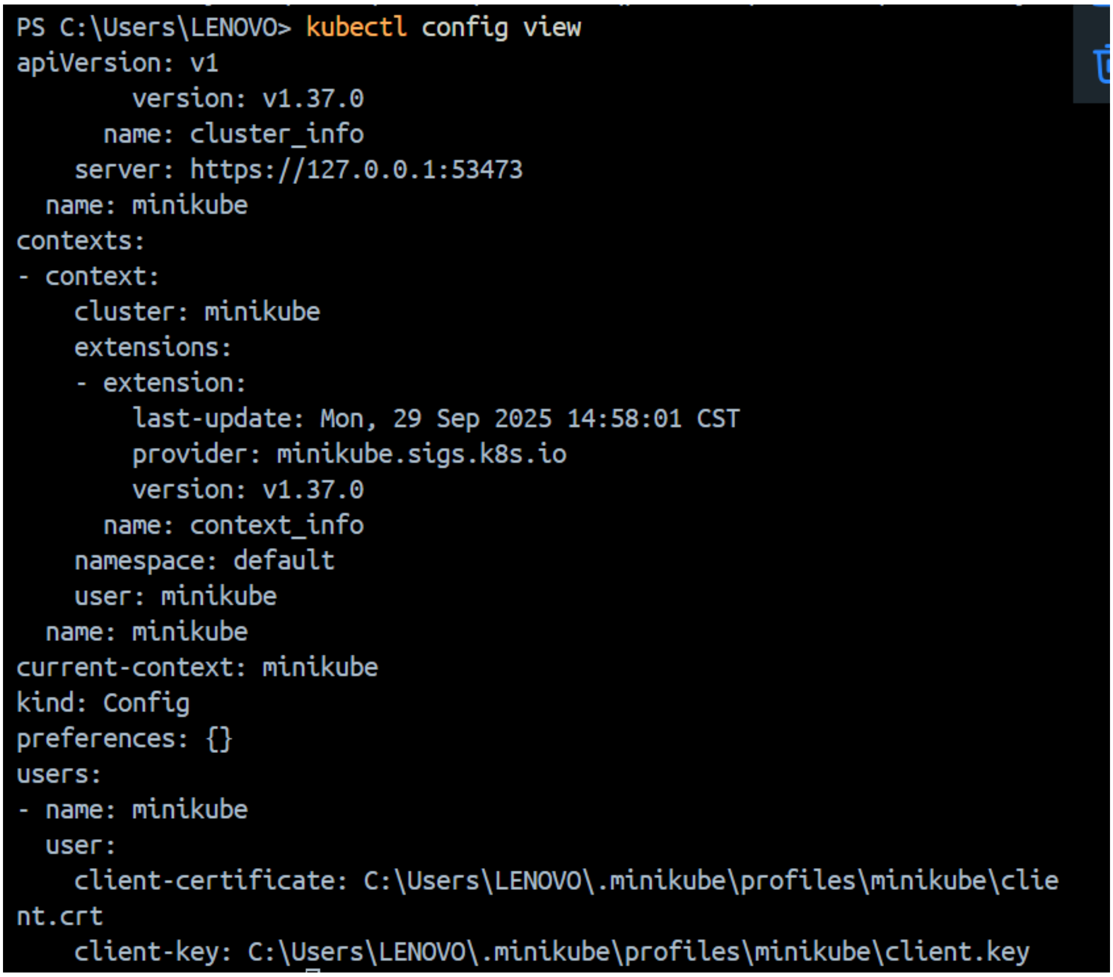


```
PS C:\Users\LENOVO> kubectl get events
LAST SEEN   TYPE     REASON                    OBJECT
          MESSAGE
5m13s       Normal   Scheduled                 pod/hello-node-6c9b5f4b59
-pkb9v    Successfully assigned default/hello-node-6c9b5f4b59-pkb9v to m
inikube
5m13s       Normal   Pulling                   pod/hello-node-6c9b5f4b59
-pkb9v    Pulling image "registry.k8s.io/e2e-test-images/agnhost:2.53"
5m2s        Normal   Pulled                    pod/hello-node-6c9b5f4b59
-pkb9v    Successfully pulled image "registry.k8s.io/e2e-test-images/agn
host:2.53" in 10.903s (10.903s including waiting). Image size: 139374622
 bytes.
5m2s        Normal   Created                   pod/hello-node-6c9b5f4b59
-pkb9v    Created container: agnhost
5m1s        Normal   Started                   pod/hello-node-6c9b5f4b59
-pkb9v    Started container agnhost
5m13s       Normal   SuccessfulCreate          replicaset/hello-node-6c9
b5f4b59   Created pod: hello-node-6c9b5f4b59-pkb9v
5m13s       Normal   ScalingReplicaSet         deployment/hello-node
          Scaled up replica set hello-node-6c9b5f4b59 from 0 to 1
14m         Normal   Starting                  node/minikube
          Starting kubelet.
14m         Normal   NodeHasSufficientMemory   node/minikube
          Node minikube status is now: NodeHasSufficientMemory
14m         Normal   NodeHasNoDiskPressure     node/minikube
          Node minikube status is now: NodeHasNoDiskPressure
14m         Normal   Starting                  node/minikube
          Starting kubelet.
14m         Normal   NodeAllocatableEnforced   node/minikube
          Updated Node Allocatable limit across pods
```

View the kubectl configuration:



```
PS C:\Users\LENOVO> kubectl config view
apiVersion: v1
clusters:
        provider: minikube.sigs.k8s.io
        version: v1.37.0
      name: context_info
    namespace: default
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: C:\Users\LENOVO\.minikube\profiles\minikube\clie
nt.crt
    client-key: C:\Users\LENOVO\.minikube\profiles\minikube\client.key
```



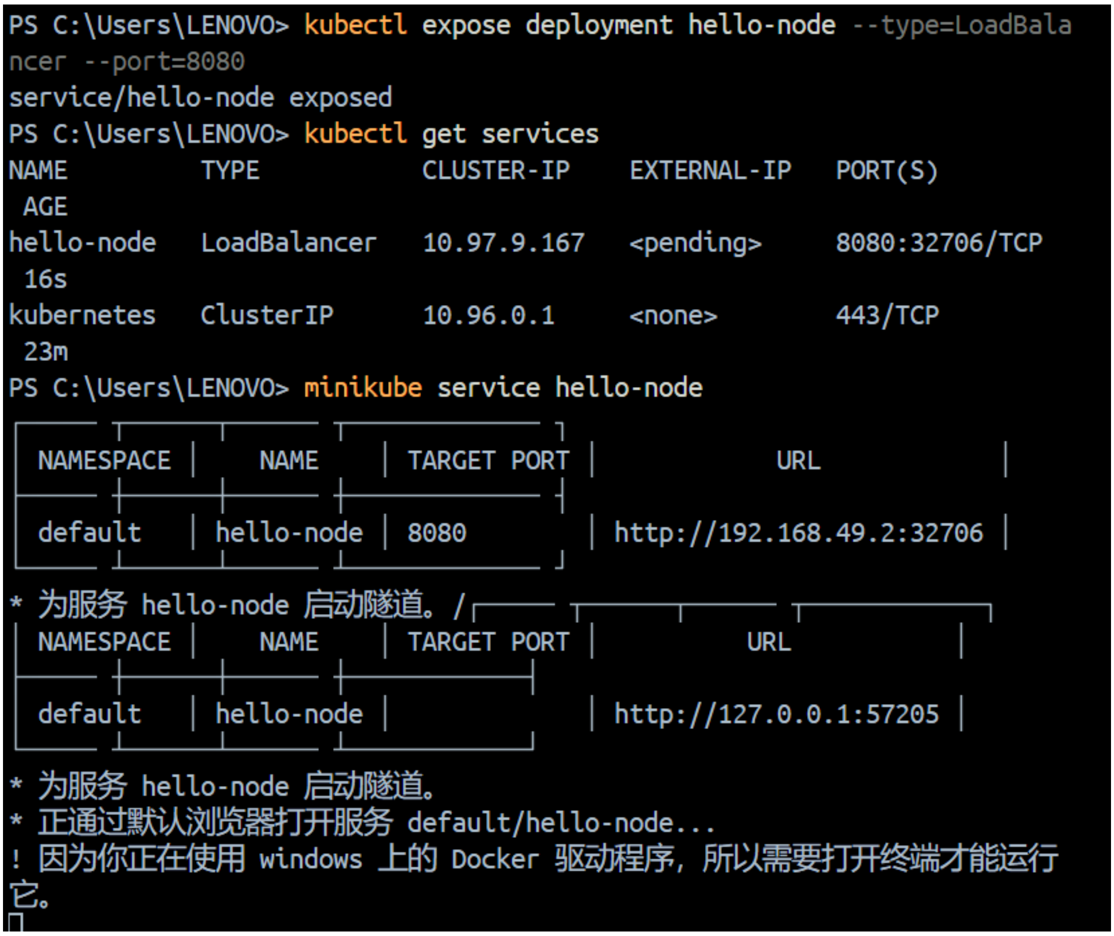
```
PS C:\Users\LENOVO> kubectl config view
apiVersion: v1
        version: v1.37.0
      name: cluster_info
    server: https://127.0.0.1:53473
  name: minikube
contexts:
- context:
    cluster: minikube
    extensions:
    - extension:
        last-update: Mon, 29 Sep 2025 14:58:01 CST
        provider: minikube.sigs.k8s.io
        version: v1.37.0
      name: context_info
    namespace: default
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: C:\Users\LENOVO\.minikube\profiles\minikube\clie
nt.crt
    client-key: C:\Users\LENOVO\.minikube\profiles\minikube\client.key
```

View application logs for a container in a pod (replace pod name with the one you got from kubectl get pods).

