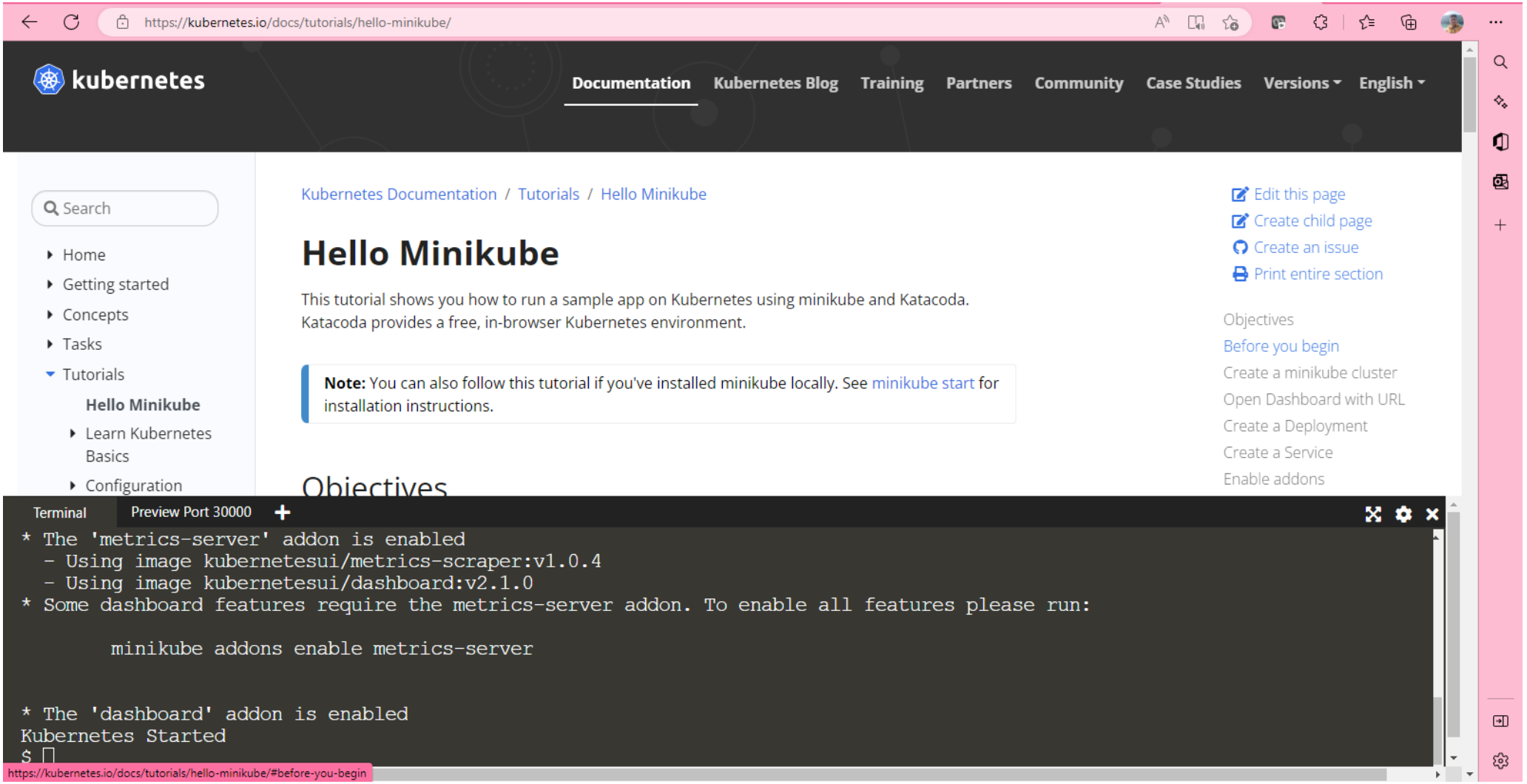
Assignment#3

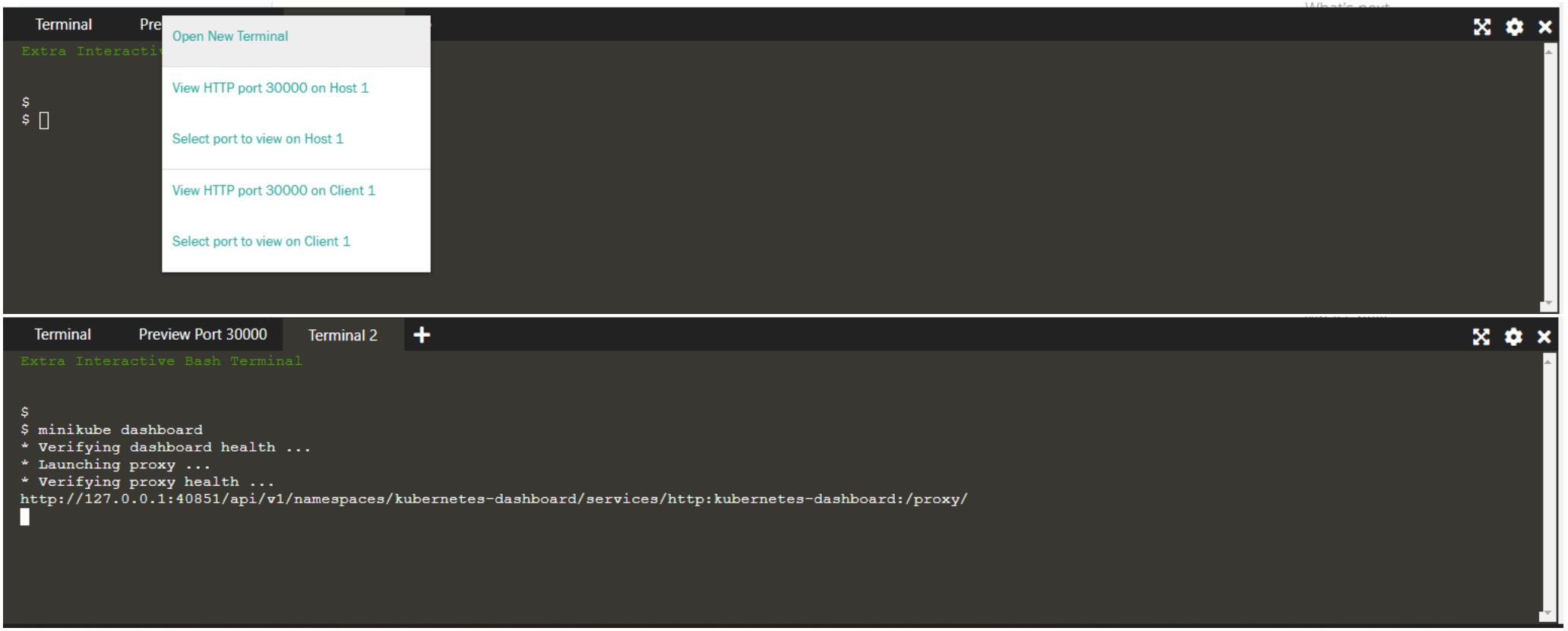
3. Launch minikube cluster and execute the commands to test the pod, container with the kubectl commands

