1. Hello-minikube

**Command:**

minikube start

minikube status

**Note:**

* minikube start launches a local Kubernetes cluster, simulating a small server environment.
* minikube status checks the status of cluster components like host, kubelet, and apiserver.



***Figure1: Start minikube***

2. Deploy-app

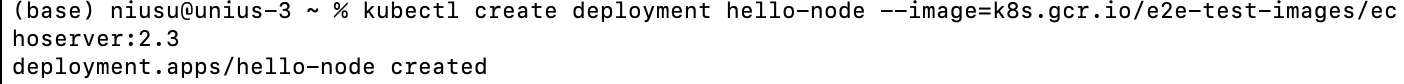
**Command:**

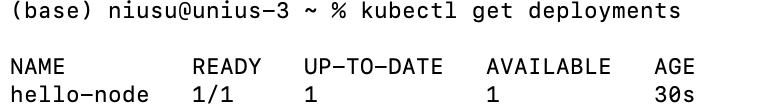
kubectl create deployment hello-node --image=k8s.gcr.io/e2e-test-images/echoserver:2.3

kubectl get deployments

**Note:**

* kubectl create deployment creates a Deployment object; Kubernetes manages Pods based on it automatically.
* --image specifies the container image; here, we use the official echoserver test image.
* kubectl get deployments shows the Deployment status and replica count.





***Figure2: Create deployment***

3. Explore

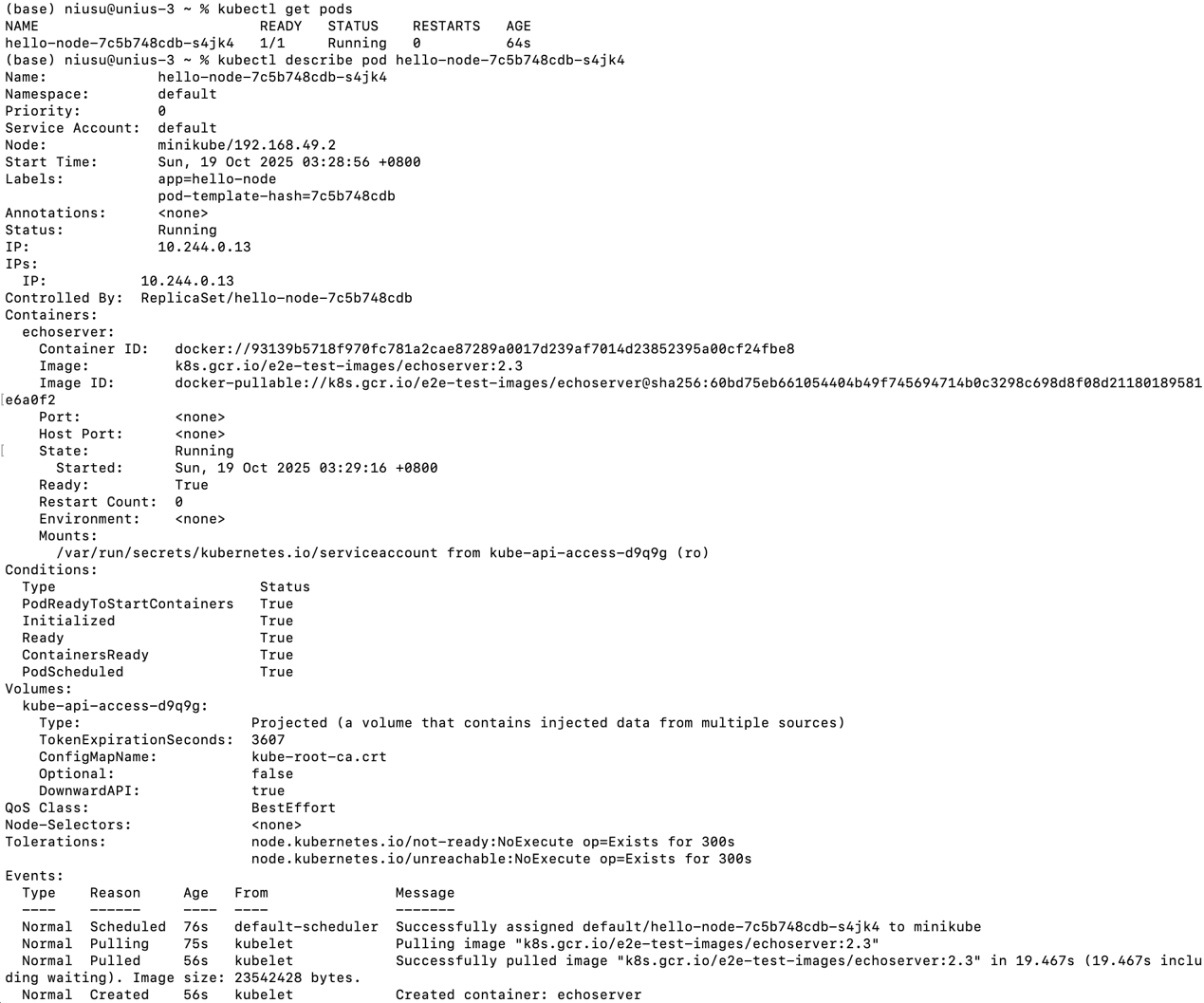
**Command:**

kubectl get pods

kubectl describe pod <pod-name>

**Note:**

* Deployment creates one or more Pods, the smallest running unit in Kubernetes.
* kubectl describe pod shows detailed Pod info: IP, status, container logs, etc.
* Pod names are dynamically generated and may change every time.