Create a Service

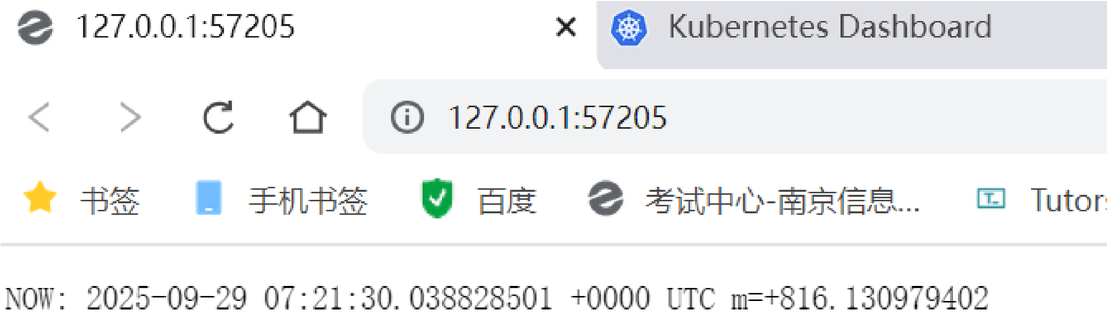
Expose the Pod to the public internet using the kubectl expose command:

View the Service you created:



```
PS C:\Users\LENOVO> kubectl expose deployment hello-node --type=LoadBala
ncer --port=8080
service/hello-node exposed
PS C:\Users\LENOVO> kubectl get services
NAME         TYPE           CLUSTER-IP    EXTERNAL-IP   PORT(S)
 AGE
hello-node   LoadBalancer   10.97.9.167   <pending>     8080:32706/TCP
 16s
kubernetes   ClusterIP      10.96.0.1     <none>        443/TCP
 23m
PS C:\Users\LENOVO> minikube service hello-node
┌───────────┬────────────┬─────────────┬───────────────────────────┐
│ NAMESPACE │    NAME    │ TARGET PORT │            URL            │
├───────────┼────────────┼─────────────┼───────────────────────────┤
│ default   │ hello-node │ 8080        │ http://192.168.49.2:32706 │
└───────────┴────────────┴─────────────┴───────────────────────────┘
* 为服务 hello-node 启动隧道。/┌───────────┬────────────┬─────────────┬────────────────────────┐
│ NAMESPACE │    NAME    │ TARGET PORT │          URL           │
├───────────┼────────────┼─────────────┼────────────────────────┤
│ default   │ hello-node │             │ http://127.0.0.1:57205 │
└───────────┴────────────┴─────────────┴────────────────────────┘
* 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。
```

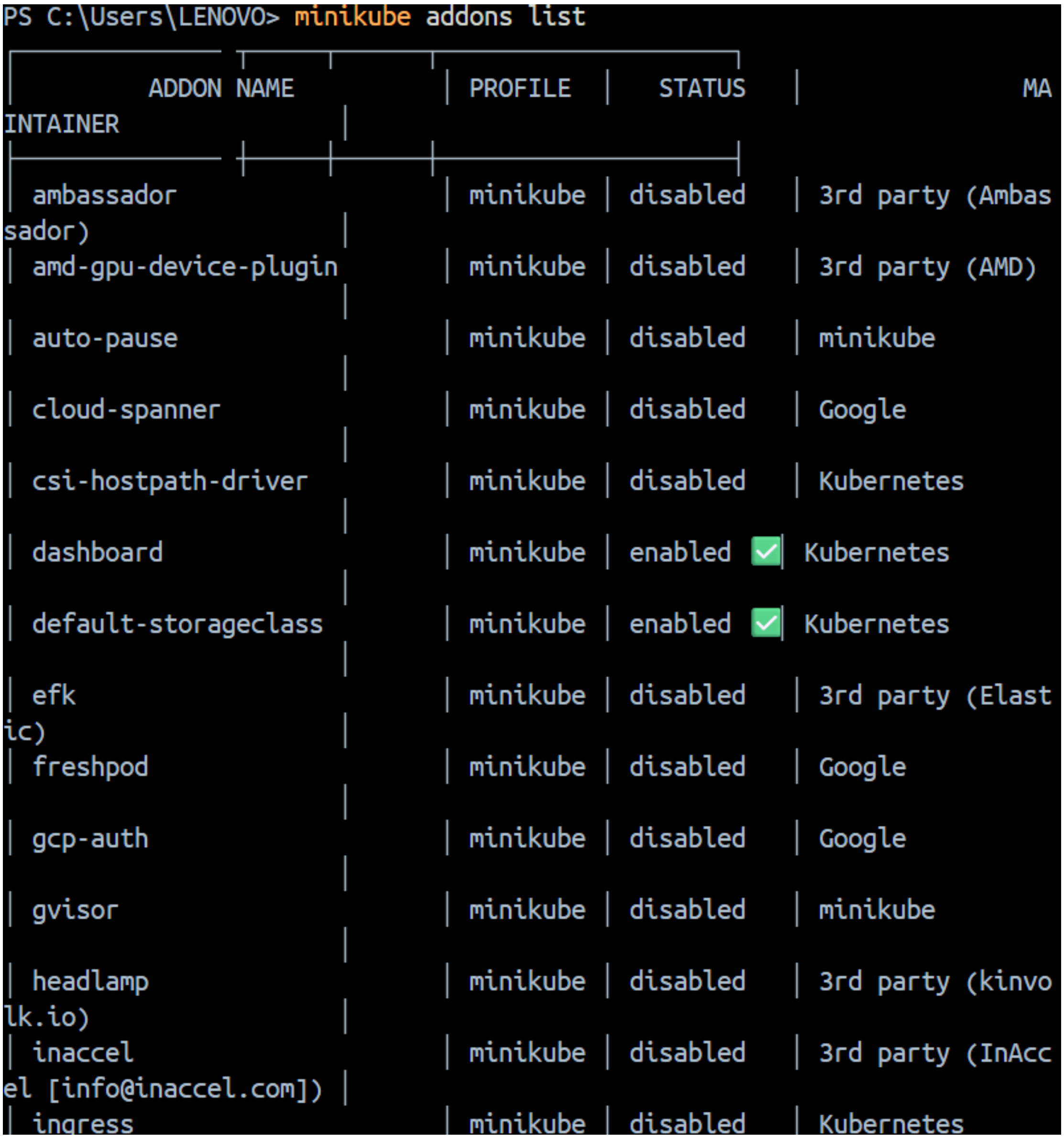
Run the command to opens up a browser windoe



```
NOW: 2025-09-29 07:21:30.038828501 +0000 UTC m=+816.130979402
```

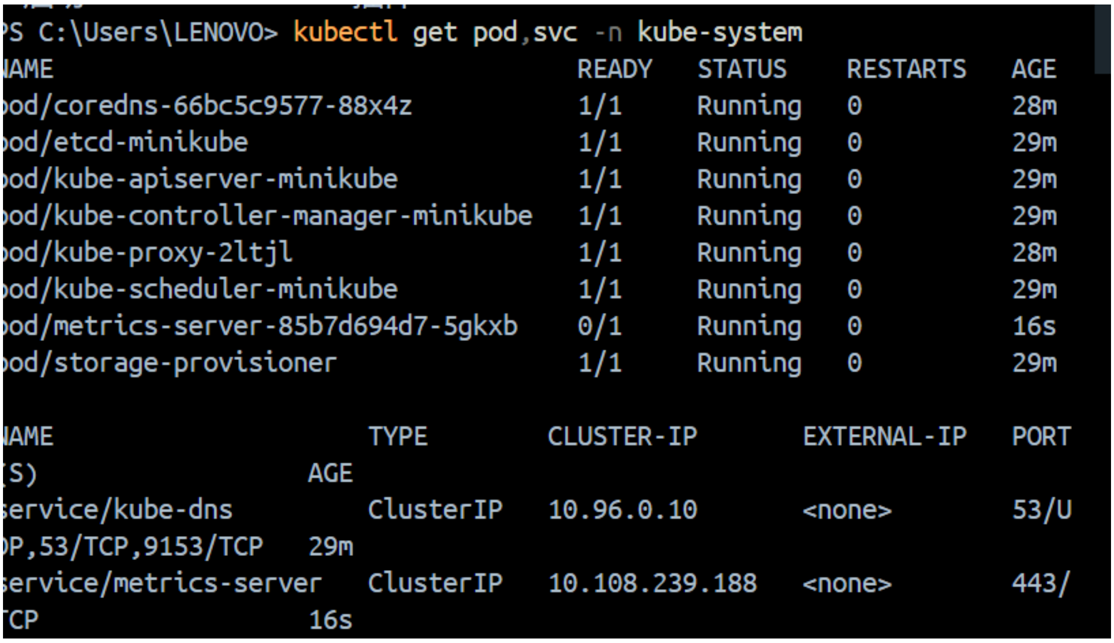
Enable addons

List the currently supported addons:



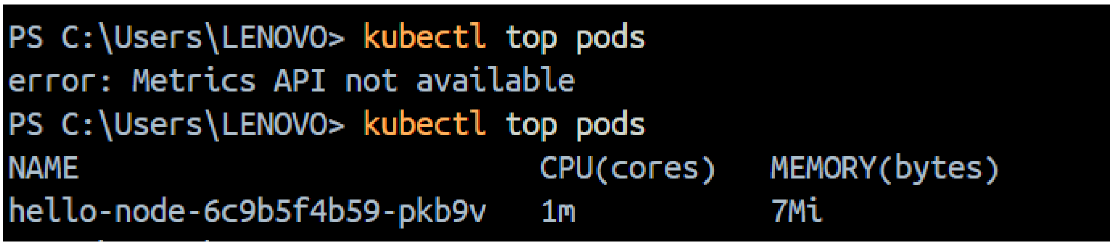
```
PS C:\Users\LENOVO> minikube addons list
┌──────────────┬┬─────┬────────────┐
│        ADDON NAME              │ PROFILE  │   STATUS    │              MA
INTAINER               │
├──────────────┼┼─────┼────────────┤
│ ambassador                     │ minikube │ disabled    │ 3rd party (Ambas
sador)                 │
│ 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 ✅ │ Kubernetes
                       │
│ default-storageclass           │ minikube │ enabled ✅ │ Kubernetes
                       │
│ efk                            │ minikube │ disabled    │ 3rd party (Elast
ic)                    │
│ freshpod                       │ minikube │ disabled    │ Google
                       │
│ gcp-auth                       │ minikube │ disabled    │ Google
                       │
│ gvisor                         │ minikube │ disabled    │ minikube
                       │
│ headlamp                       │ minikube │ disabled    │ 3rd party (kinvo
lk.io)                 │
│ inaccel                        │ minikube │ disabled    │ 3rd party (InAcc
el [info@inaccel.com]) │
│ ingress                        │ minikube │ disabled    │ Kubernetes
```

View the Pod and Service you created by installing that addon



```
S C:\Users\LENOVO> kubectl get pod,svc -n kube-system
AME                                 READY   STATUS    RESTARTS   AGE
od/coredns-66bc5c9577-88x4z          1/1     Running   0          28m
od/etcd-minikube                     1/1     Running   0          29m
od/kube-apiserver-minikube           1/1     Running   0          29m
od/kube-controller-manager-minikube  1/1     Running   0          29m
od/kube-proxy-2ltjl                  1/1     Running   0          28m
od/kube-scheduler-minikube           1/1     Running   0          29m
od/metrics-server-85b7d694d7-5gkxb   0/1     Running   0          16s
od/storage-provisioner               1/1     Running   0          29m

AME                        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT
S)                     AGE
ervice/kube-dns            ClusterIP   10.96.0.10       <none>        53/U
P,53/TCP,9153/TCP      29m
ervice/metrics-server      ClusterIP   10.108.239.188   <none>        443/
CP                     16s
```