---

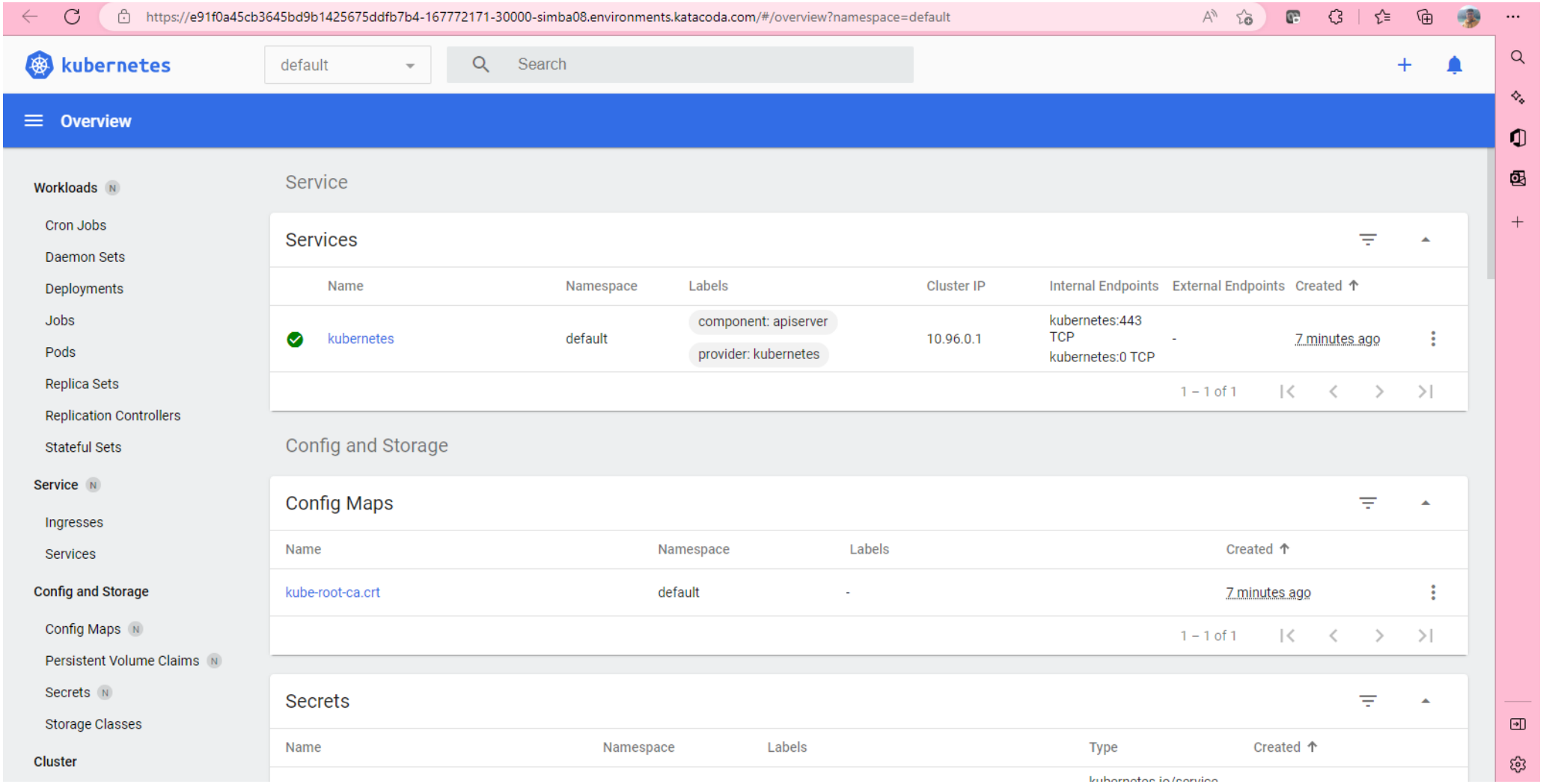


Command: `minikube start`

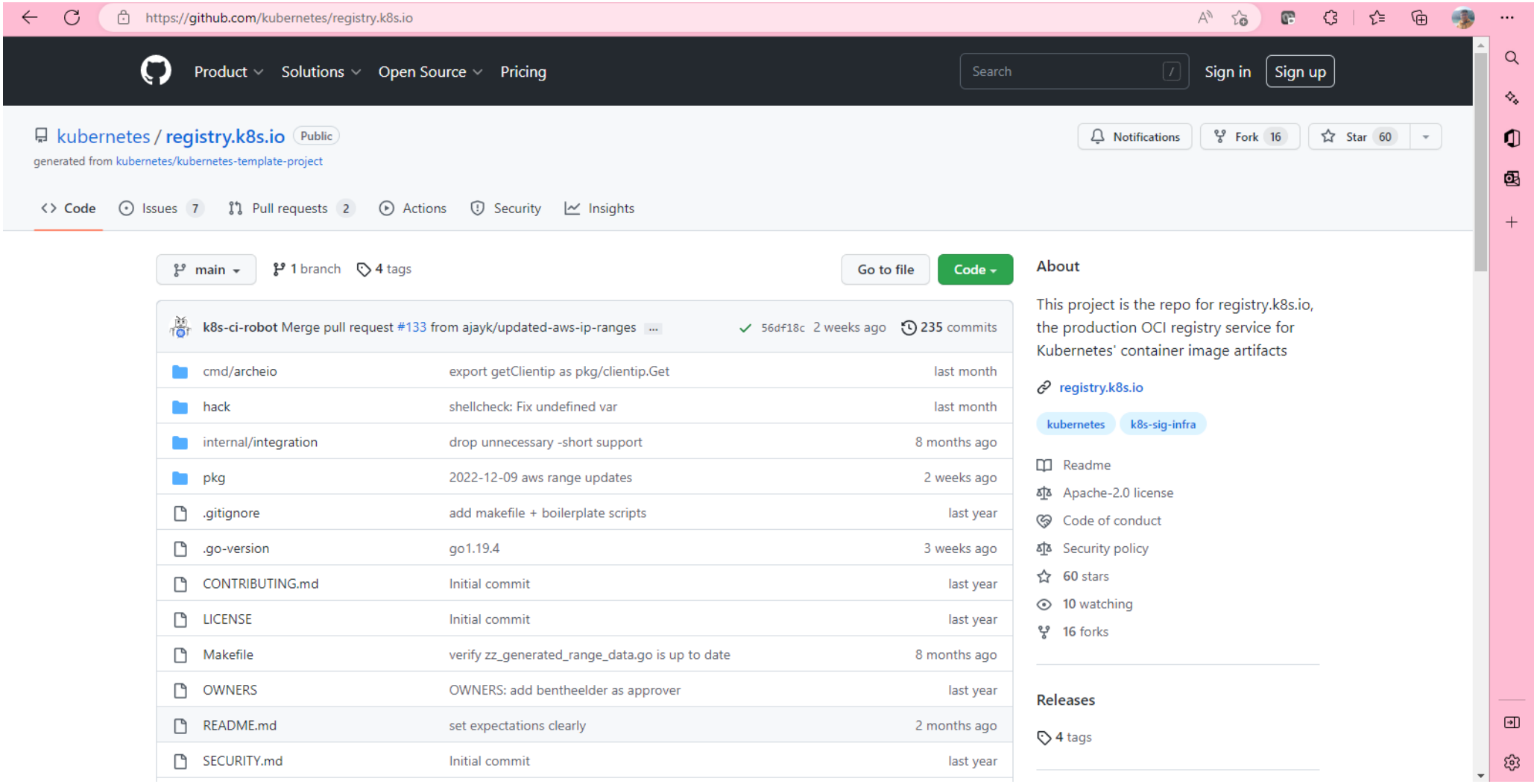
```
$ minikube start
* minikube v1.18.0 on Ubuntu 18.04 (amd64)
* Using the none driver based on existing profile

X The requested memory allocation of 2200MiB does not leave room for system overhead (total system memory: 2460MiB). You may face s
tability issues.
* Suggestion: Start minikube with less memory allocated: 'minikube start --memory=2200mb'

* Starting control plane node minikube in cluster minikube
* Updating the running none "minikube" bare metal machine ...
* OS release is Ubuntu 18.04.5 LTS
* Preparing Kubernetes v1.20.2 on Docker 19.03.13 ...
  - kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
* Configuring local host environment ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v4
  - Using image k8s.gcr.io/metrics-server-amd64:v0.2.1
  - Using image kubernetesui/dashboard:v2.1.0
  - Using image kubernetesui/metrics-scraper:v1.0.4
* Enabled addons: metrics-server, storage-provisioner, default-storageclass, dashboard
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
$
```



