Content

[**Install Minikube** 2](#_Toc218026317)

[**Hello Minikube** 4](#_Toc218026318)

[**Using kubectl to Create a Deployment** 9](#_Toc218026319)

[**Viewing Pods and Nodes** 11](#_Toc218026320)

[**Using a Service to Expose Your App** 13](#_Toc218026321)

[**Running Multiple Instances of Your App** 17](#_Toc218026322)

[**Performing a Rolling Update** 20](#_Toc218026323)

# **Install Minikube**

1. Install minikube.exe according to this website <https://minikube.cn/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download>.
2. Add minikube.exe binary file to my PATH.

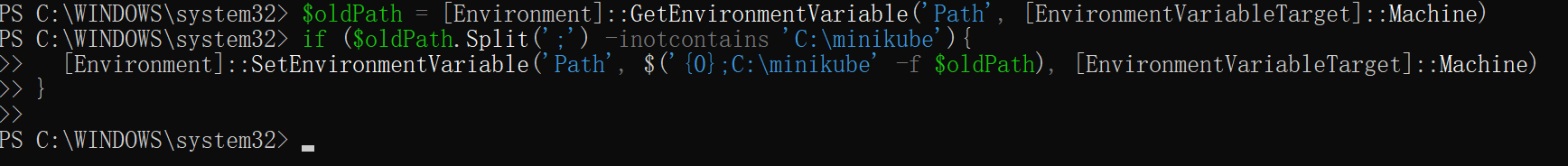
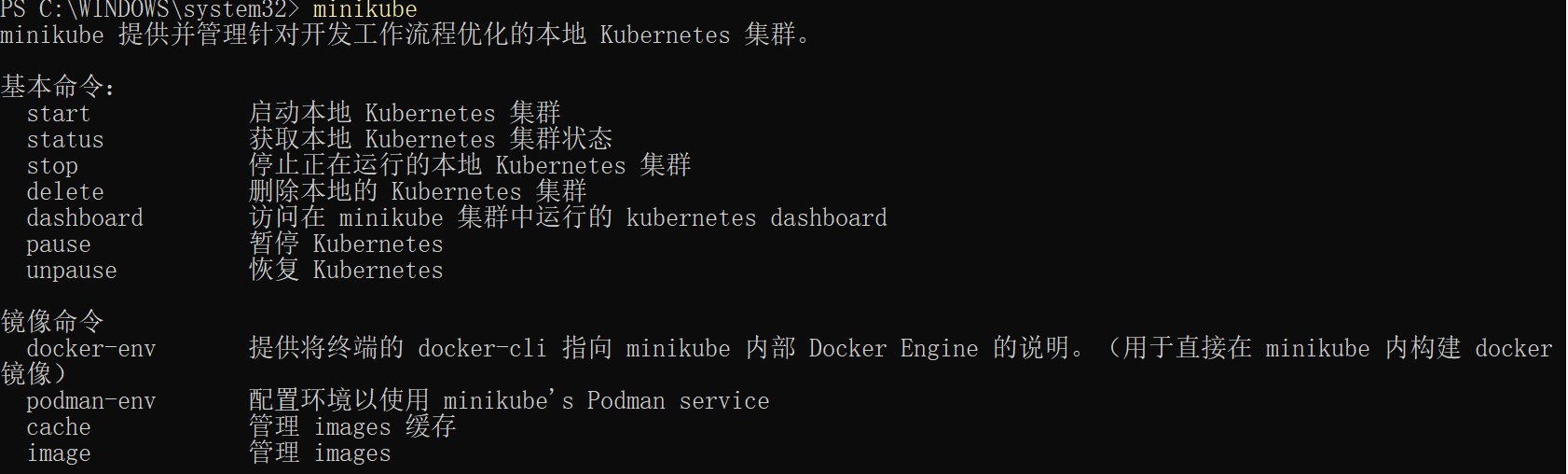
Command:

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

if ($oldPath.Split(';') -inotcontains 'C:\minikube'){

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

}

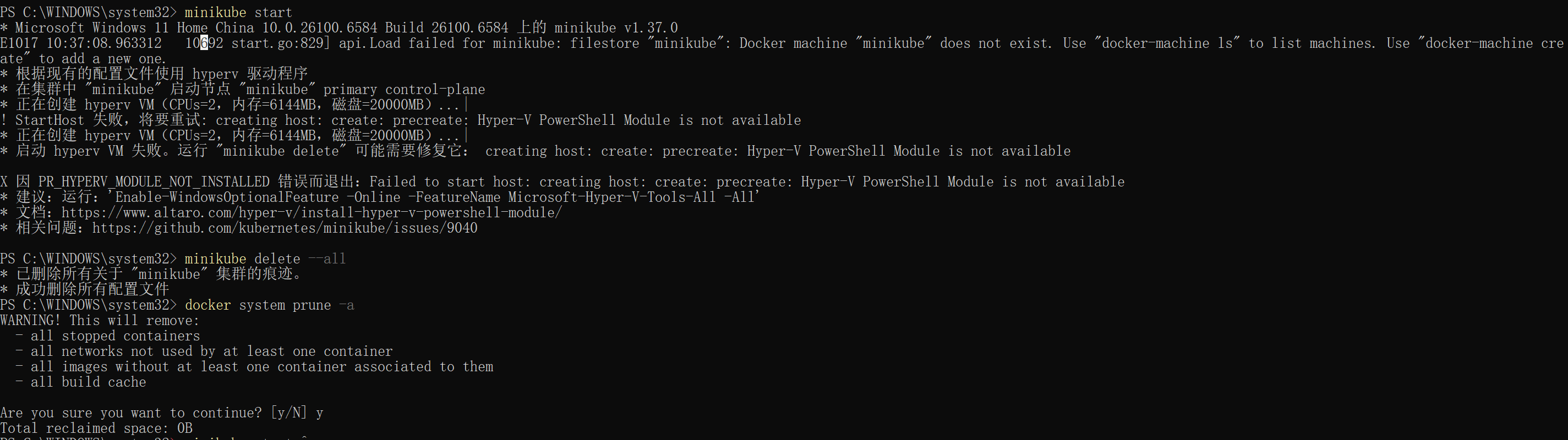


Note: When running PowerShell as administrator, C:\minukube cannot be found in the terminal.