Check the output from metrics-server

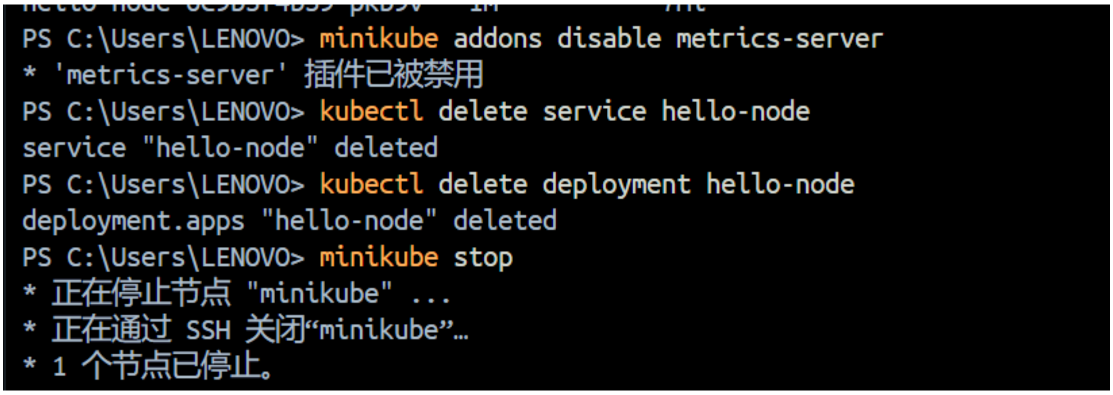


```
PS C:\Users\LENOVO> kubectl top pods
error: Metrics API not available
PS C:\Users\LENOVO> kubectl top pods
NAME                          CPU(cores)   MEMORY(bytes)
hello-node-6c9b5f4b59-pkb9v   1m           7Mi
```

Disable metrics-server

Now you can clean up the resources you created in your cluster:

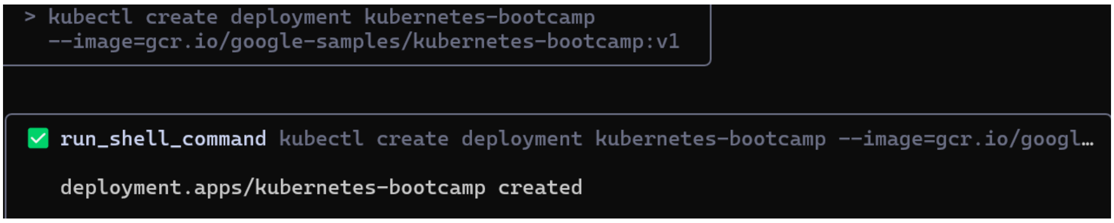
Stop the Minikube cluster



```
hello-node-6c9b5f4b59-pkb9v   1m           7Mi
PS C:\Users\LENOVO> minikube addons disable metrics-server
* 'metrics-server' 插件已被禁用
PS C:\Users\LENOVO> kubectl delete service hello-node
service "hello-node" deleted
PS C:\Users\LENOVO> kubectl delete deployment hello-node
deployment.apps "hello-node" deleted
PS C:\Users\LENOVO> minikube stop
* 正在停止节点 "minikube" ...
* 正在通过 SSH 关闭“minikube”…
* 1 个节点已停止。
```

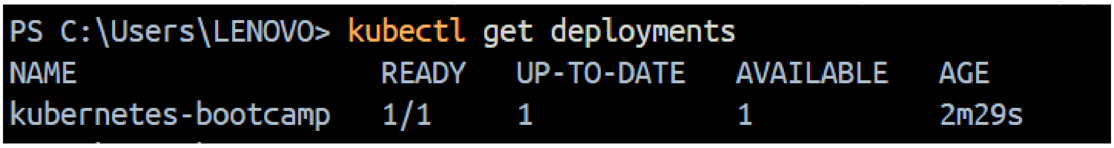
Deploy an app

Let’s deploy our first app on Kubernetes with the kubectl create deployment command. We need to provide the deployment name and app image location (include the full repository url for images hosted outside Docker Hub).



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

✅ run_shell_command kubectl create deployment kubernetes-bootcamp --image=gcr.io/googl…

   deployment.apps/kubernetes-bootcamp created
```

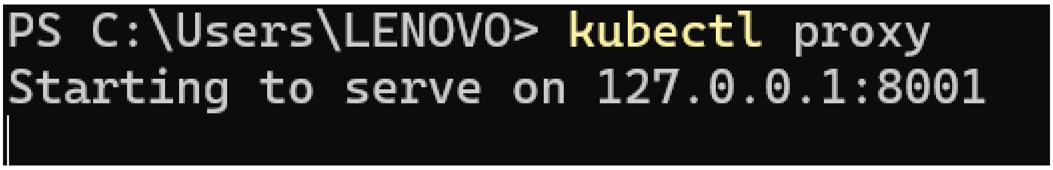
list the deployments



```
PS C:\Users\LENOVO> kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   1/1     1            1           2m29s
```

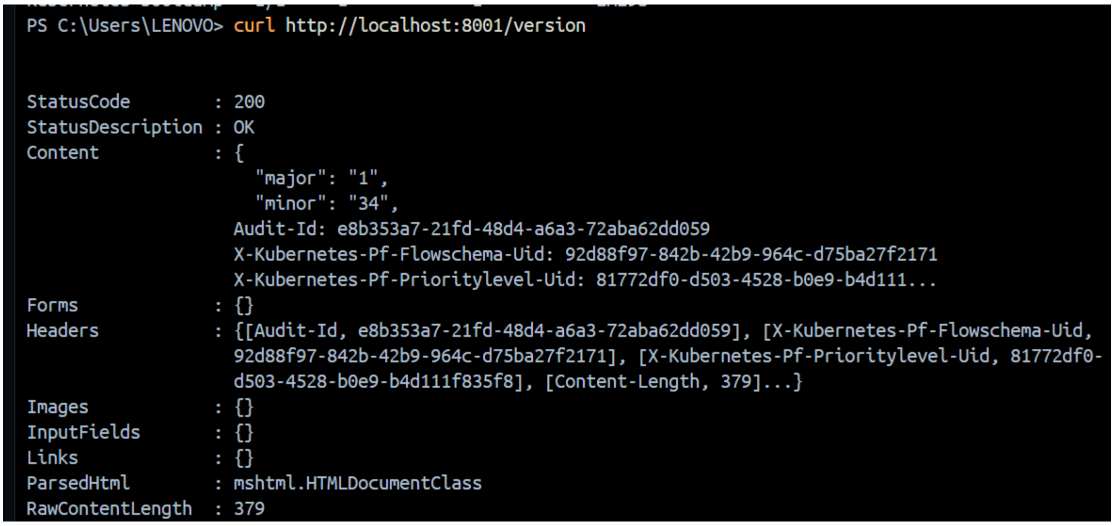
View the app

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:\Users\LENOVO> kubectl proxy
Starting to serve on 127.0.0.1:8001
```

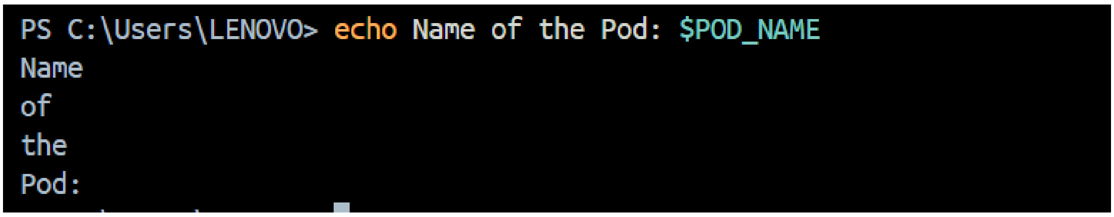
You can see all those APIs hosted through the proxy endpoint. For example, we can query the version directly through the API using the curl command:



```
PS C:\Users\LENOVO> curl http://localhost:8001/version


StatusCode        : 200
StatusDescription : OK
Content           : {
                      "major": "1",
                      "minor": "34",
                    Audit-Id: e8b353a7-21fd-48d4-a6a3-72aba62dd059
                    X-Kubernetes-Pf-Flowschema-Uid: 92d88f97-842b-42b9-964c-d75ba27f2171
                    X-Kubernetes-Pf-Prioritylevel-Uid: 81772df0-d503-4528-b0e9-b4d111...
Forms             : {}
Headers           : {[Audit-Id, e8b353a7-21fd-48d4-a6a3-72aba62dd059], [X-Kubernetes-Pf-Flowschema-Uid,
                    92d88f97-842b-42b9-964c-d75ba27f2171], [X-Kubernetes-Pf-Prioritylevel-Uid, 81772df0-
                    d503-4528-b0e9-b4d111f835f8], [Content-Length, 379]...}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 379
```

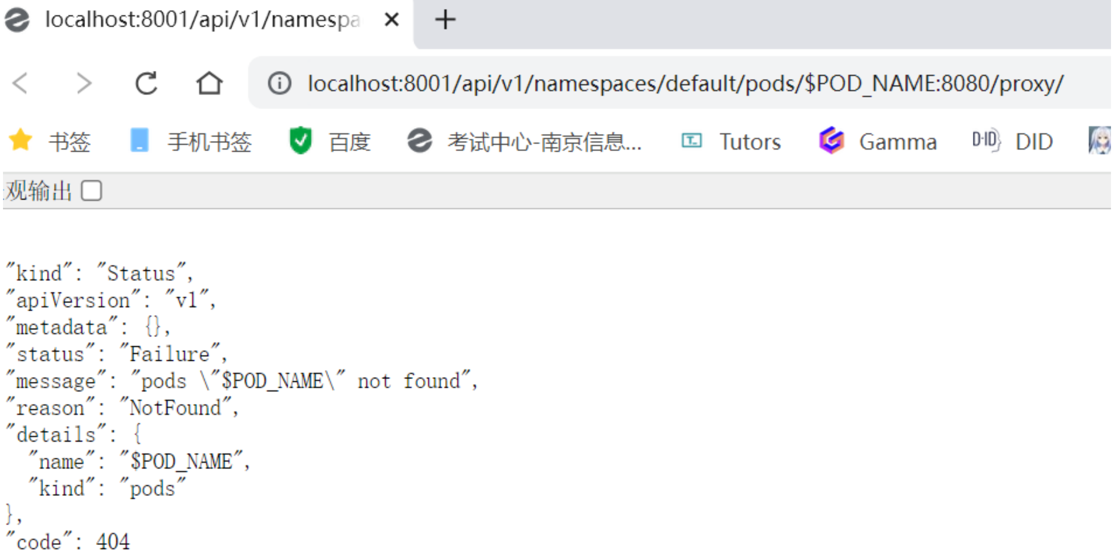
The API server will automatically create an endpoint for each pod, based on the pod name, that is also accessible through the proxy.