Minikube Dashboard


https://e91f0a45cb3645bd9b1425675ddfb7b4-167772171-30000-simba08.environments.katacoda.com/#/overview?namespace=default
kubernetes
default
Search
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
Cluster
Service
Services
Name
Namespace
Labels
Cluster IP
Internal Endpoints
External Endpoints
Created
kubernetes
default
component: apiserver
provider: kubernetes
10.96.0.1
kubernetes:443 TCP
kubernetes:0 TCP
-
7 minutes ago
1 – 1 of 1
Config and Storage
Config Maps
Name
Namespace
Labels
Created
kube-root-ca.crt
default
-
7 minutes ago
1 – 1 of 1
Secrets
Name
Namespace
Labels
Type
Created

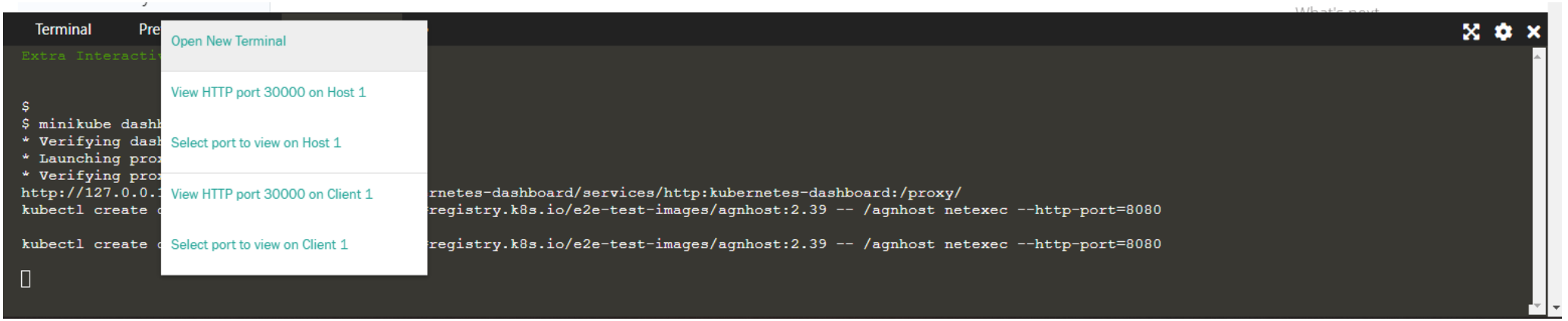
All Packages on Kubernetes GitHub repository: registry.k8s.io/



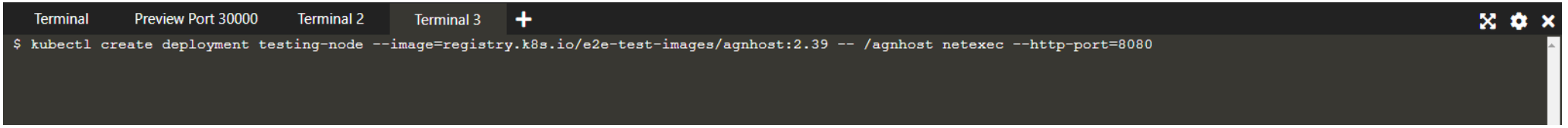
A Kubernetes Deployment checks on the health of your Pod and restarts the Pod's Container if it terminates.

Here 1 Pod=1 container

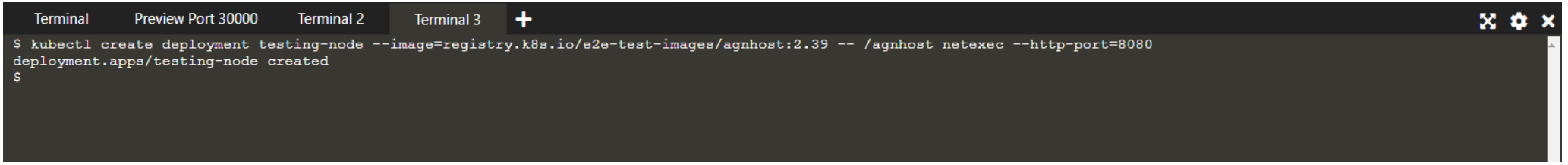
1) Use the kubectl create command to create a Deployment that manages a Pod. The Pod runs a Container based on the provided Docker image.



Creating a node, testing-node



A node created



To Display Namespaces

$ kubectl get ns

NAME STATUS AGE

default Active 23m

kube-node-lease Active 23m

kube-public Active 23m

kube-system Active 23m

kubernetes-dashboard Active 23m

kubectl get all -n kube-system

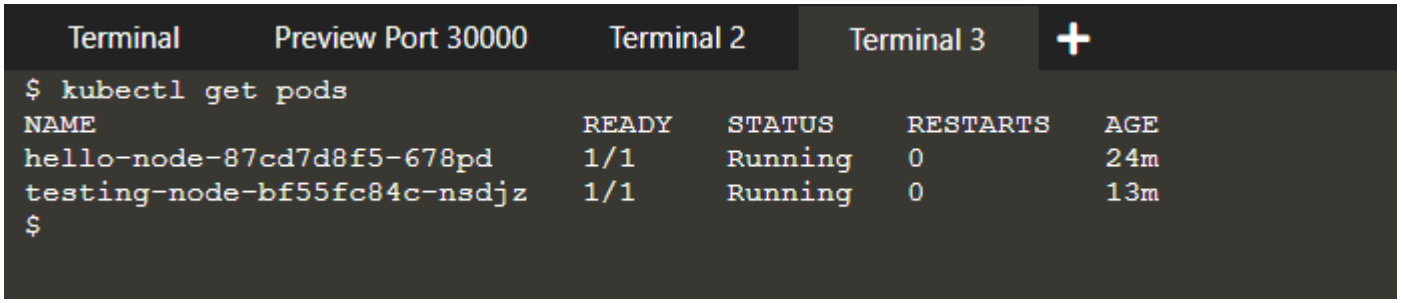
```
$ kubectl get all -n kube-system
NAME                                   READY   STATUS    RESTARTS   AGE
pod/coredns-74ff55c5b-xv985            1/1     Running   0          30m
pod/etcd-minikube                      1/1     Running   0          30m
pod/kube-apiserver-minikube            1/1     Running   0          30m
pod/kube-controller-manager-minikube   1/1     Running   0          30m
pod/kube-proxy-9lq87                   1/1     Running   0          30m
pod/kube-scheduler-minikube            1/1     Running   0          30m
pod/metrics-server-56c4f8c9d6-jh6g9    1/1     Running   0          30m
pod/storage-provisioner                1/1     Running   0          30m

NAME                     TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                  AGE
service/kube-dns         ClusterIP   10.96.0.10      <none>        53/UDP,53/TCP,9153/TCP   30m
service/metrics-server   ClusterIP   10.98.223.248   <none>        443/TCP                  30m

NAME                        DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   30m

NAME                             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/coredns          1/1     1            1           30m
deployment.apps/metrics-server   1/1     1            1           30m

NAME                                        DESIRED   CURRENT   READY   AGE
replicaset.apps/coredns-74ff55c5b           1         1         1       30m
replicaset.apps/metrics-server-56c4f8c9d6   1         1         1       30m
$ kubectl get all -n kube-system
NAME                                   READY   STATUS    RESTARTS   AGE
pod/coredns-74ff55c5b-xv985            1/1     Running   0          30m
pod/etcd-minikube                      1/1     Running   0          30m
pod/kube-apiserver-minikube            1/1     Running   0          30m
pod/kube-controller-manager-minikube   1/1     Running   0          30m
pod/kube-proxy-9lq87                   1/1     Running   0          30m
pod/kube-scheduler-minikube            1/1     Running   0          30m
pod/metrics-server-56c4f8c9d6-jh6g9    1/1     Running   0          30m
pod/storage-provisioner                1/1     Running   0          30m

NAME                     TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)                  AGE
service/kube-dns         ClusterIP   10.96.0.10      <none>        53/UDP,53/TCP,9153/TCP   30m
service/metrics-server   ClusterIP   10.98.223.248   <none>        443/TCP                  30m

NAME                        DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/kube-proxy   1         1         1       1            1           kubernetes.io/os=linux   30m

NAME                             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/coredns          1/1     1            1           30m
deployment.apps/metrics-server   1/1     1            1           30m

NAME                                        DESIRED   CURRENT   READY   AGE
replicaset.apps/coredns-74ff55c5b           1         1         1       30m
replicaset.apps/metrics-server-56c4f8c9d6   1         1         1       30m
$
```