1. Create a minikube cluster

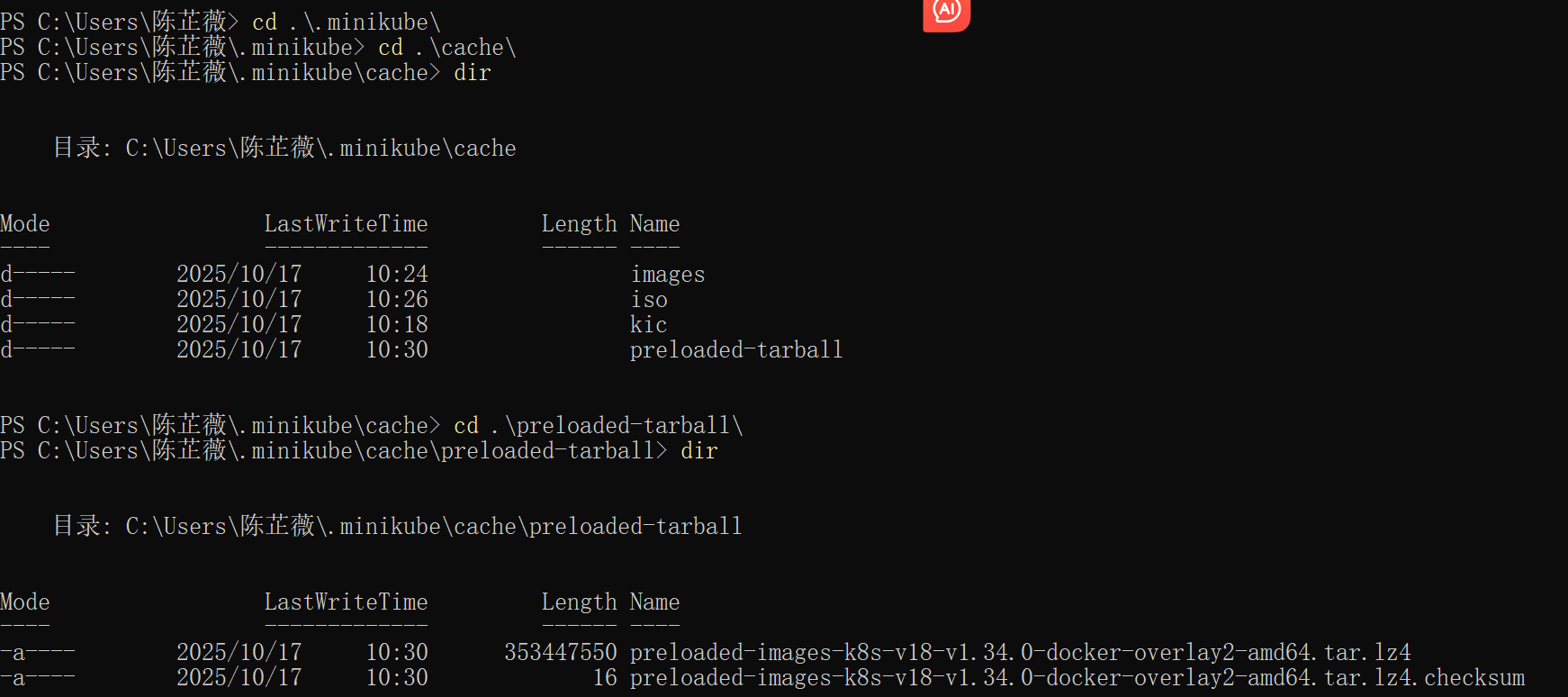
**Command:** *minikube start*

Problems that I met as the following photo



**Solutions:**

First, *enter minikube delete* and *docker system prune -a* to delete all traces of the "minikube" cluster and all configuration files. Then enter *minikube start --driver=docker --image-repository=registry.cn-hangzhou.aliyuncs.com/google\_containers*, which it is explicitly specified as Docker and a domestic mirror site is used.



Seeing the length of 353447550 in the cache explains why it took so long to download.

# **Hello Minikube**

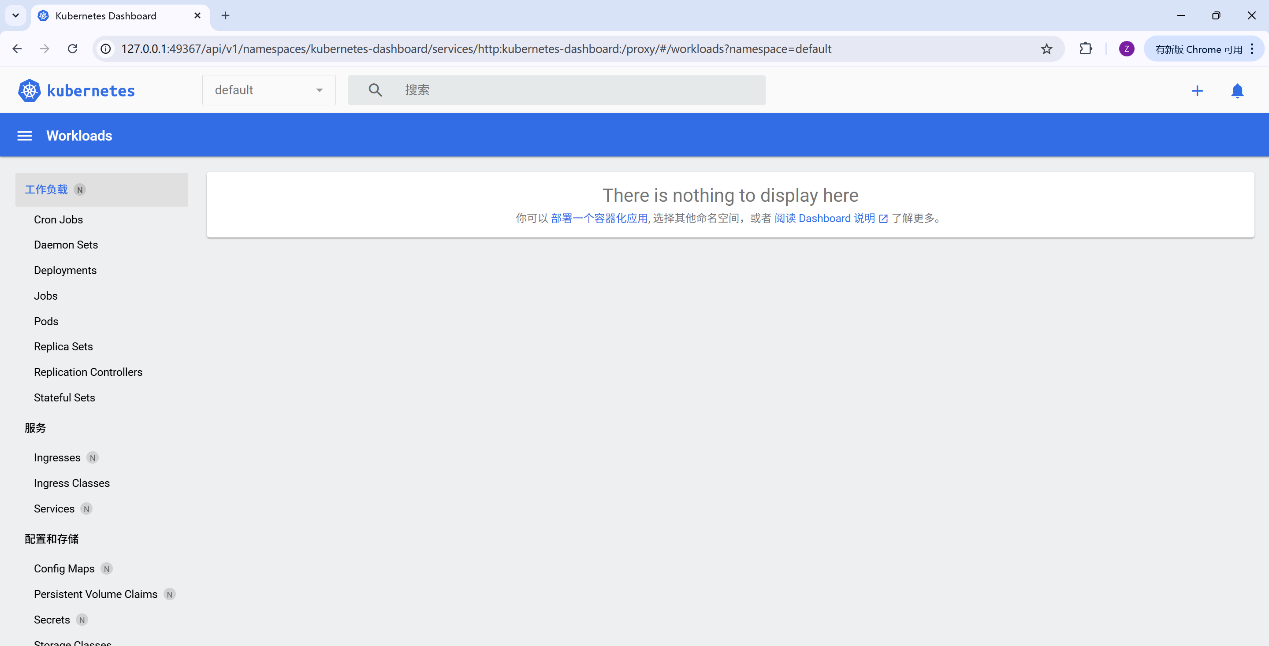
1. Open the Dashboard

**Command**: *minikube dashboard*

The *dashboard* command enables the dashboard add-on and opens the proxy in the default web browser. We can create Kubernetes resources on the dashboard such as Deployment and Service.

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。



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

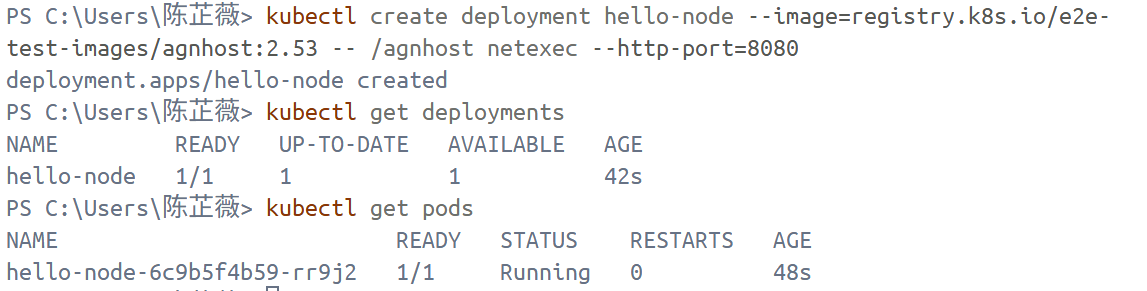
*kubectl create deployment hello-node --image=registry.k8s.io/e2e-test-images/agnhost:2.53 -- /agnhost netexec --http-port=8080*