First get the Pod name, and store it in the environment variable POD_NAME.



```
PS C:\Users\LENOVO> echo Name of the Pod: $POD_NAME
Name
of
the
Pod:
```

access the Pod through the proxied API

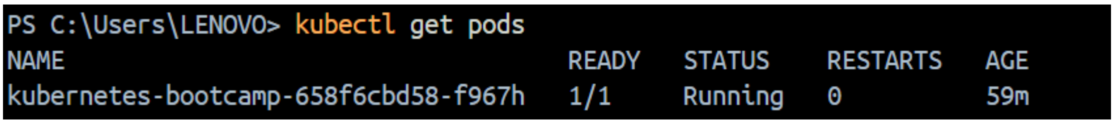


localhost:8001/api/v1/namespa
localhost:8001/api/v1/namespaces/default/pods/$POD_NAME:8080/proxy/
书签
手机书签
百度
考试中心-南京信息...
Tutors
Gamma
DID
观输出 □

```
"kind": "Status",
"apiVersion": "v1",
"metadata": {},
"status": "Failure",
"message": "pods \"$POD_NAME\" not found",
"reason": "NotFound",
"details": {
  "name": "$POD_NAME",
  "kind": "pods"
},
"code": 404
```

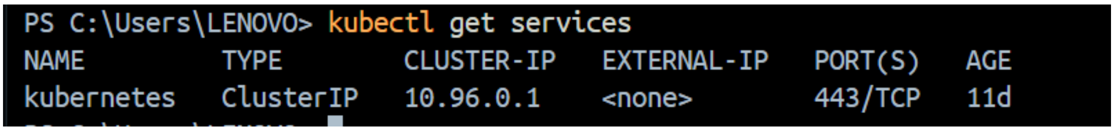
Services and Labels

use the kubectl get command and look for existing Pods:



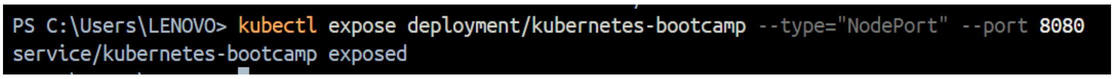
```
PS C:\Users\LENOVO> kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-658f6cbd58-f967h  1/1     Running   0          59m
```

list the current Services from our cluster:



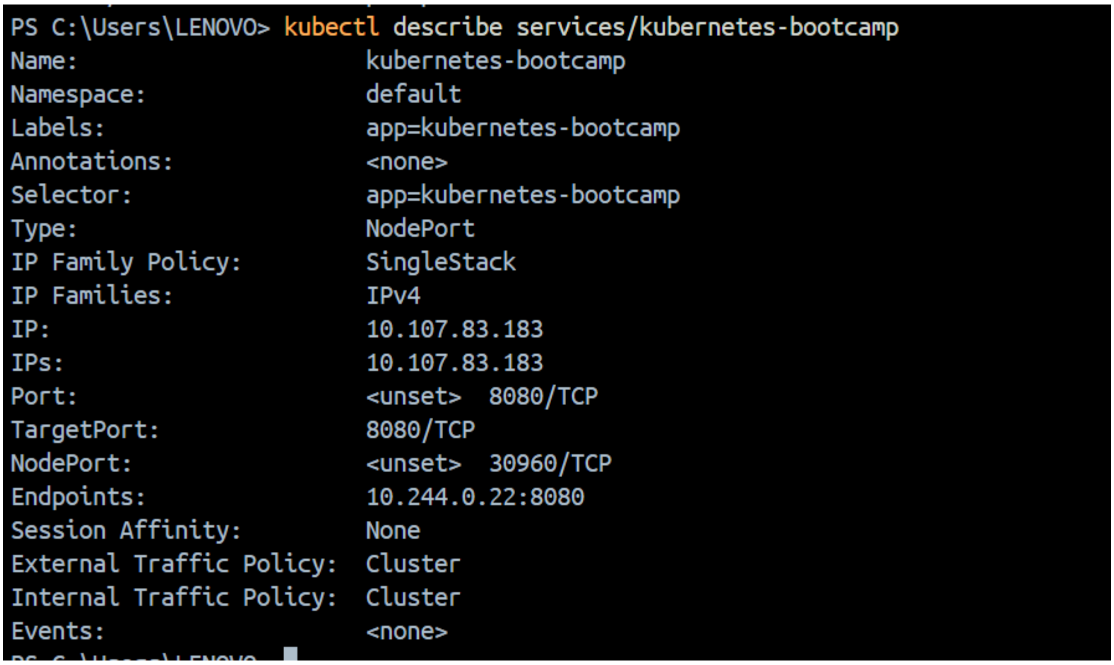
```
PS C:\Users\LENOVO> kubectl get services
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.96.0.1    <none>        443/TCP   11d
```

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



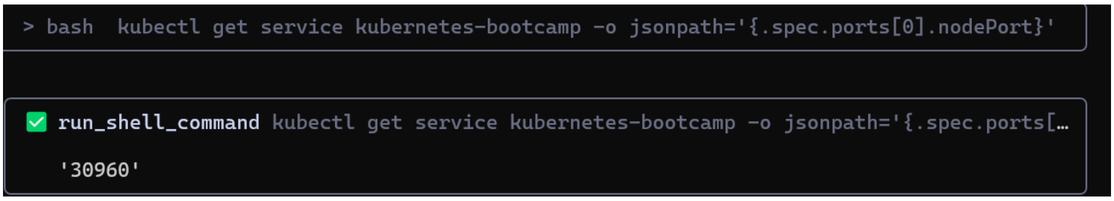
```
PS C:\Users\LENOVO> kubectl expose deployment/kubernetes-bootcamp --type="NodePort" --port 8080
service/kubernetes-bootcamp exposed
```

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

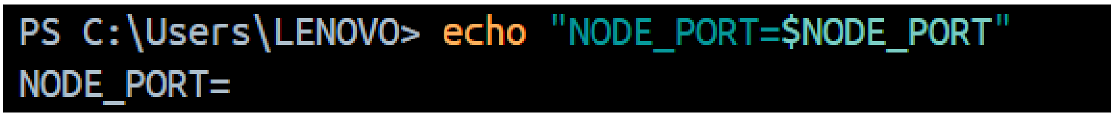


```
PS C:\Users\LENOVO> kubectl describe services/kubernetes-bootcamp
Name:                     kubernetes-bootcamp
Namespace:                default
Labels:                   app=kubernetes-bootcamp
Annotations:              <none>
Selector:                 app=kubernetes-bootcamp
Type:                     NodePort
IP Family Policy:         SingleStack
IP Families:              IPv4
IP:                       10.107.83.183
IPs:                      10.107.83.183
Port:                     <unset>  8080/TCP
TargetPort:               8080/TCP
NodePort:                 <unset>  30960/TCP
Endpoints:                10.244.0.22:8080
Session Affinity:         None
External Traffic Policy:  Cluster
Internal Traffic Policy:  Cluster
Events:                   <none>
PS C:\Users\LENOVO>
```

Create an environment variable called NODE_PORT that has the value of the Node port assigned:

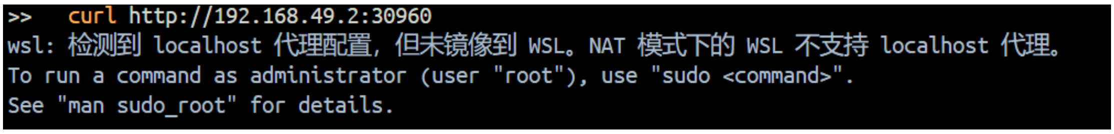


```
> bash  kubectl get service kubernetes-bootcamp -o jsonpath='{.spec.ports[0].nodePort}'

✅ run_shell_command kubectl get service kubernetes-bootcamp -o jsonpath='{.spec.ports[…

  '30960'
```



```
PS C:\Users\LENOVO> echo "NODE_PORT=$NODE_PORT"
NODE_PORT=
```