***Figure3: Explore the application***

4. Expose

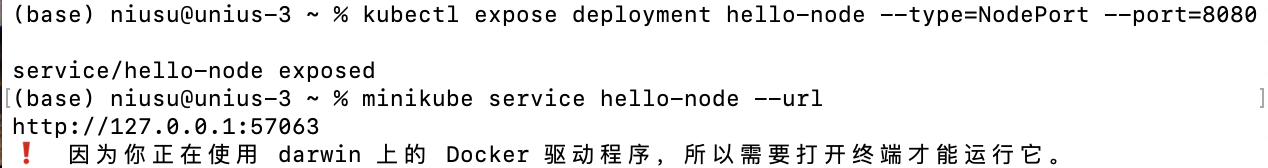
**Command:**

kubectl expose deployment hello-node --type=NodePort --port=8080

minikube service hello-node --url

**Note:**

* kubectl expose creates a Service to make Pods externally accessible.
* --type=NodePort exposes the application through a node port.
* minikube service --url provides a local URL to open the application in a browser.



***Figure4: Expose the application***

5. Scale

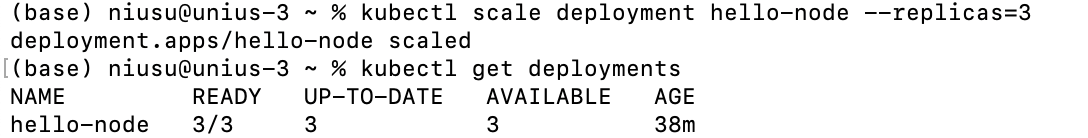
**Command:**

kubectl scale deployment hello-node --replicas=3

kubectl get deployments

**Note:**

* Scaling adjusts the number of application instances to handle more load.
* Kubernetes automatically starts or stops Pods to match the desired replica count.



***Figure5: Scale the application***

6. Update

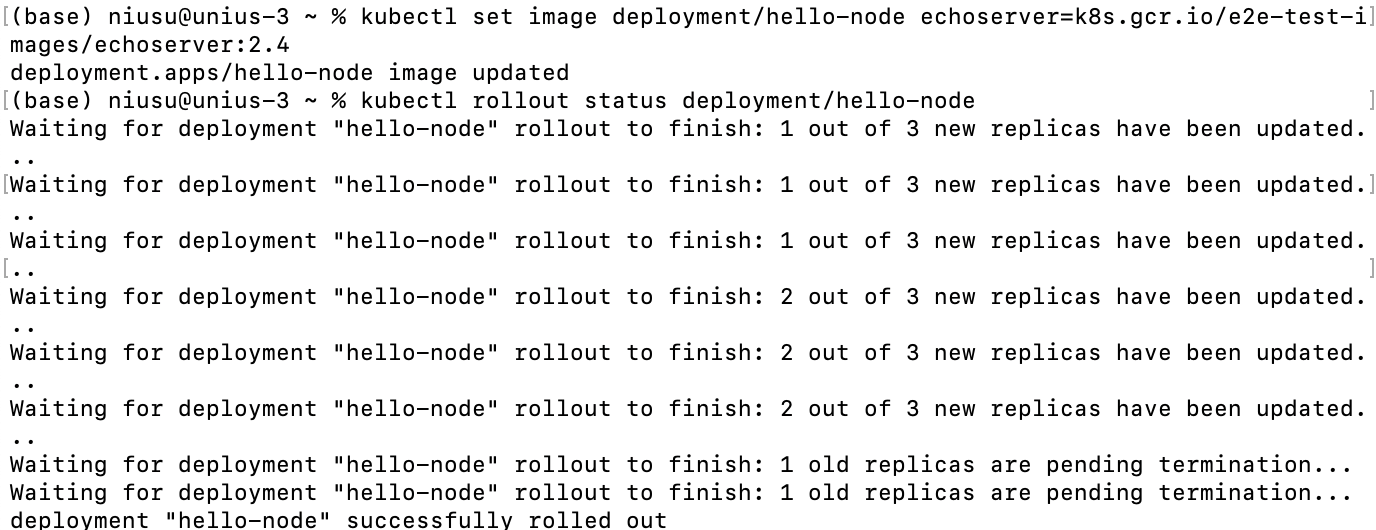
**Command:**

kubectl set image deployment/hello-node echoserver=k8s.gcr.io/e2e-test-images/echoserver:2.4

kubectl rollout status deployment/hello-node

**Note:**

* Updating the Deployment image triggers a rolling update; Kubernetes ensures the service remains available.
* kubectl rollout status shows the progress and confirms the new Pods are running.



***Figure6: Update the application***