* 1. View the deployment

*kubectl get deployments*

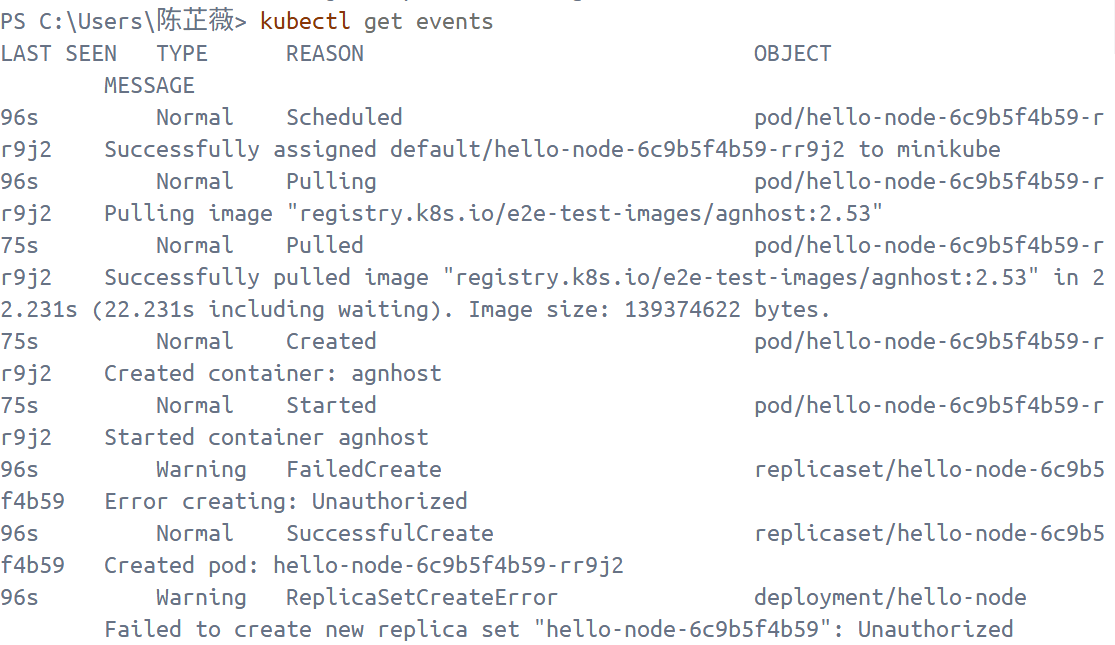
* 1. View the pod

*kubectl get pods*



* 1. View cluster events:

*kubectl get events*



* 1. View the kubectl configuration

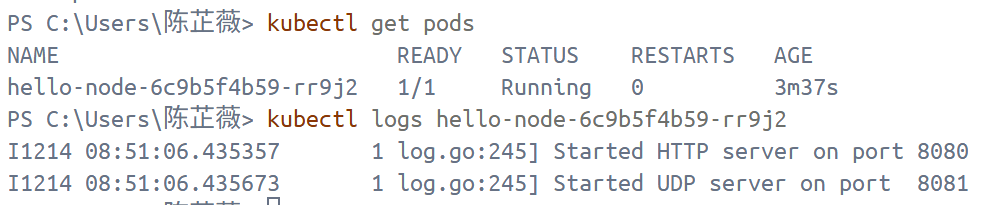
*kubectl config view*

图片包含 文本

AI 生成的内容可能不正确。

* 1. View application logs for a container in a pod

Here I used *kubectl get pods* to get the pod name as hello-node-6c9bf4b59-rr9j2. And then enter kubectl logs hello-node-6c9bf4b59-rr9j2 to view application logs.



1. Create a service
   1. Expose the Pod to the public internet

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

* 1. View the Service

*kubectl get services*

From the result, it shows that on cloud providers that support load balancers, an external IP address would be provisioned to access the Service. On minikube, the LoadBalancer type makes the Service accessible through the *minikube service* command.

* 1. Open up a browser window that serves my app and shows the app's response.

*minikube service hello-node*

日程表

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

1. Enable addons
   1. List the currently supported addons

*minikube addons list*

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

* 1. Enable an “metrics-server” addon

*minikube addons enable metrics-server*

文本

AI 生成的内容可能不正确。

* 1. View the Pod and Service I created by installing that addon

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

From the picture, it can find the status of “metrics server” is wrong. So entere the command “*kubectl describe pod*” to check its problems.



*kubectl describe pod metrics-server-74576d8779-8kxcg -n kube-system*



文本

AI 生成的内容可能不正确。

**Cause:**

it's not a network timeout, but rather the image reference syntax leading to a "manifest not found" error. The `manifest for .../metrics-server@sha256:8925... not found: manifest unknown` indicates that the current image is written as a "tag + digest": `registry.cn-hangzhou.aliyuncs.com/google\_containers/metrics-server:v0.8.0@sha256:8925...`. Many image acceleration/synchronization repositories (including your Alibaba Cloud path) may not retain or support the manifest corresponding to this digest. As a result, although `v0.8.0` may exist, the precise fingerprint `@sha256:...` cannot be found in that repository ⇒ pull will inevitably fail ⇒ `ImagePullBackOff` ⇒ `kubectl top` will also show no metrics.

**Solution**:

*kubectl set image deploy/metrics-server -n kube-system metrics-server=registry.vn-hangzhou.aliyuncs.com/google\_containers/metrics-server:v0.8.0*

*kubectl rollout restart deploy/metrics-server -n kube-system deployment.apps/metrics-server restarted*

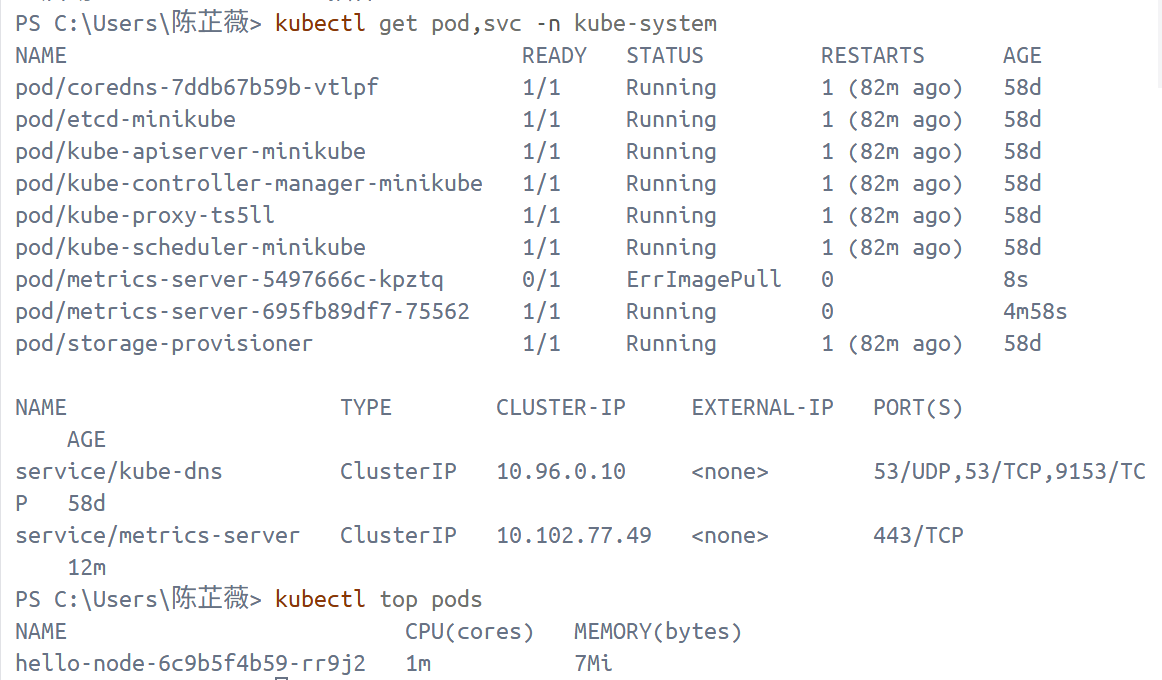
*kubectl get pod -n kube-system | findstr metrics-server*

*kubectl get apiservice v1beta1.metrics.k8s.io*

文本

AI 生成的内容可能不正确。

Final result: new metrics-server is running and the command *kubectl top pods* returns its name, CPU and momory rather than “error: Metrics API not available”.