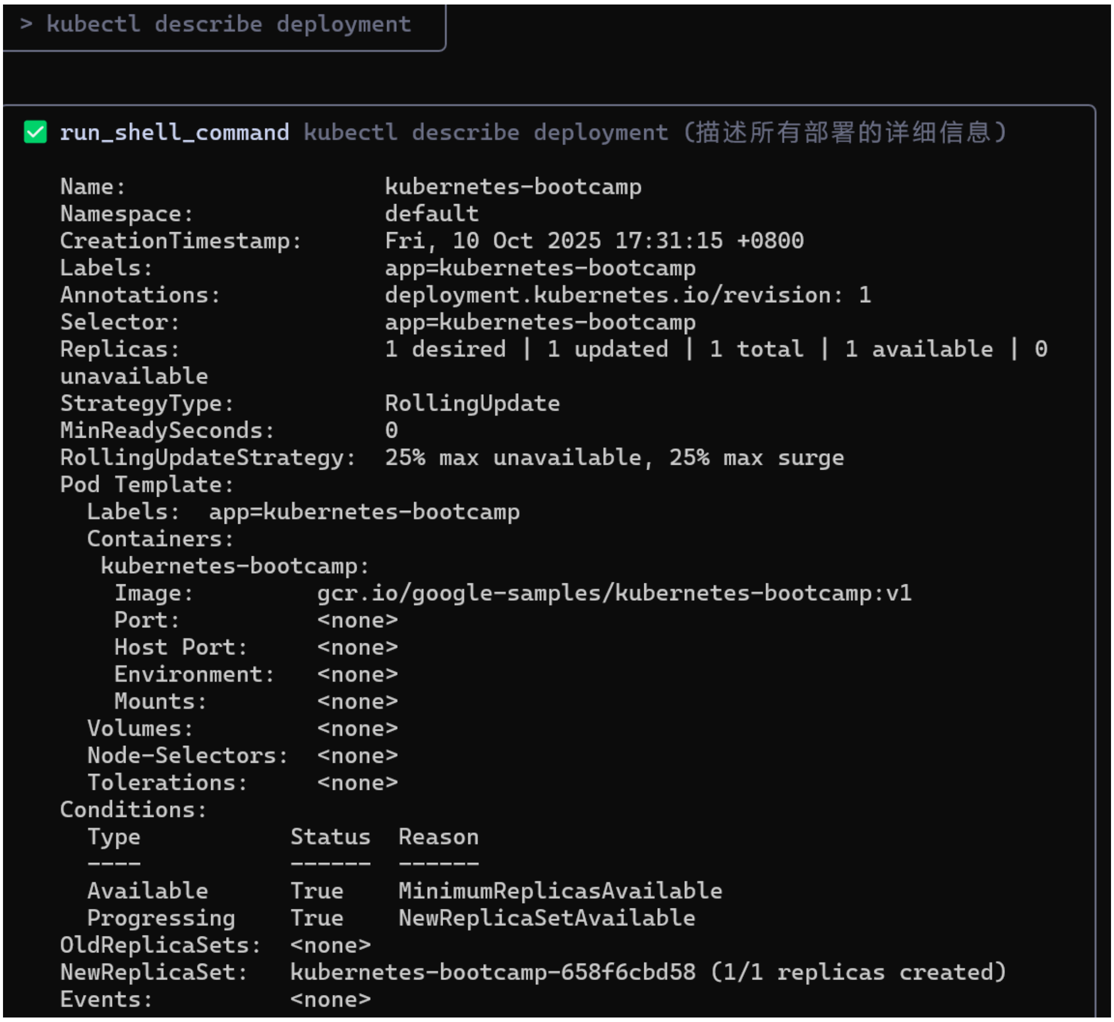
Now we can test that the app is exposed outside of the cluster using curl, the IP address of the Node and the externally exposed port:



```
>>   curl http://192.168.49.2:30960
wsl: 检测到 localhost 代理配置，但未镜像到 WSL。NAT 模式下的 WSL 不支持 localhost 代理。
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

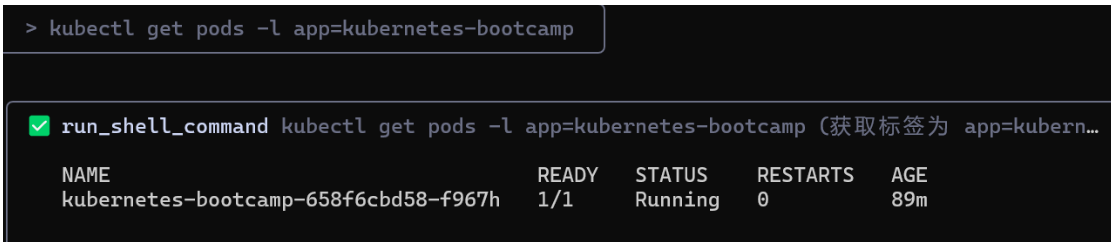
Using labels

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



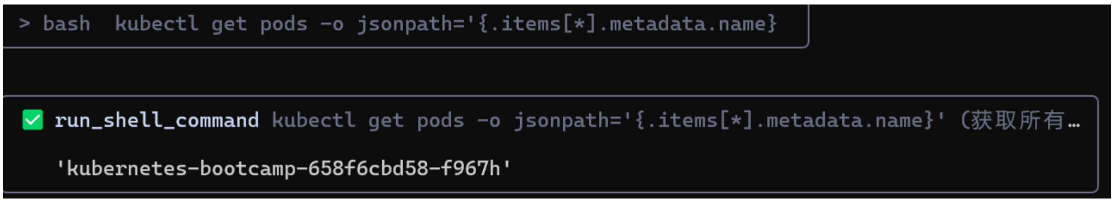
```
> kubectl describe deployment

✓ run_shell_command kubectl describe deployment (描述所有部署的详细信息)

  Name:                   kubernetes-bootcamp
  Namespace:              default
  CreationTimestamp:      Fri, 10 Oct 2025 17:31:15 +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:          <none>
```

use the kubectl get pods command with -l as a parameter, followed by the label values:

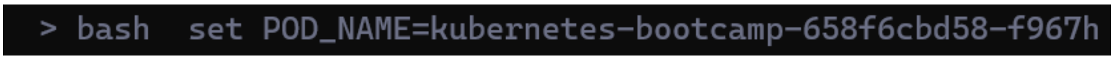


```
> kubectl get pods -l app=kubernetes-bootcamp

✓ run_shell_command kubectl get pods -l app=kubernetes-bootcamp (获取标签为 app=kubern…

  NAME                                 READY   STATUS    RESTARTS   AGE
  kubernetes-bootcamp-658f6cbd58-f967h   1/1     Running   0          89m
```

Get the name of the Pod and store it in the POD_NAME environment variable:

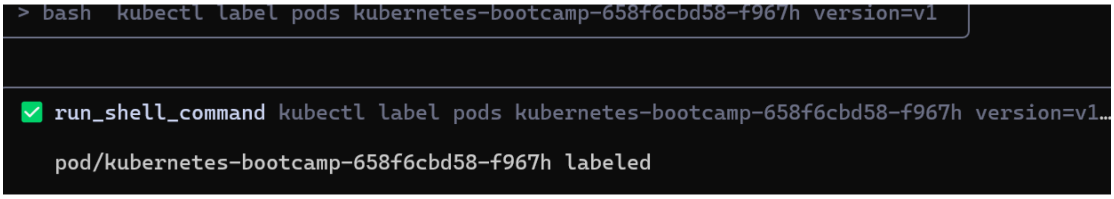


```
> bash  kubectl get pods -o jsonpath='{.items[*].metadata.name}

✓ run_shell_command kubectl get pods -o jsonpath='{.items[*].metadata.name}' (获取所有…

  'kubernetes-bootcamp-658f6cbd58-f967h'
```



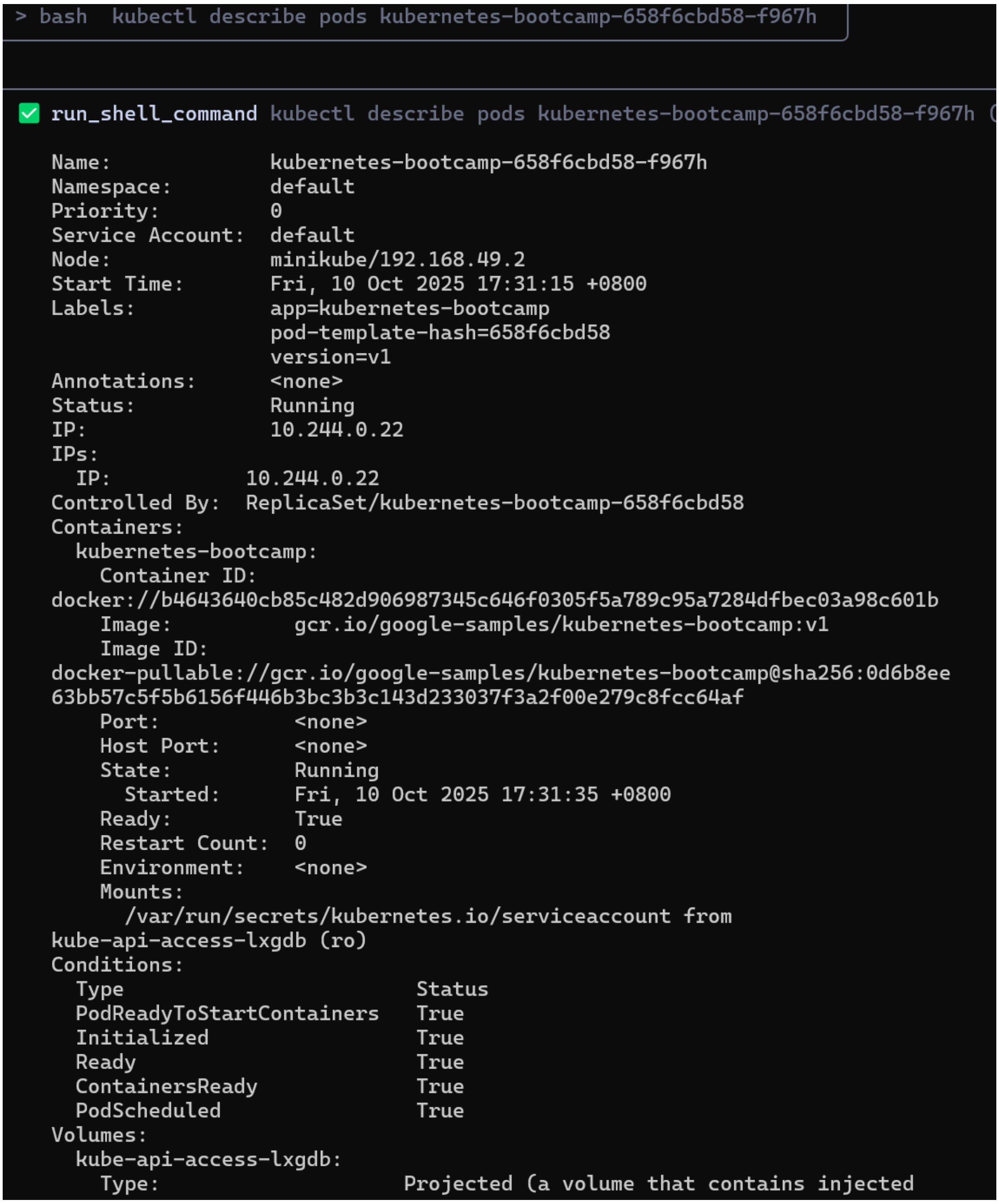
```
> bash  set POD_NAME=kubernetes-bootcamp-658f6cbd58-f967h
```

To apply a new label we use the label subcommand followed by the object type, object name and the new label:

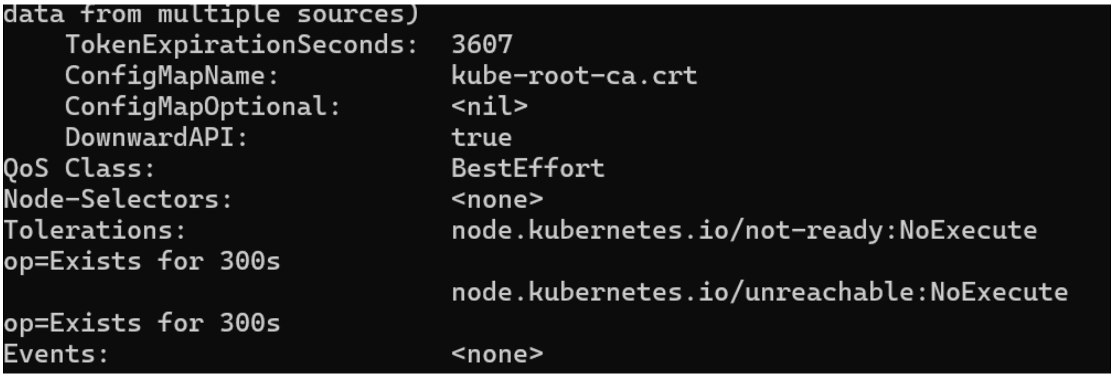


```
> bash  kubectl label pods kubernetes-bootcamp-658f6cbd58-f967h version=v1

✅ run_shell_command kubectl label pods kubernetes-bootcamp-658f6cbd58-f967h version=v1…

   pod/kubernetes-bootcamp-658f6cbd58-f967h 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