Now, creating test name space

```
$ kubectl create ns test
namespace/test created
$
```

Get all pods:

Command: kubectl get pods



Two pods are getting.

To get logs of Pods:

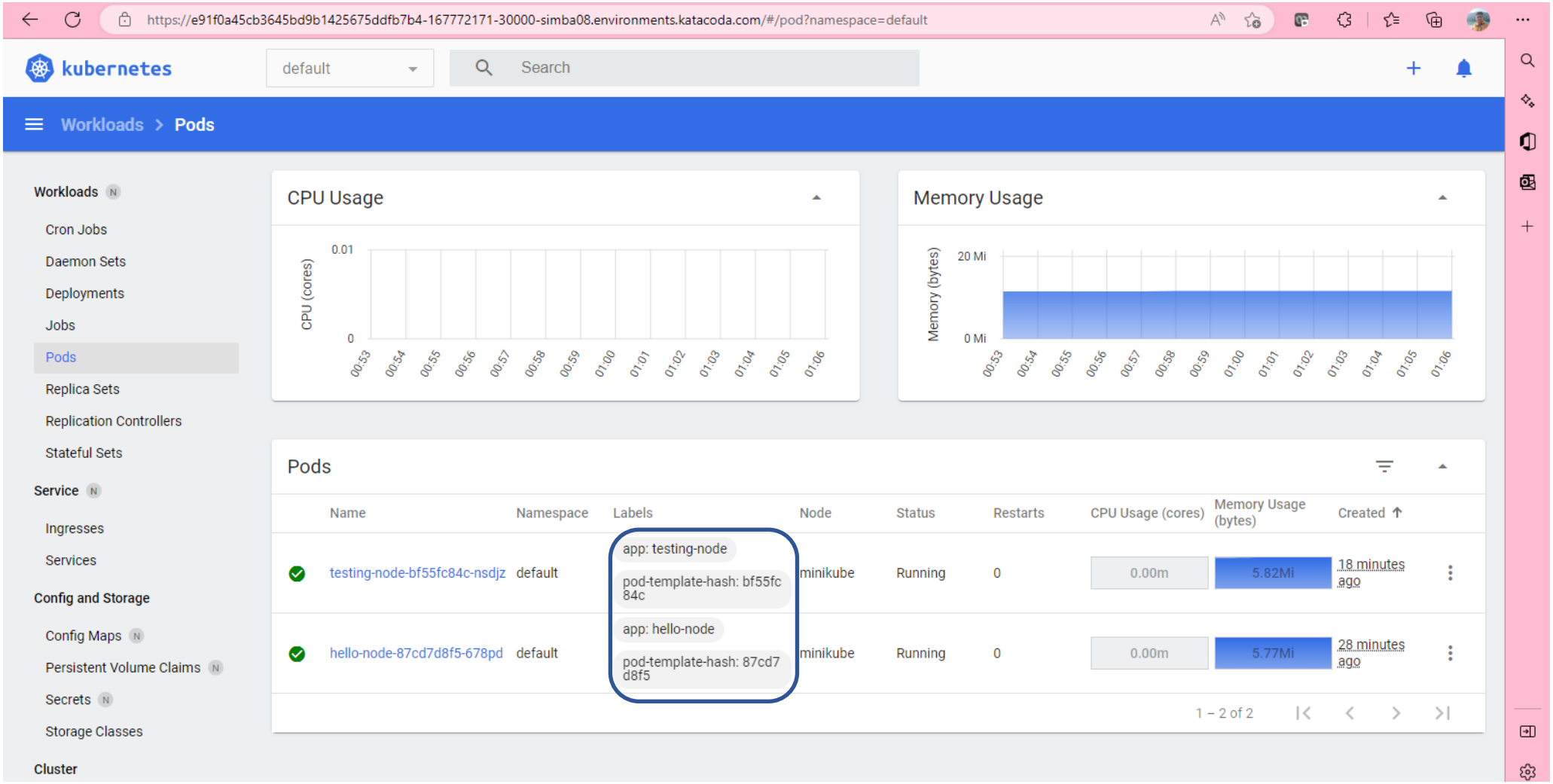
Command: kubectl logs [pod-name]

```
$ kubectl logs hello-node-87cd7d8f5-15lmv
I1226 20:05:07.998871       1 log.go:195] Started HTTP server on port 8080
I1226 20:05:07.999437       1 log.go:195] Started UDP server on port  8081
$ kubectl logs testing-node-bf55fc84c-dt78s
I1226 20:06:09.240532       1 log.go:195] Started HTTP server on port 8080
I1226 20:06:09.241238       1 log.go:195] Started UDP server on port  8081
$
```

To get how many pods are deployed:

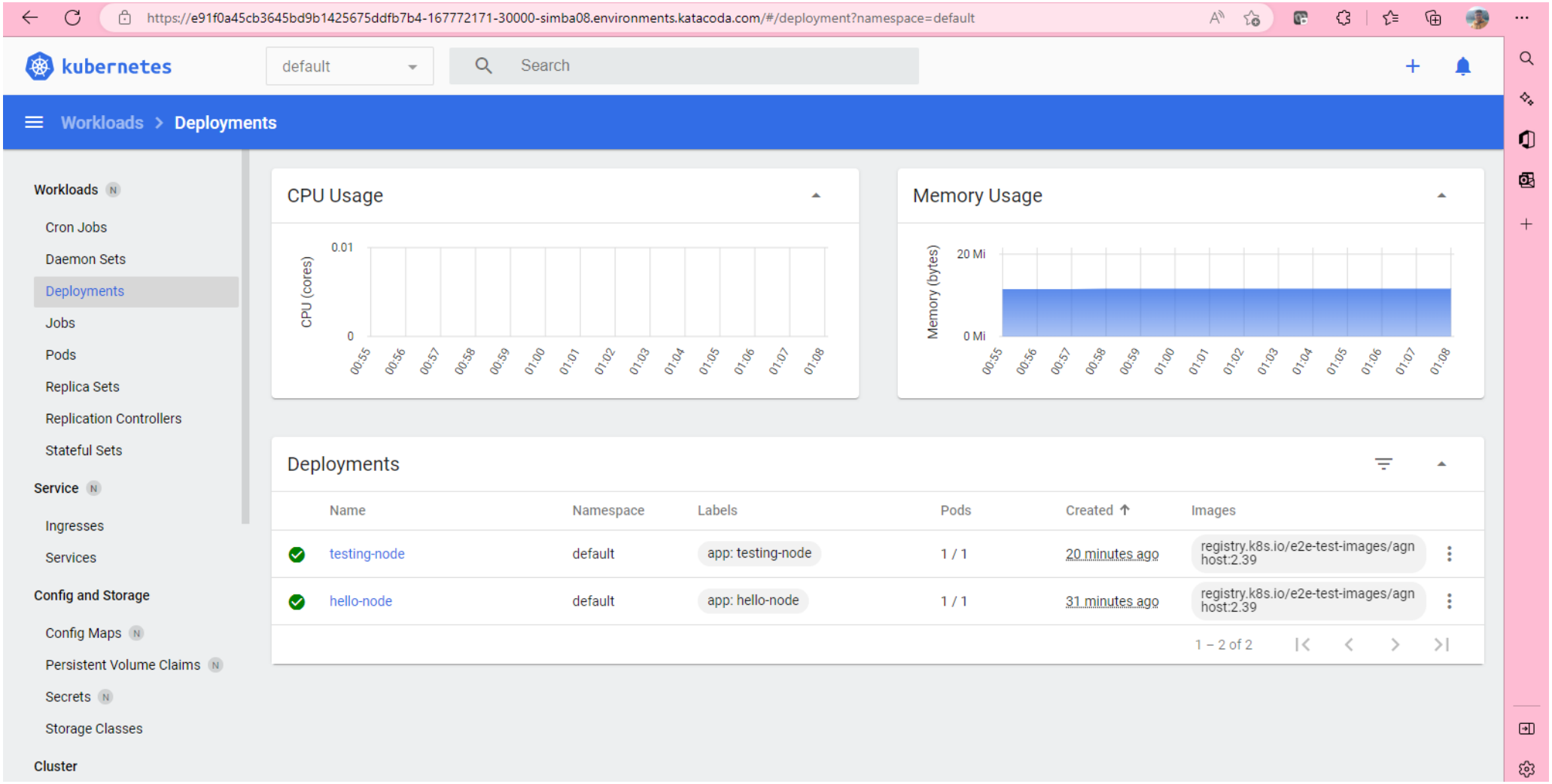
```
$ kubectl get deployment
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
hello-node     1/1     1            1           27m
testing-node   1/1     1            1           16m
$
```

Now from Dashboard Pods Status:



Thus, two pods are showing.

Now Status about deployments:



Now kubectl describe pods

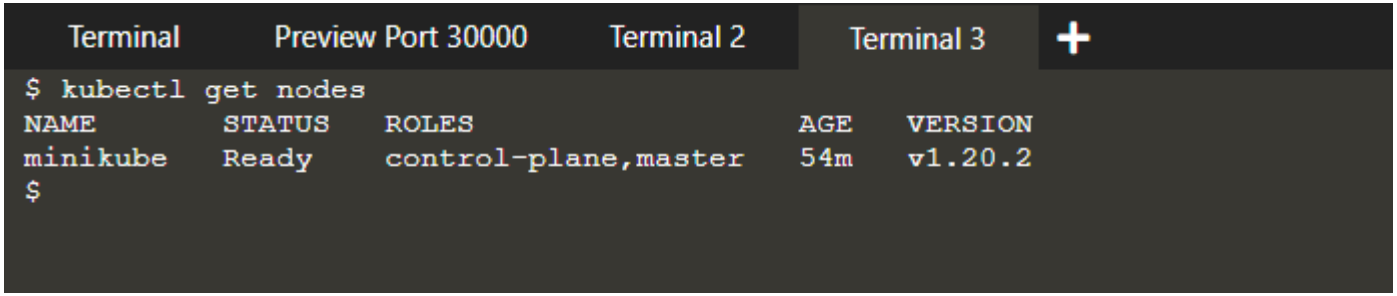
Terminal | Preview Port 30000 | Terminal 2 | Terminal 3 | +

```
$ kubectl describe pods
Name:         hello-node-87cd7d8f5-678pd
Namespace:    default
Priority:     0
Node:         minikube/10.0.0.11
Start Time:   Mon, 26 Dec 2022 19:08:46 +0000
Labels:       app=hello-node
              pod-template-hash=87cd7d8f5
Annotations:  <none>
Status:       Running
IP:           172.18.0.6
IPs:
  IP:           172.18.0.6
Controlled By:  ReplicaSet/hello-node-87cd7d8f5
Containers:
  agnhost:
    Container ID:  docker://da864bbb4daac3eef816298bfc525334c92de9c9ff1a63010fcc61671c012a5b
    Image:         registry.k8s.io/e2e-test-images/agnhost:2.39
    Image ID:      docker-pullable://registry.k8s.io/e2e-test-images/agnhost@sha256:7e8bdd271312fd25fc5ff5a8f04727be84044eb3d7d8d03611972a6752e2e11e
    Port:          <none>
    Host Port:     <none>
    Command:
      /agnhost
      netexec
      --http-port=8080
    State:          Running
      Started:      Mon, 26 Dec 2022 19:08:51 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tp279 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  default-token-tp279:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-tp279
    Optional:    false
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:
```

```
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  33m   default-scheduler  Successfully assigned default/hello-node-87cd7d8f5-678pd to minikube
  Normal  Pulling    33m   kubelet            Pulling image "registry.k8s.io/e2e-test-images/agnhost:2.39"
  Normal  Pulled     33m   kubelet            Successfully pulled image "registry.k8s.io/e2e-test-images/agnhost:2.39" in 3.832334726s
  Normal  Created    33m   kubelet            Created container agnhost
  Normal  Started    33m   kubelet            Started container agnhost


Name:         testing-node-bf55fc84c-nsdjz
Namespace:    default
Priority:     0
Node:         minikube/10.0.0.11
Start Time:   Mon, 26 Dec 2022 19:19:31 +0000
Labels:       app=testing-node
              pod-template-hash=bf55fc84c
Annotations:  <none>
Status:       Running
IP:           172.18.0.7
IPs:
  IP:           172.18.0.7
Controlled By:  ReplicaSet/testing-node-bf55fc84c
Containers:
  agnhost:
    Container ID:  docker://12788fdc2ff1cdb1a3deba71d735519625fadc39bd1933328583cad226dff283
    Image:         registry.k8s.io/e2e-test-images/agnhost:2.39
    Image ID:      docker-pullable://registry.k8s.io/e2e-test-images/agnhost@sha256:7e8bdd271312fd25fc5ff5a8f04727be84044eb3d7d8d03611972a6752e2e11e
    Port:          <none>
    Host Port:     <none>
    Command:
      /agnhost
      netexec
      --http-port=8080
    State:          Running
      Started:      Mon, 26 Dec 2022 19:19:32 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-tp279 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
```

```
Volumes:
  default-token-tp279:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-tp279
    Optional:    false
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:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  22m   default-scheduler  Successfully assigned default/testing-node-bf55fc84c-nsdjz to minikube
  Normal  Pulled     22m   kubelet            Container image "registry.k8s.io/e2e-test-images/agnhost:2.39" already present on machine
  Normal  Created    22m   kubelet            Created container agnhost
  Normal  Started    22m   kubelet            Started container agnhost
$
```

To get Nodes details:



To Check Cluster Events:

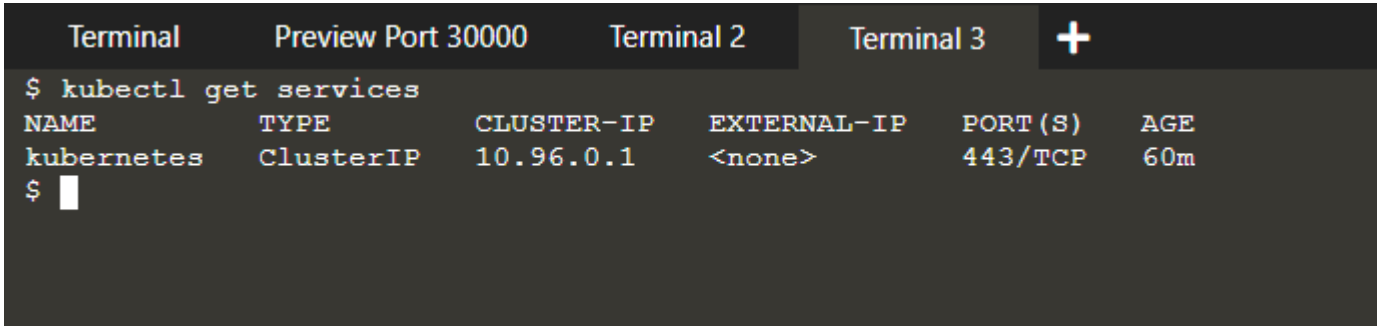
Commands: kubectl get events

```
$ kubectl get events
LAST SEEN   TYPE     REASON                    OBJECT                                  MESSAGE
44m         Normal   Scheduled                 pod/hello-node-87cd7d8f5-678pd          Successfully assigned default/hello-node-87cd7d8f5-678pd to minikube
44m         Normal   Pulling                   pod/hello-node-87cd7d8f5-678pd          Pulling image "registry.k8s.io/e2e-test-images/agnhost:2.39"
44m         Normal   Pulled                    pod/hello-node-87cd7d8f5-678pd          Successfully pulled image "registry.k8s.io/e2e-test-images/agnhost:2.39" in 3.832334726s
44m         Normal   Created                   pod/hello-node-87cd7d8f5-678pd          Created container agnhost
44m         Normal   Started                   pod/hello-node-87cd7d8f5-678pd          Started container agnhost
44m         Normal   SuccessfulCreate          replicaset/hello-node-87cd7d8f5         Created pod: hello-node-87cd7d8f5-678pd
44m         Normal   ScalingReplicaSet         deployment/hello-node                   Scaled up replica set hello-node-87cd7d8f5 to 1
55m         Normal   NodeHasSufficientMemory   node/minikube                           Node minikube status is now: NodeHasSufficientMemory
55m         Normal   NodeHasNoDiskPressure     node/minikube                           Node minikube status is now: NodeHasNoDiskPressure
55m         Normal   NodeHasSufficientPID      node/minikube                           Node minikube status is now: NodeHasSufficientPID
55m         Normal   Starting                  node/minikube                           Starting kubelet.
55m         Normal   NodeHasSufficientMemory   node/minikube                           Node minikube status is now: NodeHasSufficientMemory
55m         Normal   NodeHasNoDiskPressure     node/minikube                           Node minikube status is now: NodeHasNoDiskPressure
55m         Normal   NodeHasSufficientPID      node/minikube                           Node minikube status is now: NodeHasSufficientPID
55m         Normal   NodeNotReady              node/minikube                           Node minikube status is now: NodeNotReady
55m         Normal   NodeAllocatableEnforced   node/minikube                           Updated Node Allocatable limit across pods
55m         Normal   RegisteredNode            node/minikube                           Node minikube event: Registered Node minikube in Controller
55m         Normal   NodeReady                 node/minikube                           Node minikube status is now: NodeReady
55m         Normal   Starting                  node/minikube                           Starting kube-proxy.
33m         Normal   Scheduled                 pod/testing-node-bf55fc84c-nsdjz        Successfully assigned default/testing-node-bf55fc84c-nsdjz to minikube
33m         Normal   Pulled                    pod/testing-node-bf55fc84c-nsdjz        Container image "registry.k8s.io/e2e-test-images/agnhost:2.39" already present on machine
33m         Normal   Created                   pod/testing-node-bf55fc84c-nsdjz        Created container agnhost
33m         Normal   Started                   pod/testing-node-bf55fc84c-nsdjz        Started container agnhost
33m         Normal   SuccessfulCreate          replicaset/testing-node-bf55fc84c       Created pod: testing-node-bf55fc84c-nsdjz
33m         Normal   ScalingReplicaSet         deployment/testing-node                 Scaled up replica set testing-node-bf55fc84c to 1
$
```

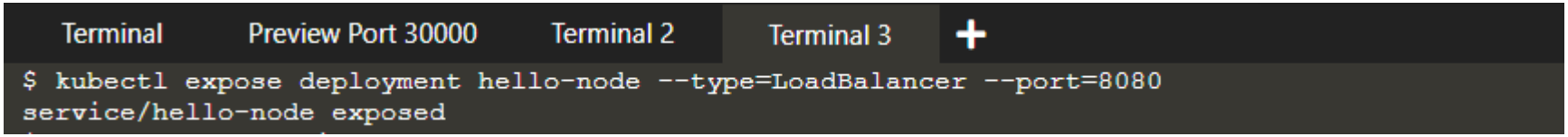
To check configurations:

```
$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority: /root/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Mon, 26 Dec 2022 19:01:10 UTC
        provider: minikube.sigs.k8s.io
        version: v1.18.0
      name: cluster_info
    server: https://10.0.0.11:8443
  name: minikube
contexts:
- context:
    cluster: minikube
    extensions:
    - extension:
        last-update: Mon, 26 Dec 2022 19:01:10 UTC
        provider: minikube.sigs.k8s.io
        version: v1.18.0
      name: context_info
    namespace: default
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: minikube
  user:
    client-certificate: /root/.minikube/profiles/minikube/client.crt
    client-key: /root/.minikube/profiles/minikube/client.key
$
```

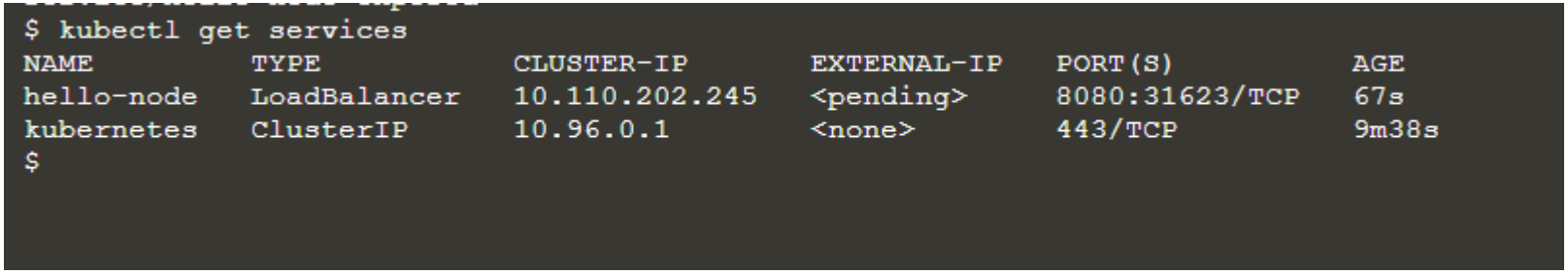
To check services by: kubectl get services



Pod to the public internet using the kubectl expose command:



To Check the created service:



To Check Cluster info:

```
$ kubectl cluster-info
Kubernetes control plane is running at https://10.0.0.17:8443
KubeDNS is running at https://10.0.0.17:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
$
```

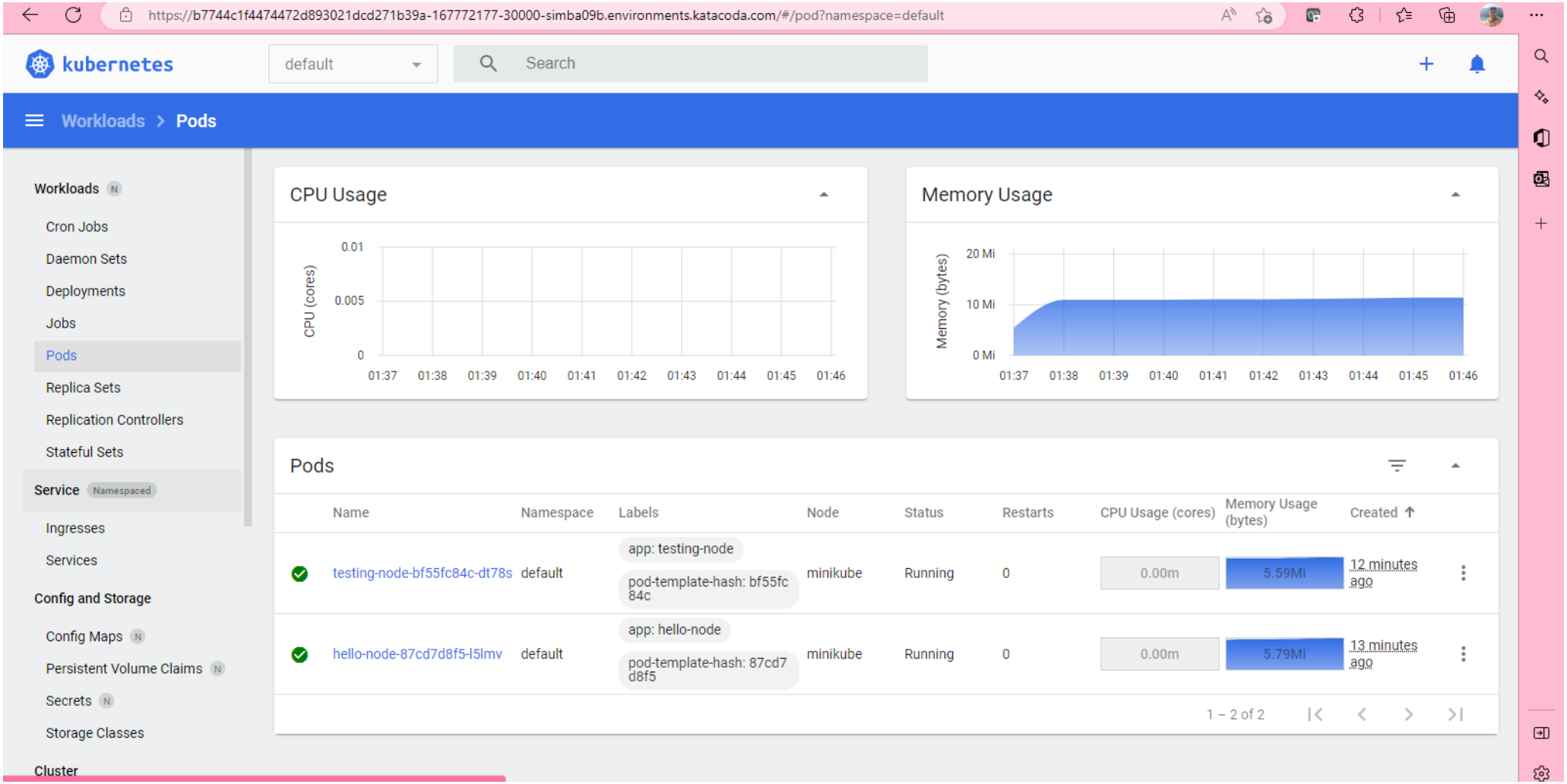
Checking Service of Specific node:

```
$ kubectl get services
NAME         TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
hello-node   LoadBalancer   10.110.202.245   <pending>     8080:31623/TCP   67s
kubernetes   ClusterIP      10.96.0.1        <none>        443/TCP          9m38s
$ minikube service hello-node
|-----------|------------|-------------|-------------------------|
| NAMESPACE |    NAME    | TARGET PORT |           URL           |
|-----------|------------|-------------|-------------------------|
| default   | hello-node |        8080 | http://10.0.0.17:31623  |
|-----------|------------|-------------|-------------------------|
* Opening service default/hello-node in default browser...
Minikube Dashboard is not supported via the interactive terminal experience.

Please click the 'Preview Port 30000' link above to access the dashboard.
This will now exit. Please continue with the rest of the tutorial.

X Exiting due to HOST_BROWSER: exit status 1
*
* If the above advice does not help, please let us know:
  - https://github.com/kubernetes/minikube/issues/new/choose

$
```

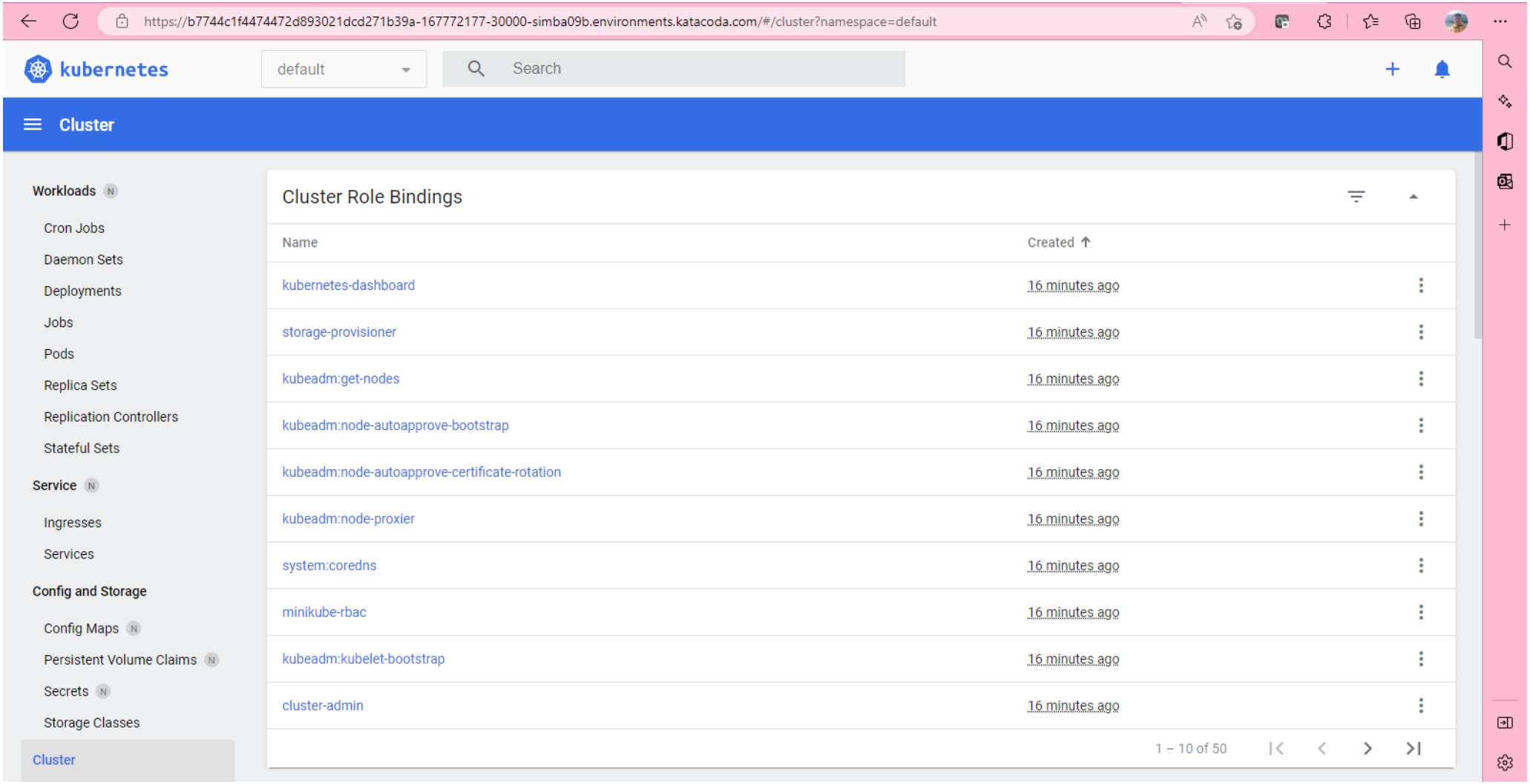
Now testing-node:

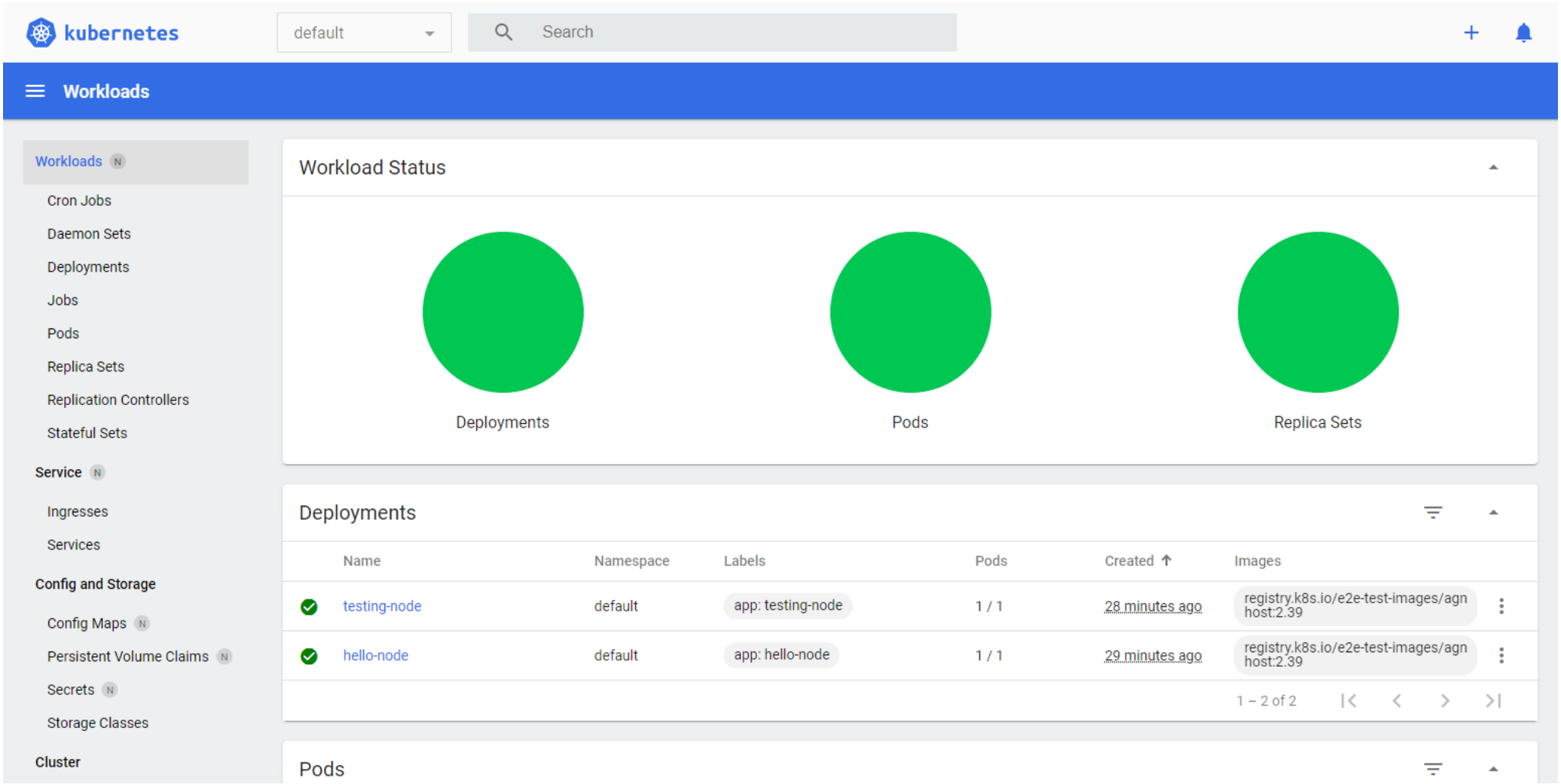
```
$ kubectl expose deployment testing-node --type=LoadBalancer --port=8080
service/testing-node exposed
$ minikube service testing-node
|-----------|--------------|-------------|------------------------|
| NAMESPACE |     NAME     | TARGET PORT |          URL           |
|-----------|--------------|-------------|------------------------|
| default   | testing-node |        8080 | http://10.0.0.17:31528 |
|-----------|--------------|-------------|------------------------|
* Opening service default/testing-node in default browser...
Minikube Dashboard is not supported via the interactive terminal experience.

Please click the 'Preview Port 30000' link above to access the dashboard.
This will now exit. Please continue with the rest of the tutorial.

X Exiting due to HOST_BROWSER: exit status 1
*
* If the above advice does not help, please let us know:
  - https://github.com/kubernetes/minikube/issues/new/choose

$
```

Now pods Status:



Now Cluster status:


https://b7744c1f4474472d893021dcd271b39a-167772177-30000-simba09b.environments.katacoda.com/#/cluster?namespace=default
kubernetes
default
Search
Cluster
Workloads N
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service N
Ingresses
Services
Config and Storage
Config Maps N
Persistent Volume Claims N
Secrets N
Storage Classes
Cluster
Cluster Role Bindings
Name
Created
kubernetes-dashboard 16 minutes ago
storage-provisioner 16 minutes ago
kubeadm:get-nodes 16 minutes ago
kubeadm:node-autoapprove-bootstrap 16 minutes ago
kubeadm:node-autoapprove-certificate-rotation 16 minutes ago
kubeadm:node-proxier 16 minutes ago
system:coredns 16 minutes ago
minikube-rbac 16 minutes ago
kubeadm:kubelet-bootstrap 16 minutes ago
cluster-admin 16 minutes ago
1 – 10 of 50


kubernetes
default
Search
Workloads
Workloads N
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service N
Ingresses
Services
Config and Storage
Config Maps N
Persistent Volume Claims N
Secrets N
Storage Classes
Cluster
Workload Status
Deployments
Pods
Replica Sets
Deployments
Name
Namespace
Labels
Pods
Created
Images
testing-node
default
app: testing-node
1 / 1
28 minutes ago
registry.k8s.io/e2e-test-images/agnhost:2.39
hello-node
default
app: hello-node
1 / 1
29 minutes ago
registry.k8s.io/e2e-test-images/agnhost:2.39
1 – 2 of 2
Pods