* 1. Disable metrics-server

*minikube addons disable metrics-server*

1. Cleanup
   1. Clean up the resources I created in my cluster

*kubectl delete service hello-node*

*kubectl delete deployment hello-node*

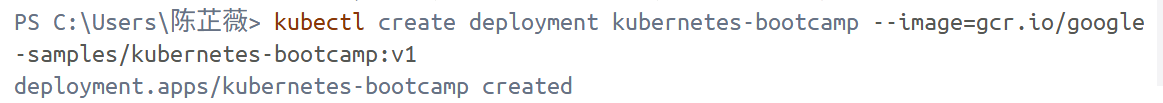
* 1. Stop the Minikube cluster

*minikube stop*



# **Using kubectl to Create a Deployment**

1. Deploy an app
   1. Use *kubectl create deployment*command to deploy the first app with the provided deployment name and app image location.



* 1. List deployments use the *kubectl get deployments* command

图形用户界面, 文本

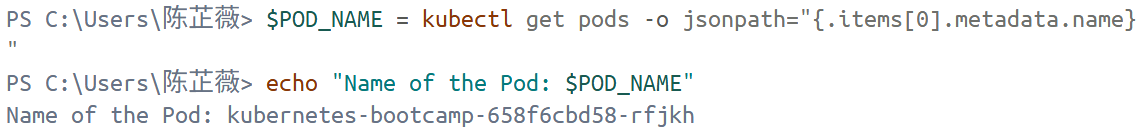
AI 生成的内容可能不正确。

1. View the app

Open a second terminal window to run the proxy with the command *kubectl proxy*. Then query the version directly through the API using the *curl* command



First, get the Pod name and store it in the environment variable POD\_NAME. Then access the Pod through the proxied API



Problem：due to how Windows PowerShell parses commands and variables

Reason 1. In PowerShell:

* curl is actually an alias for Invoke-WebRequest
* It doesn't handle $POD\_NAME like bash does
* As a result, $POD\_NAME:8080/proxy is treated as a strange string

The Kubernetes API mistakenly thinks you are accessing /pods/proxy, so it returns "pods "proxy" not found".

Reason 2:

Variable concatenation is incorrect in PowerShell. In PowerShell, a colon (:) immediately following a variable will cause parsing failure.

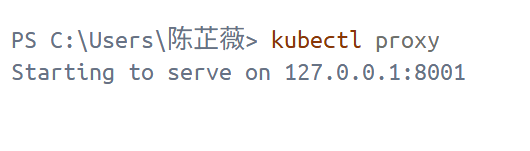
Reason 3:

kubectl proxy should run in the same terminal, such as PowerShell, not one PowerShell and one CMD.

文本

AI 生成的内容可能不正确。

Result:

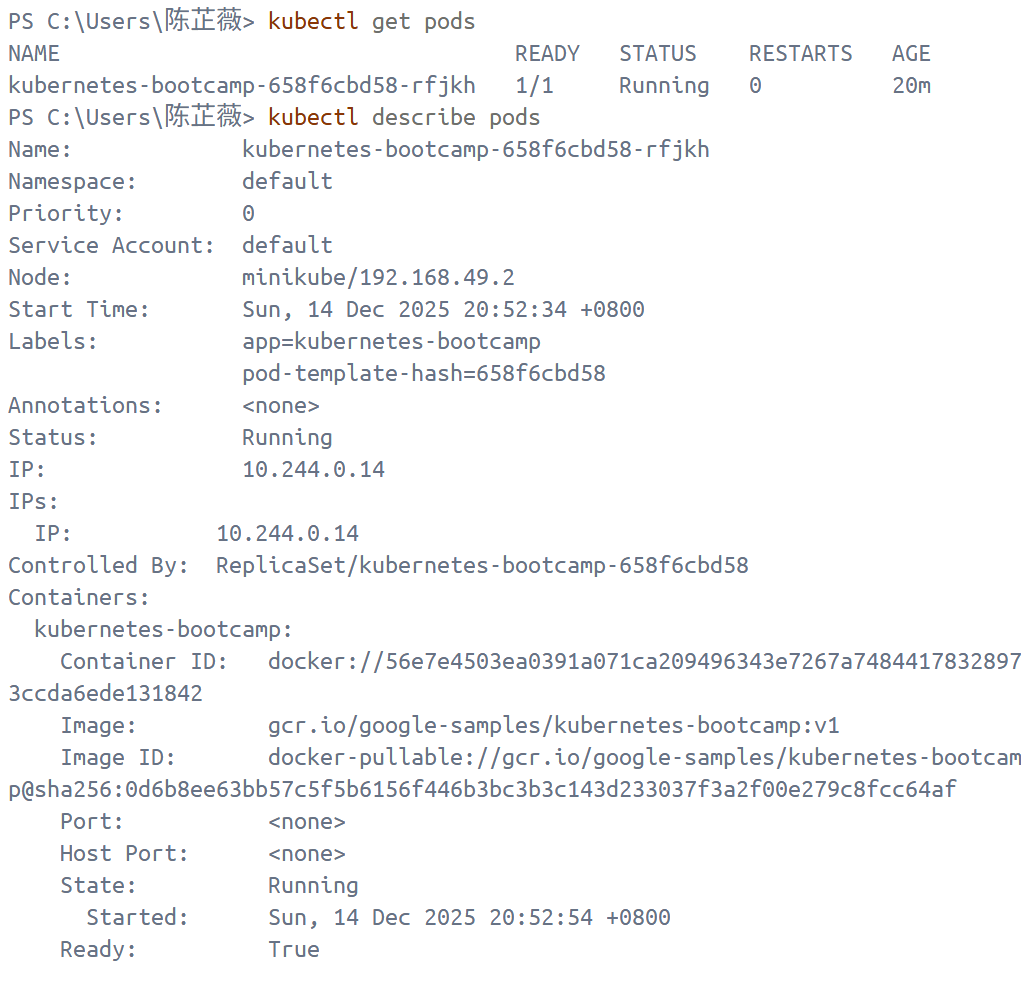




# **Viewing Pods and Nodes**

1. Check application configuration
   1. Use the *kubectl get* command and look for existing Pods
   2. View what containers are inside that Pod and what images are used to build those containers

*kubectl describe pods*



1. Show app in the terminal
   1. Use the *kubectl proxy* command to run a proxy in a second terminal
   2. Get the Pod name and query that pod directly through the proxy.

文本

AI 生成的内容可能不正确。

* 1. See the output of our application by running curl request

Problem: PowerShell's curl (actually Invoke-WebRequest) defaults to "N/No" in the security prompt, so it cancels the request directly.