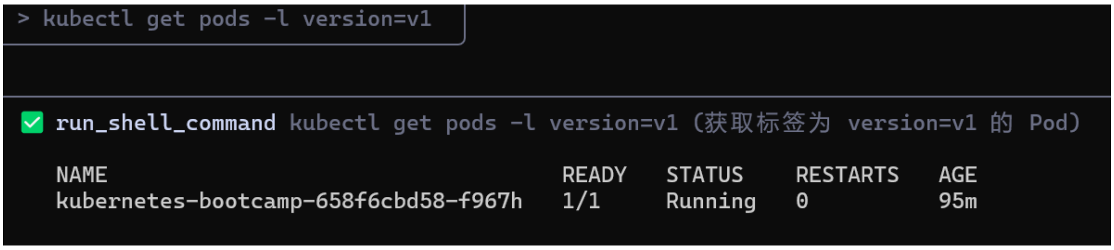


```
> bash  kubectl describe pods kubernetes-bootcamp-658f6cbd58-f967h

✅ run_shell_command kubectl describe pods kubernetes-bootcamp-658f6cbd58-f967h (

   Name:             kubernetes-bootcamp-658f6cbd58-f967h
   Namespace:        default
   Priority:         0
   Service Account:  default
   Node:             minikube/192.168.49.2
   Start Time:       Fri, 10 Oct 2025 17:31:15 +0800
   Labels:           app=kubernetes-bootcamp
                     pod-template-hash=658f6cbd58
                     version=v1
   Annotations:      <none>
   Status:           Running
   IP:               10.244.0.22
   IPs:
     IP:           10.244.0.22
   Controlled By:  ReplicaSet/kubernetes-bootcamp-658f6cbd58
   Containers:
     kubernetes-bootcamp:
       Container ID:
   docker://b4643640cb85c482d906987345c646f0305f5a789c95a7284dfbec03a98c601b
       Image:          gcr.io/google-samples/kubernetes-bootcamp:v1
       Image ID:
   docker-pullable://gcr.io/google-samples/kubernetes-bootcamp@sha256:0d6b8ee
   63bb57c5f5b6156f446b3bc3b3c143d233037f3a2f00e279c8fcc64af
       Port:           <none>
       Host Port:      <none>
       State:          Running
         Started:      Fri, 10 Oct 2025 17:31:35 +0800
       Ready:          True
       Restart Count:  0
       Environment:    <none>
       Mounts:
         /var/run/secrets/kubernetes.io/serviceaccount from
   kube-api-access-lxgdb (ro)
   Conditions:
     Type                        Status
     PodReadyToStartContainers   True
     Initialized                 True
     Ready                       True
     ContainersReady             True
     PodScheduled                True
   Volumes:
     kube-api-access-lxgdb:
       Type:                    Projected (a volume that contains injected
```



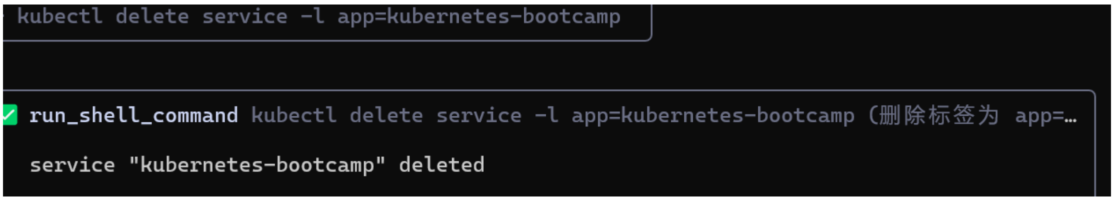
```
data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute
op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute
op=Exists for 300s
Events:                      <none>
```

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

✓ run_shell_command kubectl get pods -l version=v1 (获取标签为 version=v1 的 Pod)

  NAME                                   READY   STATUS    RESTARTS   AGE
  kubernetes-bootcamp-658f6cbd58-f967h   1/1     Running   0          95m
```

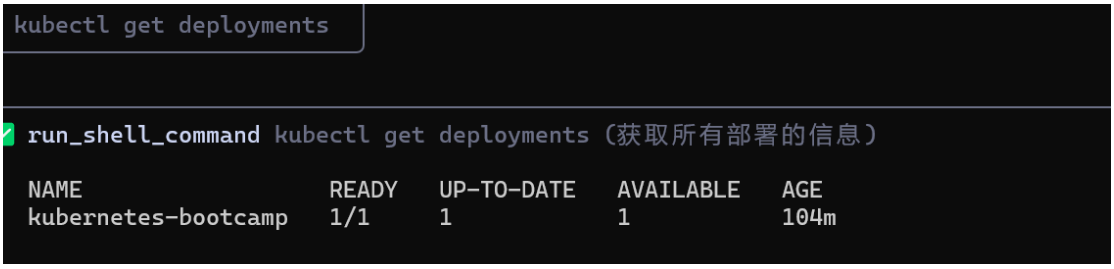
Deleting a service



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

✓ run_shell_command kubectl delete service -l app=kubernetes-bootcamp (删除标签为 app=…

  service "kubernetes-bootcamp" deleted
```

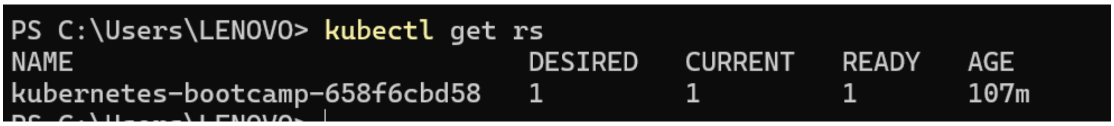
Scaling a Deployment

list Deployments



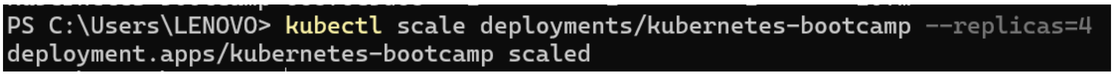
```
kubectl get deployments

✓ run_shell_command kubectl get deployments (获取所有部署的信息)

  NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
  kubernetes-bootcamp   1/1     1            1           104m
```

see the ReplicaSet created by the Deployment

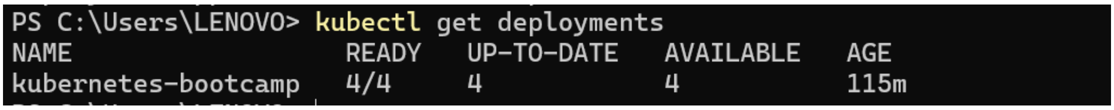


```
PS C:\Users\LENOVO> kubectl get rs
NAME                             DESIRED   CURRENT   READY   AGE
kubernetes-bootcamp-658f6cbd58   1         1         1       107m
PS C:\Users\LENOVO>
```

scale the Deployment to 4 replicas.

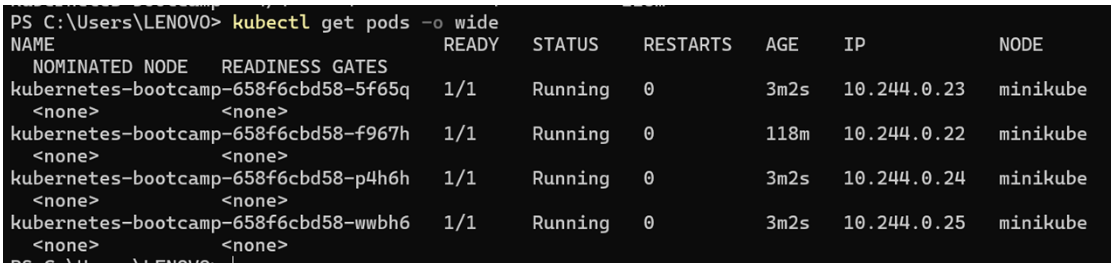


```
PS C:\Users\LENOVO> kubectl scale deployments/kubernetes-bootcamp --replicas=4
deployment.apps/kubernetes-bootcamp scaled
```

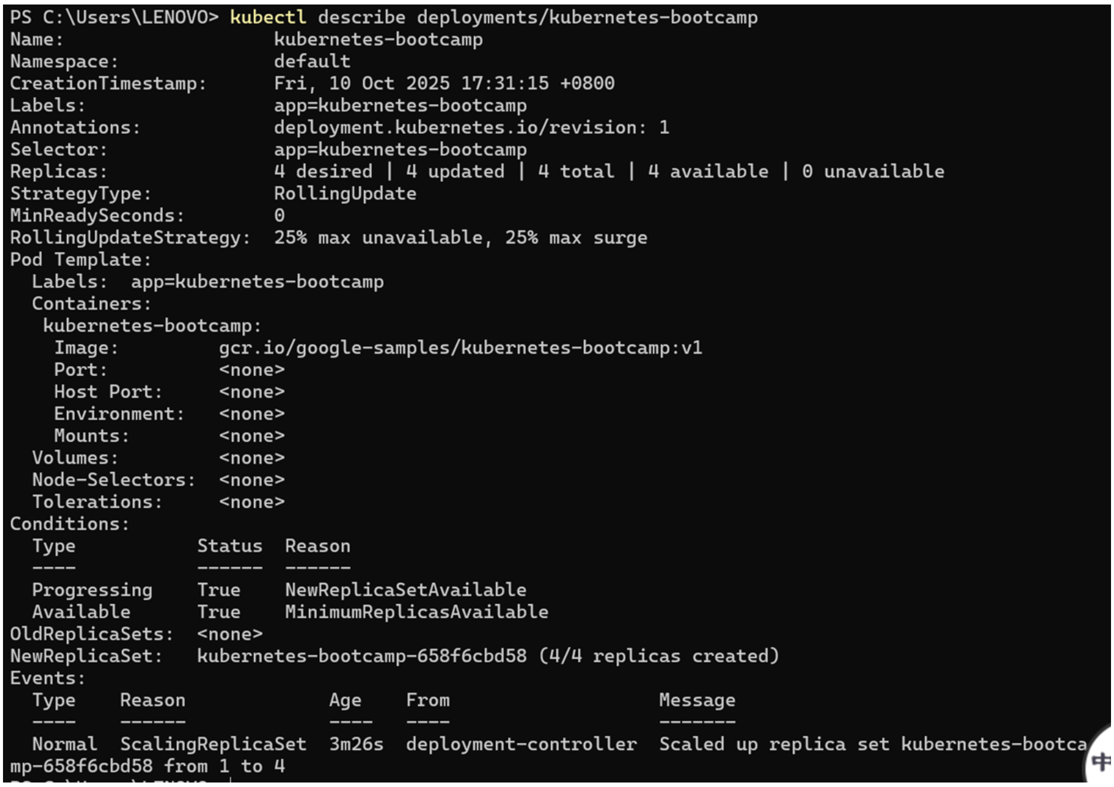


```
PS C:\Users\LENOVO> kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   4/4     4            4           115m
```