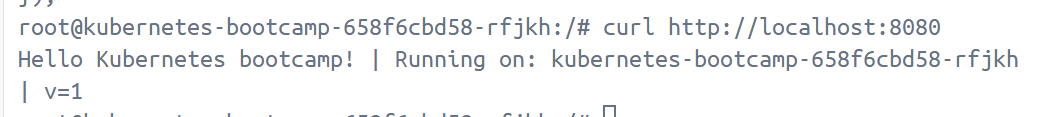


**Solution**:

文本

AI 生成的内容可能不正确。





# **Using a Service to Expose Your App**

1. Create a new service
   1. Verify that our application is running by the *kubectl get* command to look for existing Pods
   2. List the current Services from our cluster

kubectl get services

* 1. Expose the deployment to external traffic

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

* 1. Find out what port was opened externally

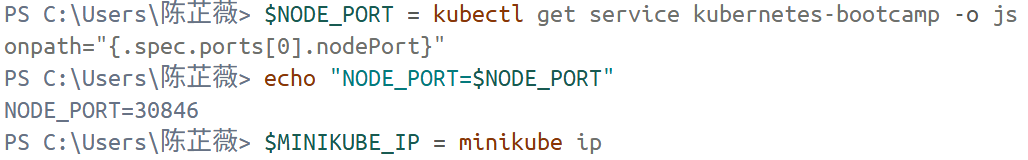
kubectl describe services/kubernetes-bootcamp



* 1. Create an environment variable called NODE\_PORT that has the value of the Node port assigned

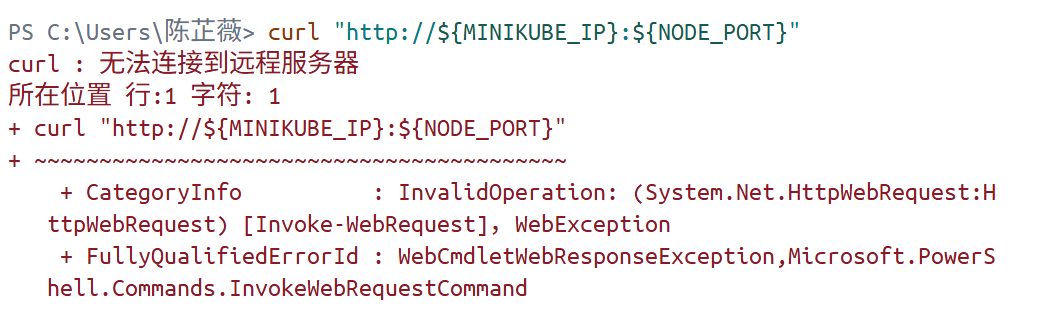
*$NODE\_PORT = kubectl get service Kubernetes-bootcamp -o js onpath=”.spec.ports[0].nodePort)”*

*$MINIKUBE\_IP = minikube ip*



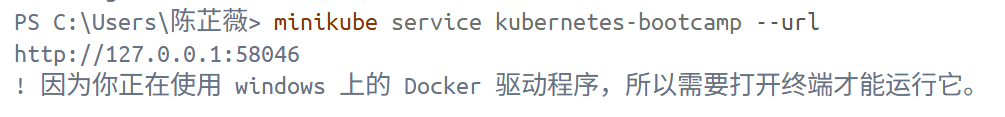
* 1. Test that the app is exposed outside of the cluster using *curl*

*curl http://"****$(****MINIKUBE\_IP****)****:$NODE\_PORT"*



This is because containers inside Docker Desktop are isolated from my host computer. So if I am running minikube with Docker Desktop as the container driver, a minikube tunnel is needed.

Open a separate terminal window, execute *minikube service kubernetes-bootcamp –url.*



Then use the given URL to access the app.

curl 127.0.0.1:58046

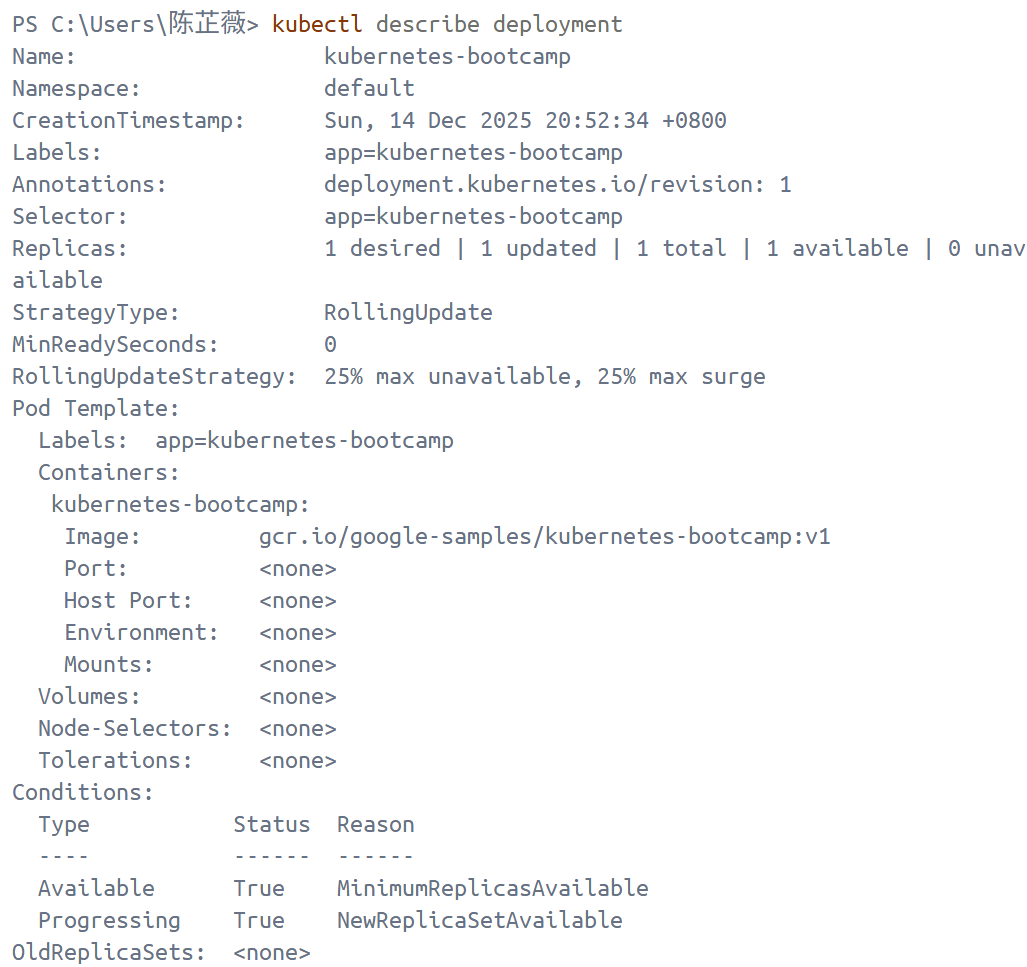
文本

AI 生成的内容可能不正确。

Analysis: When using the Minikube Docker driver on Windows, NodePort may not be directly accessible from the host via `minikube ip:nodePort` because this IP belongs to the Docker internal network. In this case, you should use `minikube service <svc> --url` to obtain the access address mapped to `127.0.0.1:<port>`, or use `kubectl port-forward` to forward the service to the local port. Accessing `http://127.0.0.1:<port>` returns a 200 OK response, proving that the application has been successfully exposed to the outside of the cluster (host side).