check if the number of Pods changed

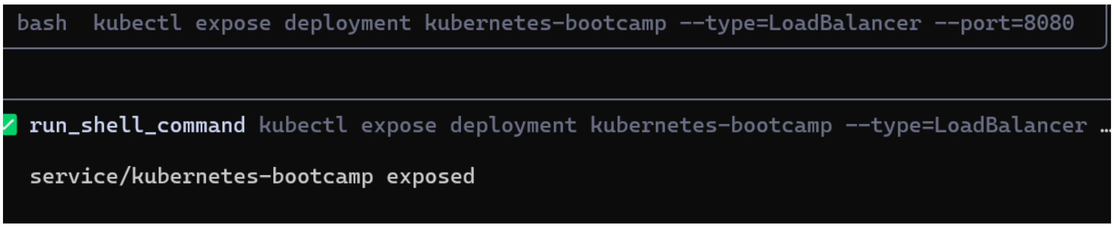


```
PS C:\Users\LENOVO> kubectl get pods -o wide
NAME                                   READY   STATUS    RESTARTS   AGE    IP            NODE
  NOMINATED NODE   READINESS GATES
kubernetes-bootcamp-658f6cbd58-5f65q   1/1     Running   0          3m2s   10.244.0.23   minikube
  <none>           <none>
kubernetes-bootcamp-658f6cbd58-f967h   1/1     Running   0          118m   10.244.0.22   minikube
  <none>           <none>
kubernetes-bootcamp-658f6cbd58-p4h6h   1/1     Running   0          3m2s   10.244.0.24   minikube
  <none>           <none>
kubernetes-bootcamp-658f6cbd58-wwbh6   1/1     Running   0          3m2s   10.244.0.25   minikube
  <none>           <none>
```

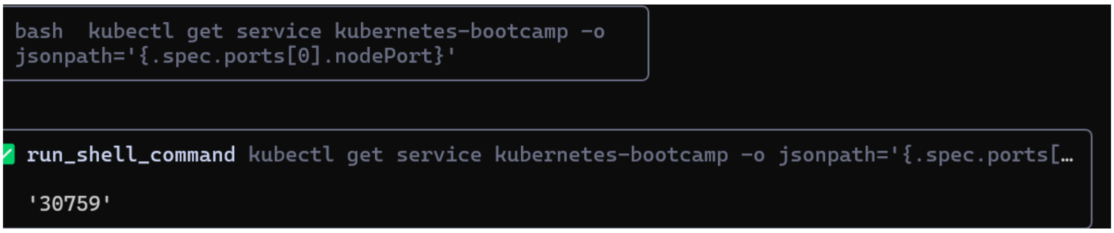


```
PS C:\Users\LENOVO> kubectl describe deployments/kubernetes-bootcamp
Name:                   kubernetes-bootcamp
Namespace:              default
CreationTimestamp:      Fri, 10 Oct 2025 17:31:15 +0800
Labels:                 app=kubernetes-bootcamp
Annotations:            deployment.kubernetes.io/revision: 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  3m26s  deployment-controller  Scaled up replica set kubernetes-bootca
mp-658f6cbd58 from 1 to 4
```

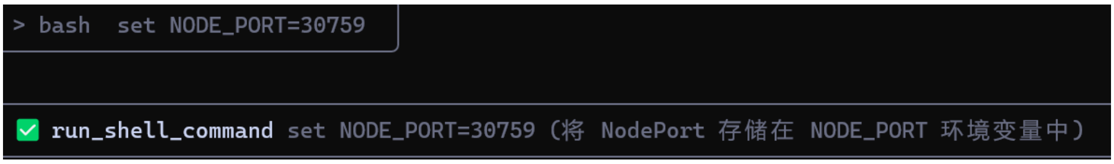
check that the Service is load-balancing the traffic.



```
bash  kubectl expose deployment kubernetes-bootcamp --type=LoadBalancer --port=8080

✅ run_shell_command kubectl expose deployment kubernetes-bootcamp --type=LoadBalancer …

  service/kubernetes-bootcamp exposed
```

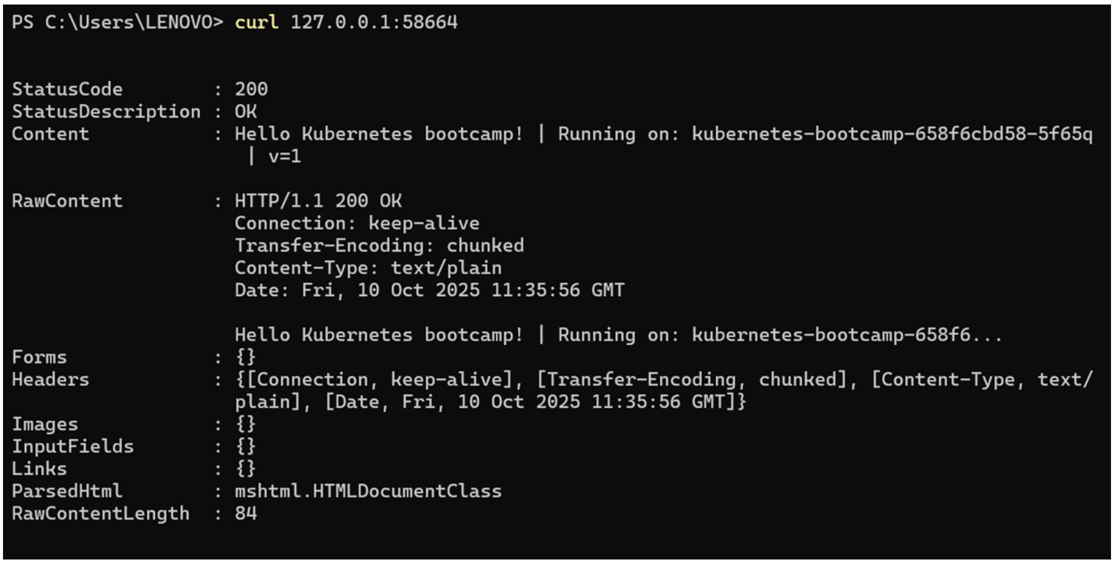


```
bash  kubectl get service kubernetes-bootcamp -o
jsonpath='{.spec.ports[0].nodePort}'

✅ run_shell_command kubectl get service kubernetes-bootcamp -o jsonpath='{.spec.ports[…

 '30759'
```



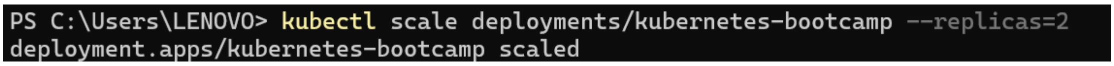
```
> bash  set NODE_PORT=30759

✅ run_shell_command set NODE_PORT=30759 (将 NodePort 存储在 NODE_PORT 环境变量中)
```

do a curl to the exposed IP address and port.



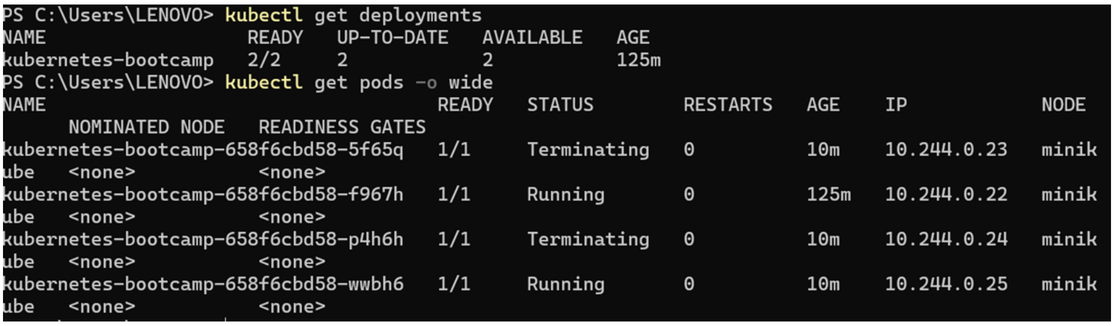
```
PS C:\Users\LENOVO> curl 127.0.0.1:58664


StatusCode        : 200
StatusDescription : OK
Content           : Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-658f6cbd58-5f65q
                     | v=1

RawContent        : HTTP/1.1 200 OK
                    Connection: keep-alive
                    Transfer-Encoding: chunked
                    Content-Type: text/plain
                    Date: Fri, 10 Oct 2025 11:35:56 GMT

                    Hello Kubernetes bootcamp! | Running on: kubernetes-bootcamp-658f6...
Forms             : {}
Headers           : {[Connection, keep-alive], [Transfer-Encoding, chunked], [Content-Type, text/
                    plain], [Date, Fri, 10 Oct 2025 11:35:56 GMT]}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 84
```

Scale Down



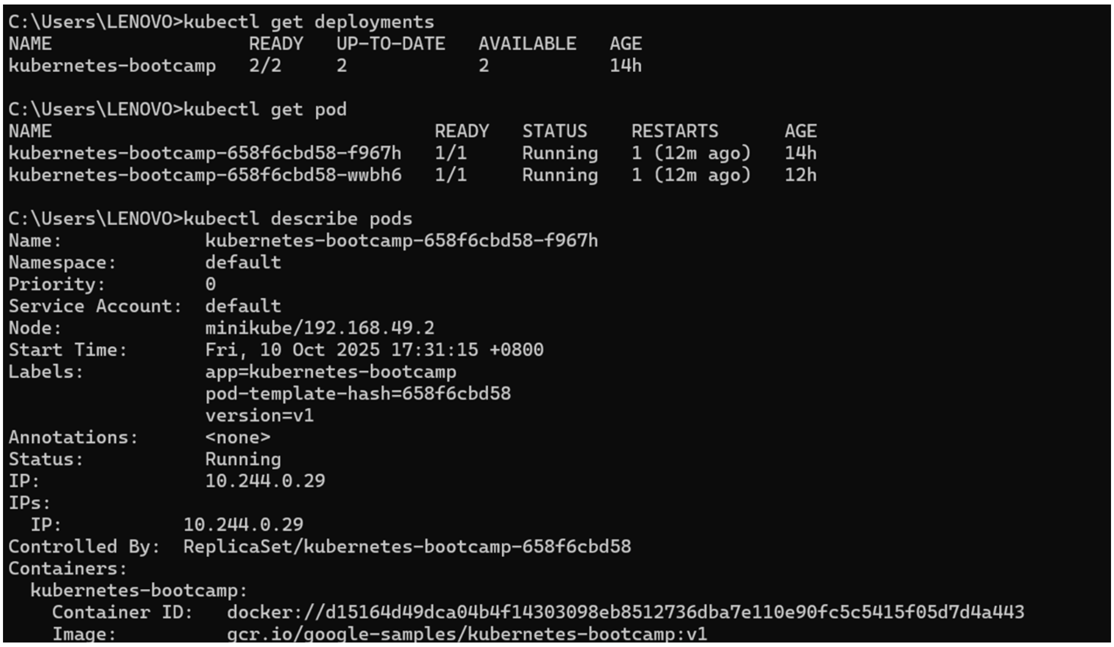
```
PS C:\Users\LENOVO> kubectl scale deployments/kubernetes-bootcamp --replicas=2
deployment.apps/kubernetes-bootcamp scaled
```

The number of replicas decreased to 2.