1. Using labels
   1. See the name (the *key*) of that label that created by Deployment for our pod

*kubectl describe deployment*



* 1. Use this label to query our list of Pods

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

* 1. List the existing Services

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

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

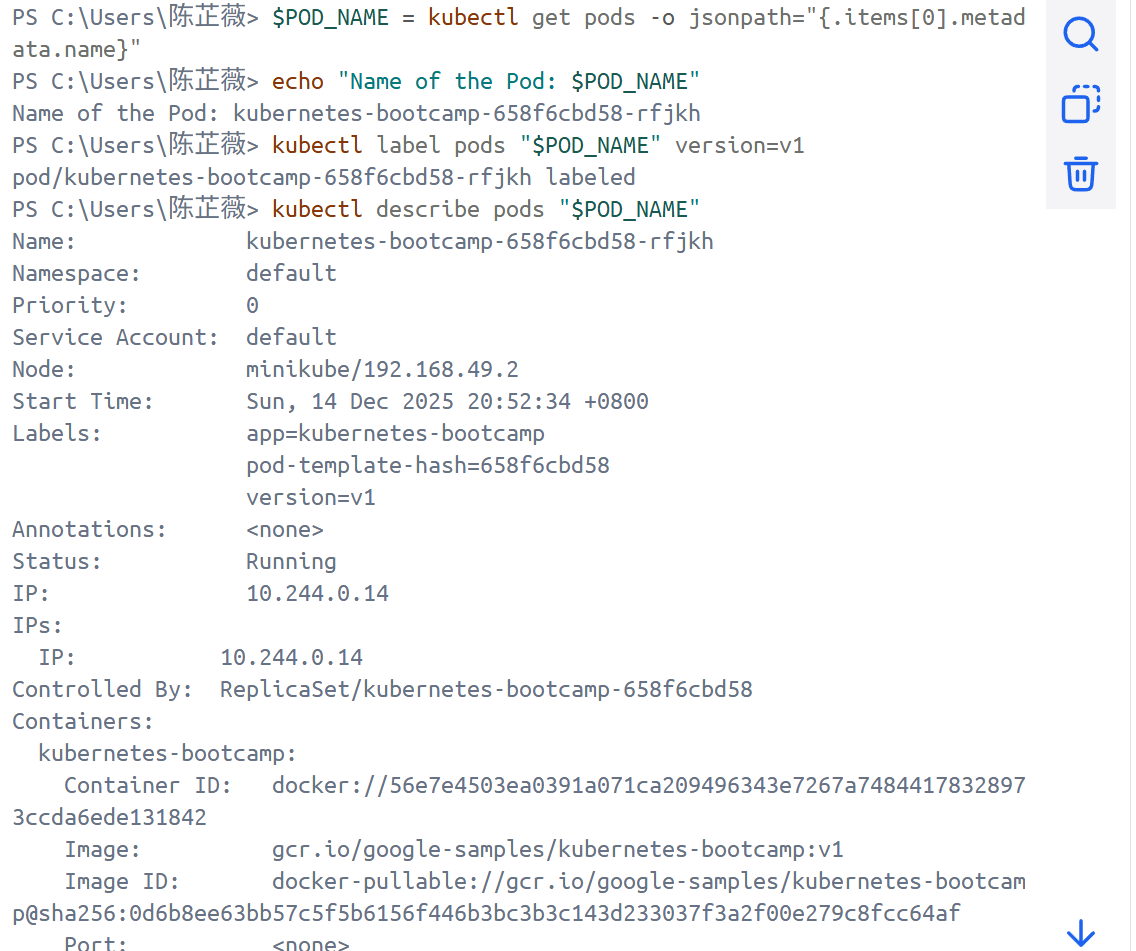
* 1. Get the name of the Pod and store it in the POD\_NAME environment variable

*$POD\_NAME = kubectl get pods -o jsonpath=”{.items[0].metadata.name}”*

* 1. Use the label subcommand followed by the object type to apply a new label

*kubectl label pods "$POD\_NAME" version=v1*

* 1. Check it with the *describe pod* command

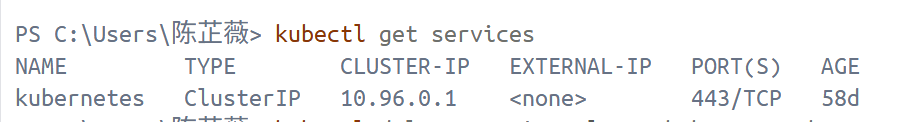


1. Delete a service
   1. delete Services

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

* 1. Confirm that the Service is gone

*kubectl get services*



* 1. Use curl the previously exposed IP and port to confirm that route is not exposed anymore

*curl 127.0.0.1:58046*

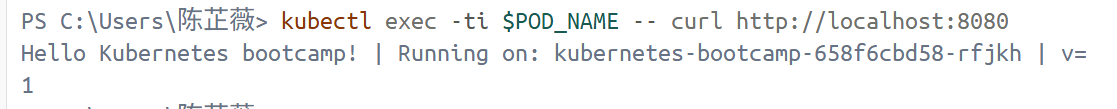
图形用户界面, 文本

AI 生成的内容可能不正确。

* 1. Confirm that the app is still running with a curl from inside the pod

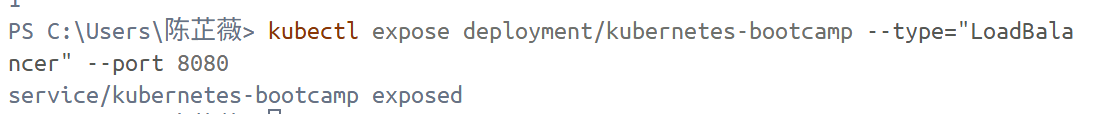
kubectl exec -ti $POD\_NAME -- curl <http://localhost:8080>

Here that the application is up. This is because the Deployment is managing the application. To shut down the application, the Deployment is needed to be deleted.



# **Running Multiple Instances of Your App**

1. Create a new Service with its type set to LoadBalancer

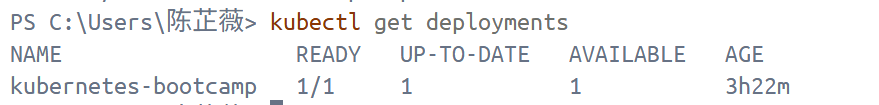


1. Scaling a Deployment
   1. List my Deployments

*kubectl get deployments*

This shows:

* *NAME* lists the names of the Deployments in the cluster.
* *READY* shows the ratio of CURRENT/DESIRED replicas
* *UP-TO-DATE* displays the number of replicas that have been updated to achieve the desired state.
* *AVAILABLE* displays how many replicas of the application are available to your users.
* *AGE* displays the amount of time that the application has been running.



* 1. See the ReplicaSet created by the Deployment

*kubectl get rs*

Notice that the name of the ReplicaSet is always formatted as [DEPLOYMENT-NAME]-[RANDOM-STRING]. The random string is randomly generated and uses the pod-template-hash as a seed. DESIRED displays the desired number of replicas of the application, which you define when you create the Deployment. This is the desired state. *CURRENT* displays how many replicas are currently running.

* 1. Scale the Deployment to 4 replicas

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

* 1. List the Deployment

*kubectl get deployments*

* 1. Check if the number of Pods changed

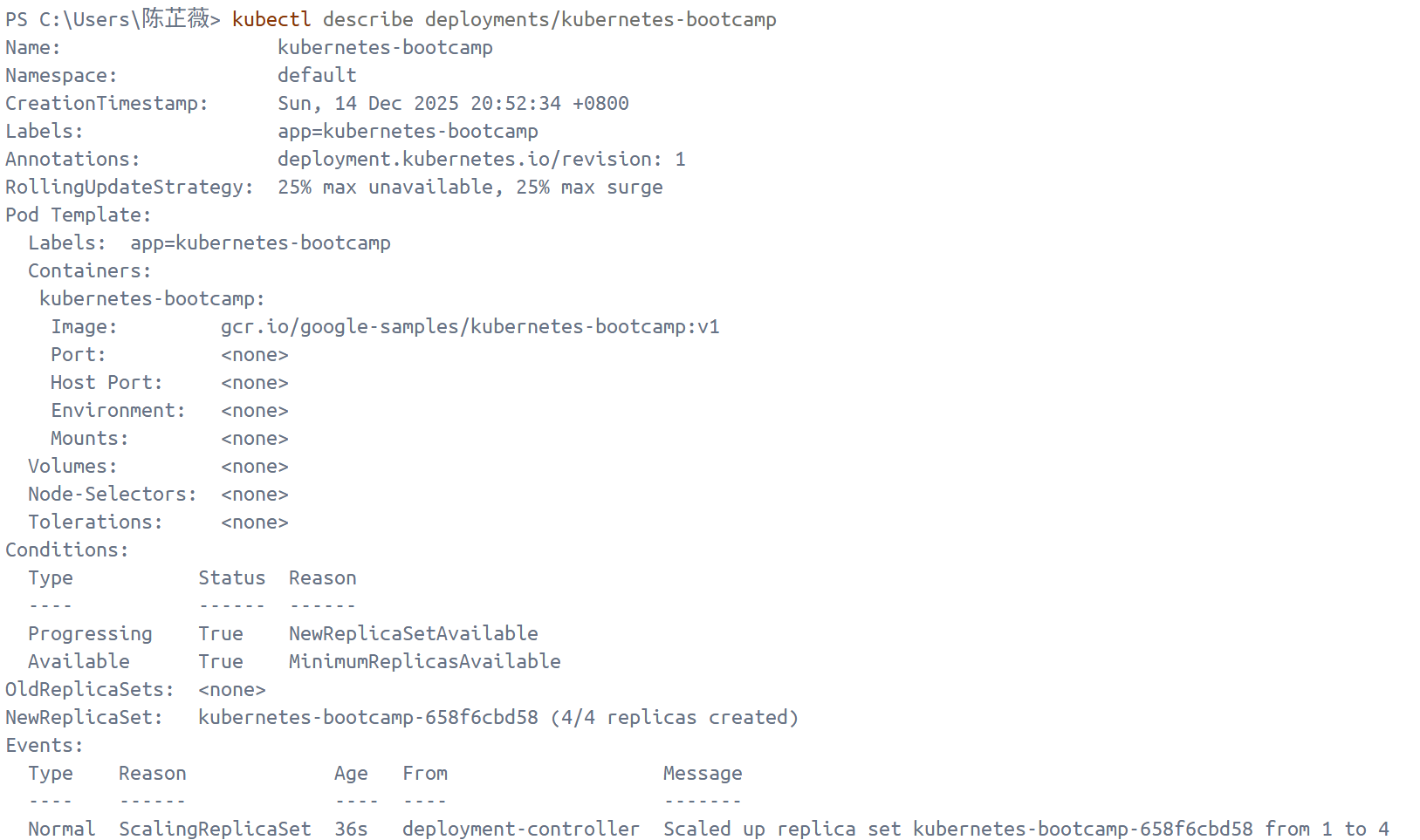
*kubectl get pods -o wide*

文本, 表格

AI 生成的内容可能不正确。

* 1. Check the changes of Pods in the Deployment events log

*kubectl describe deployments/kubernetes-bootcamp*



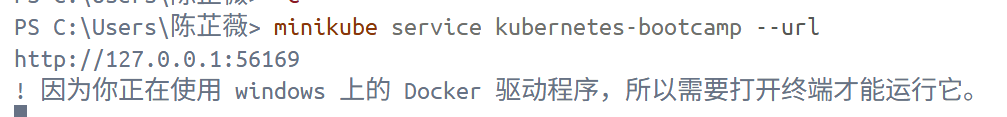
1. Load balancing
   1. Use *describe service* to find out the exposed IP and Port

图形用户界面, 文本

AI 生成的内容可能不正确。

* 1. Open a minikube tunnel in a separate terminal window

*minikube service kubernetes-bootcamp --url*



* 1. Curl to the exposed IP address and port.

*curl 127.0.0.1:51082*



1. Scale down
   1. Scale down the Deployment to 2 replicas

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

* 1. List the Deployments to check if the change was applied with the *get deployments* subcommand
  2. List the number of Pods, with*get pods*