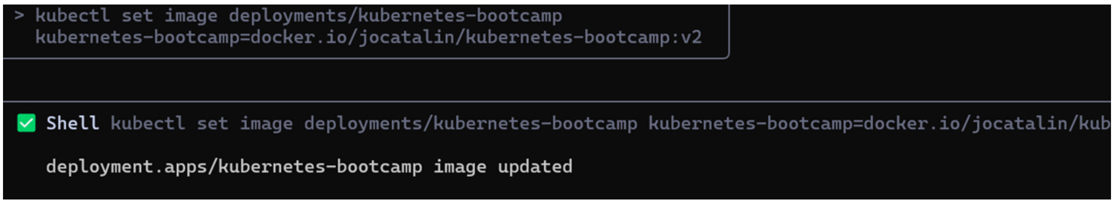
```
PS C:\Users\LENOVO> kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   2/2     2            2           125m
PS C:\Users\LENOVO> kubectl get pods -o wide
NAME                                   READY   STATUS        RESTARTS   AGE    IP            NODE
     NOMINATED NODE   READINESS GATES
kubernetes-bootcamp-658f6cbd58-5f65q   1/1     Terminating   0          10m    10.244.0.23   minik
ube    <none>           <none>
kubernetes-bootcamp-658f6cbd58-f967h   1/1     Running       0          125m   10.244.0.22   minik
ube    <none>           <none>
kubernetes-bootcamp-658f6cbd58-p4h6h   1/1     Terminating   0          10m    10.244.0.24   minik
ube    <none>           <none>
kubernetes-bootcamp-658f6cbd58-wwbh6   1/1     Running       0          10m    10.244.0.25   minik
ube    <none>           <none>
```

Update the version of the app



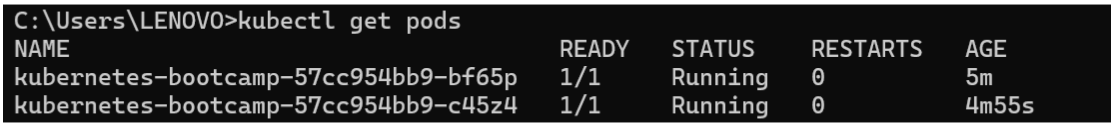
```
C:\Users\LENOVO>kubectl get deployments
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
kubernetes-bootcamp   2/2     2            2           14h

C:\Users\LENOVO>kubectl get pod
NAME                                   READY   STATUS    RESTARTS      AGE
kubernetes-bootcamp-658f6cbd58-f967h   1/1     Running   1 (12m ago)   14h
kubernetes-bootcamp-658f6cbd58-wwbh6   1/1     Running   1 (12m ago)   12h

C:\Users\LENOVO>kubectl describe pods
Name:             kubernetes-bootcamp-658f6cbd58-f967h
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Fri, 10 Oct 2025 17:31:15 +0800
Labels:           app=kubernetes-bootcamp
                  pod-template-hash=658f6cbd58
                  version=v1
Annotations:      <none>
Status:           Running
IP:               10.244.0.29
IPs:
  IP:           10.244.0.29
Controlled By:  ReplicaSet/kubernetes-bootcamp-658f6cbd58
Containers:
  kubernetes-bootcamp:
    Container ID:   docker://d15164d49dca04b4f14303098eb8512736dba7e110e90fc5c5415f05d7d4a443
    Image:          gcr.io/google-samples/kubernetes-bootcamp:v1
```

update the image of the application to version 2



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

✓ Shell kubectl set image deployments/kubernetes-bootcamp kubernetes-bootcamp=docker.io/jocatalin/kub

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

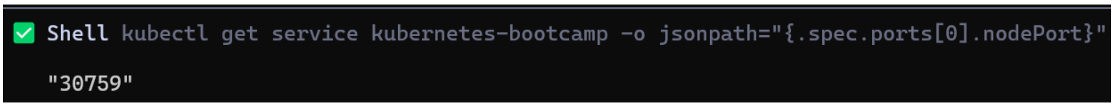
Check the status of the new Pods, and view the old one terminating with the get pods subcommand:



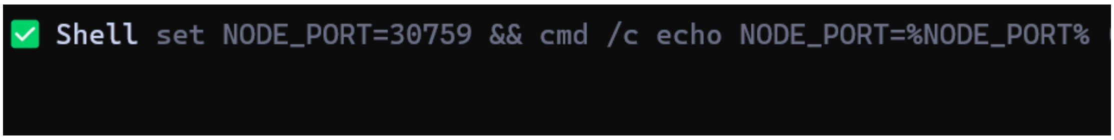
```
C:\Users\LENOVO>kubectl get pods
NAME                                   READY   STATUS    RESTARTS   AGE
kubernetes-bootcamp-57cc954bb9-bf65p   1/1     Running   0          5m
kubernetes-bootcamp-57cc954bb9-c45z4   1/1     Running   0          4m55s
```

Verify an update

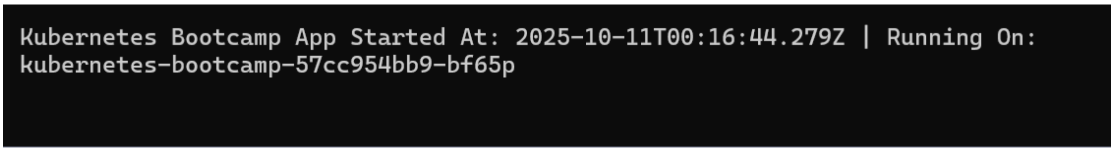
Create an environment variable called NODE_PORT that has the value of the Node port assigned:



```
✓ Shell kubectl get service kubernetes-bootcamp -o jsonpath="{.spec.ports[0].nodePort}"

  "30759"
```

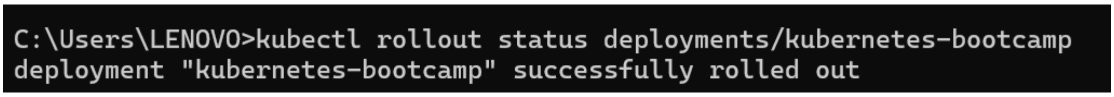


```
✓ Shell set NODE_PORT=30759 && cmd /c echo NODE_PORT=%NODE_PORT%
```



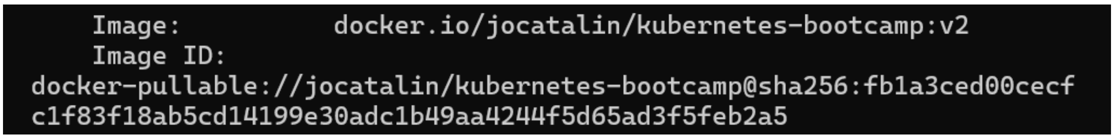
```
Kubernetes Bootcamp App Started At: 2025-10-11T00:16:44.279Z | Running On:
kubernetes-bootcamp-57cc954bb9-bf65p
```

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



```
C:\Users\LENOVO>kubectl rollout status deployments/kubernetes-bootcamp
deployment "kubernetes-bootcamp" successfully rolled out
```

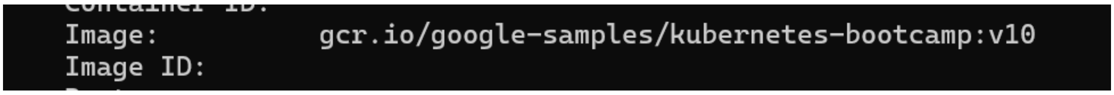
To view the current image version of the app, run the describe pods subcommand:



```
    Image:          docker.io/jocatalin/kubernetes-bootcamp:v2
    Image ID:
 docker-pullable://jocatalin/kubernetes-bootcamp@sha256:fb1a3ced00cecf
 c1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
```

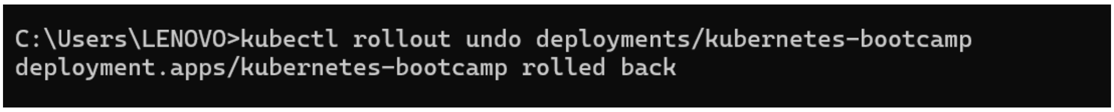
Roll back an update

deploy an image tagged with v10:



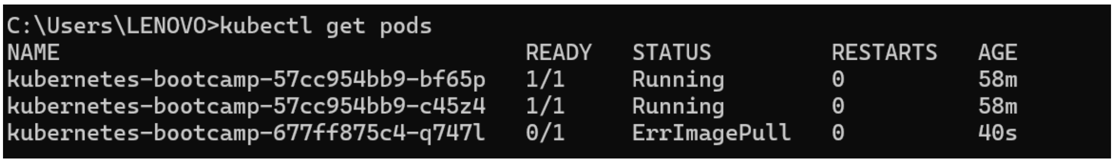
```
    Container ID:
    Image:          gcr.io/google-samples/kubernetes-bootcamp:v10
    Image ID:
```

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



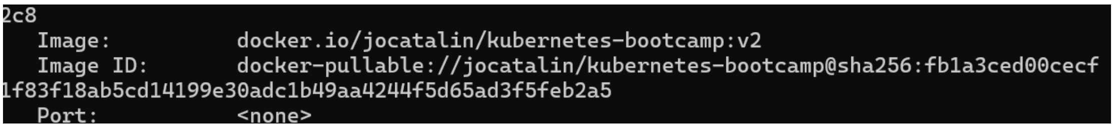
```
C:\Users\LENOVO>kubectl rollout undo deployments/kubernetes-bootcamp
deployment.apps/kubernetes-bootcamp rolled back
```

Use the get pods subcommand to list the Pods again:



```
C:\Users\LENOVO>kubectl get pods
NAME                                   READY   STATUS         RESTARTS   AGE
kubernetes-bootcamp-57cc954bb9-bf65p   1/1     Running        0          58m
kubernetes-bootcamp-57cc954bb9-c45z4   1/1     Running        0          58m
kubernetes-bootcamp-677ff875c4-q747l   0/1     ErrImagePull   0          40s
```

check the image deployed on the running Pods



```
2c8
    Image:          docker.io/jocatalin/kubernetes-bootcamp:v2
    Image ID:       docker-pullable://jocatalin/kubernetes-bootcamp@sha256:fb1a3ced00cecf
1f83f18ab5cd14199e30adc1b49aa4244f5d65ad3f5feb2a5
    Port:           <none>
```