文本

AI 生成的内容可能不正确。

# **Performing a Rolling Update**

* 1. Update the version of the app
  2. List my Deployments

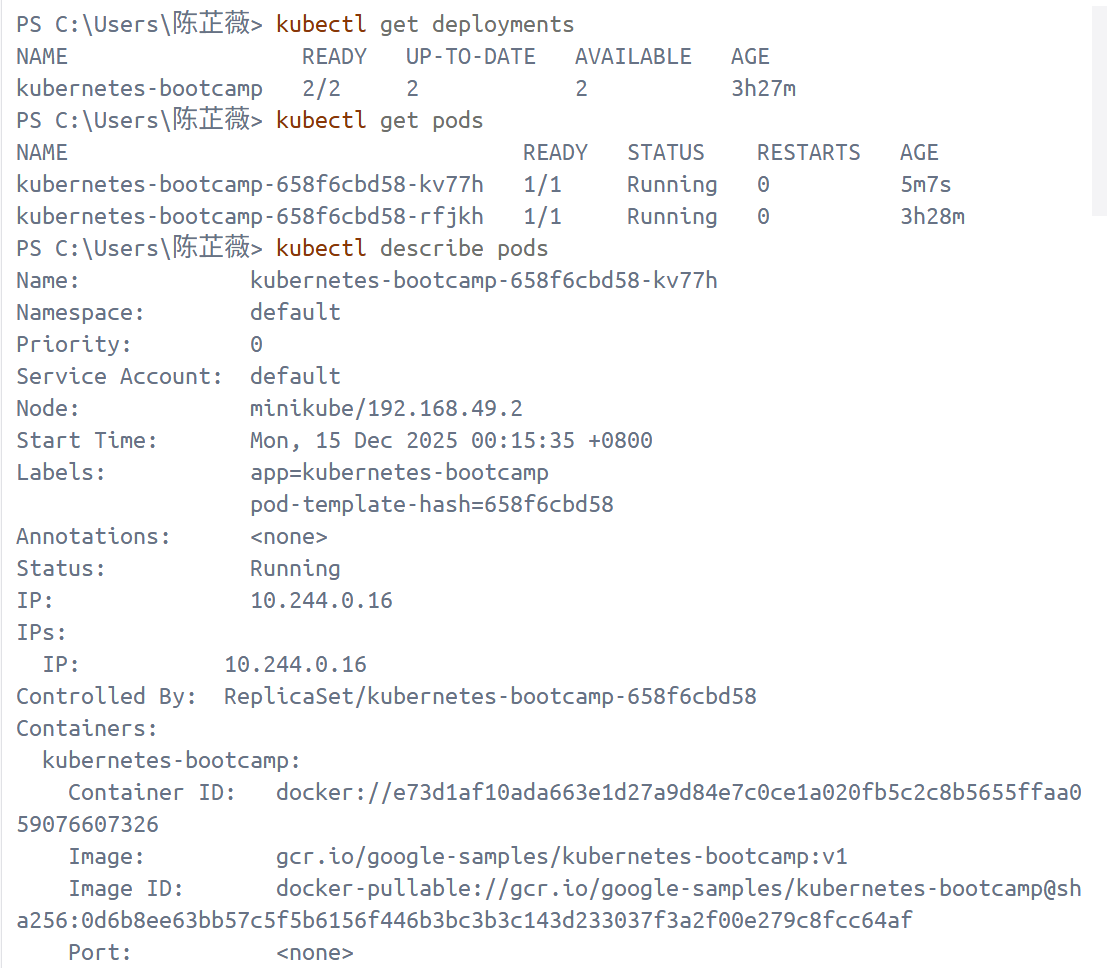
*kubectl get deployments*

* 1. List the running Pods

*kubectl get pods*

* 1. View the current image version of the app

*kubectl describe pods*



* 1. Update the image of the application to version 2

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

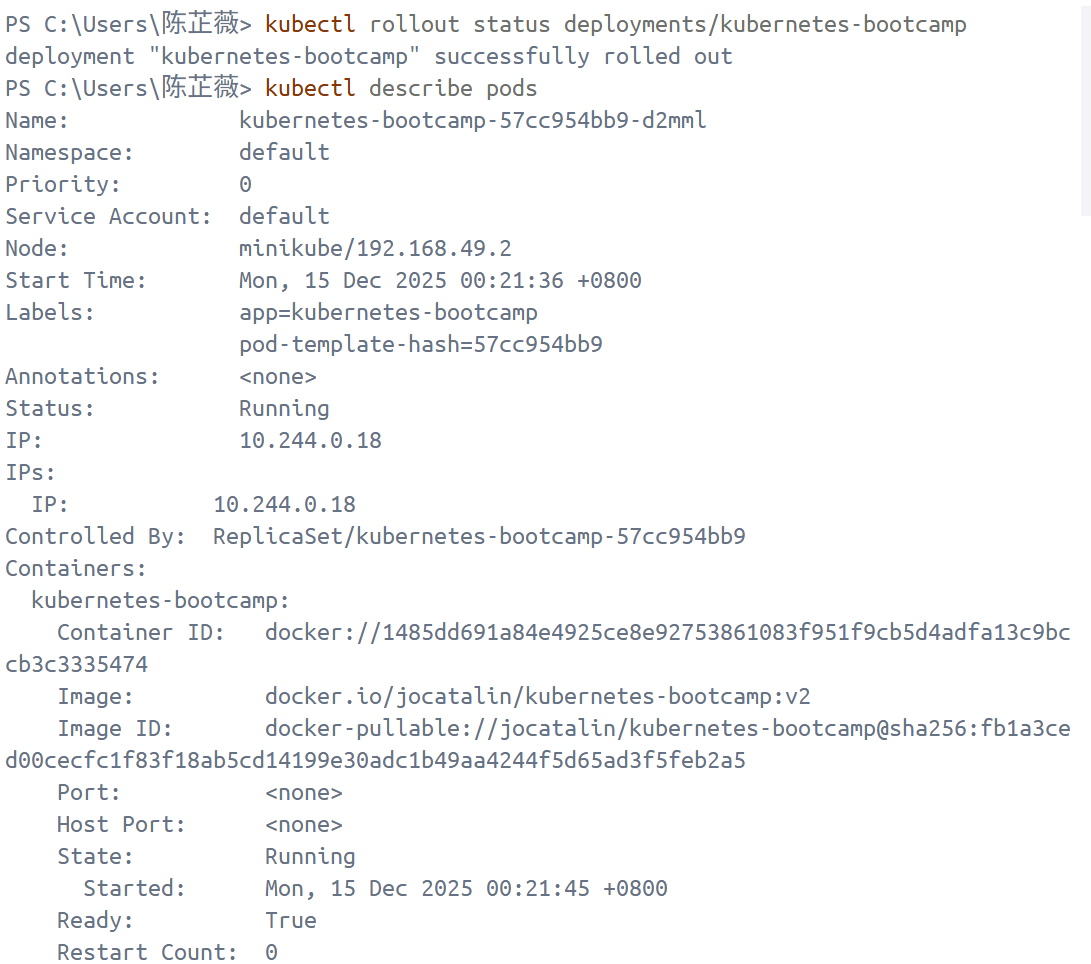
* 1. Check the status of the new Pods, and view the old one terminating with the *g*et pods subcommand

表格

AI 生成的内容可能不正确。

* 1. verify an update
  2. Confirm the update by running the *rollout status* subcommand
  3. View the current image version of the app

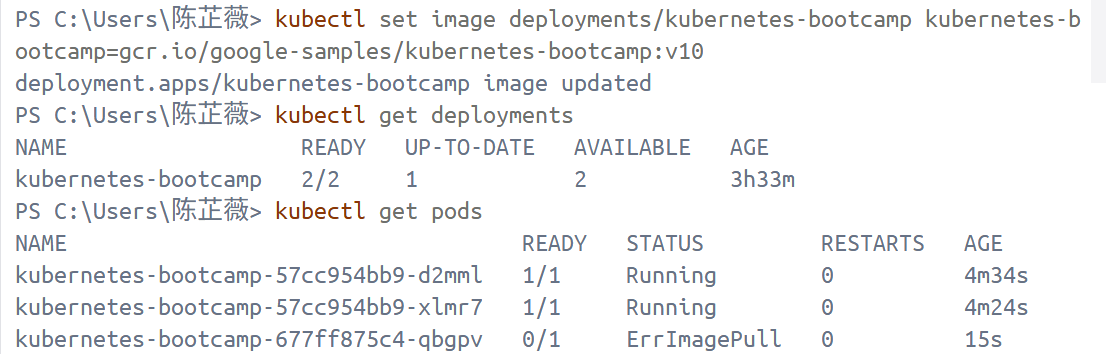
*kubectl describe pods*



1. Roll back an update
   1. Deploy an image tagged with v10

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

* 1. Use *get deployments* to see the status of the deployment
  2. Run the *get pods* subcommand to list all Pods



* 1. Get more insight into the problem about ImagePullBackOff

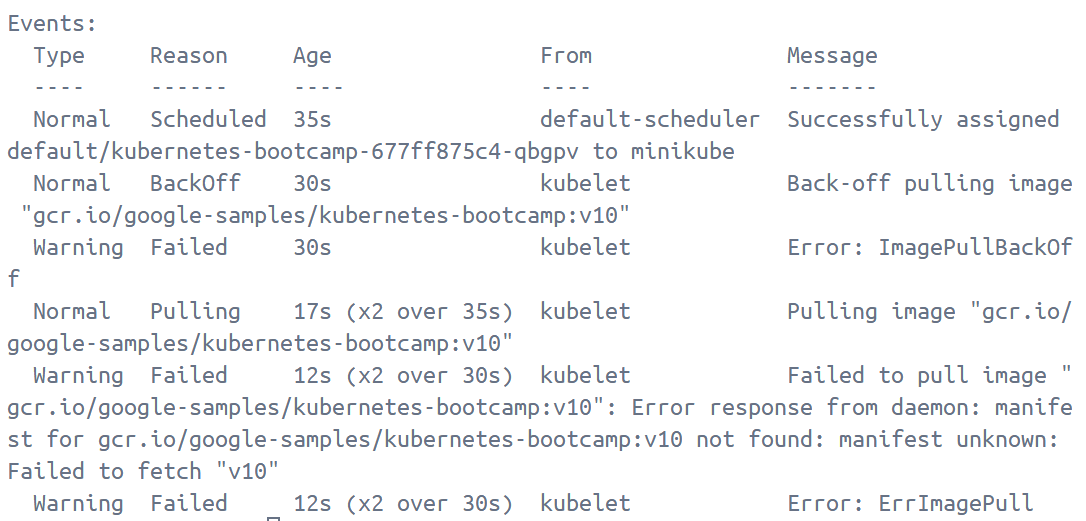
*kubectl describe pods*

图片包含 表格

AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。



In the Events section of the output for the affected Pods, notice that the v10 image version did not exist in the repository.

* 1. Roll back the deployment to the last working version

*kubectl rollout undo deployments/kubernetes-bootcamp*

The rollout undo command reverts the deployment to the previous known state (v2 of the image).

* 1. List the Pods

*kubectl get pods*

* 1. Check the image deployed on the running Pods

*kubectl describe pods*

* 1. Clean up the local cluster

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

文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。