Project: **Managing Deployments Using Google Kubernetes Engine (GKE)**

kubernetes

Google Cloud Platform

ANSIBLE

HashiCorp
Terraform

Overview

Dev Ops practices make use of multiple deployments to manage application deployment scenarios such as "Continuous deployment", "Blue-Green deployments", "Canary deployments" and more.

This project demonstrates how to scale and manage containers so you can accomplish these common scenarios where multiple heterogeneous deployments are being used.

Project contains the following tasks:

- Use the kubectl tool
- Create deployment YAML files
- Launch, update, and scale deployments
- Update deployments and learn about deployment styles

Heterogeneous deployments typically involve connecting two or more distinct infrastructure environments or regions.

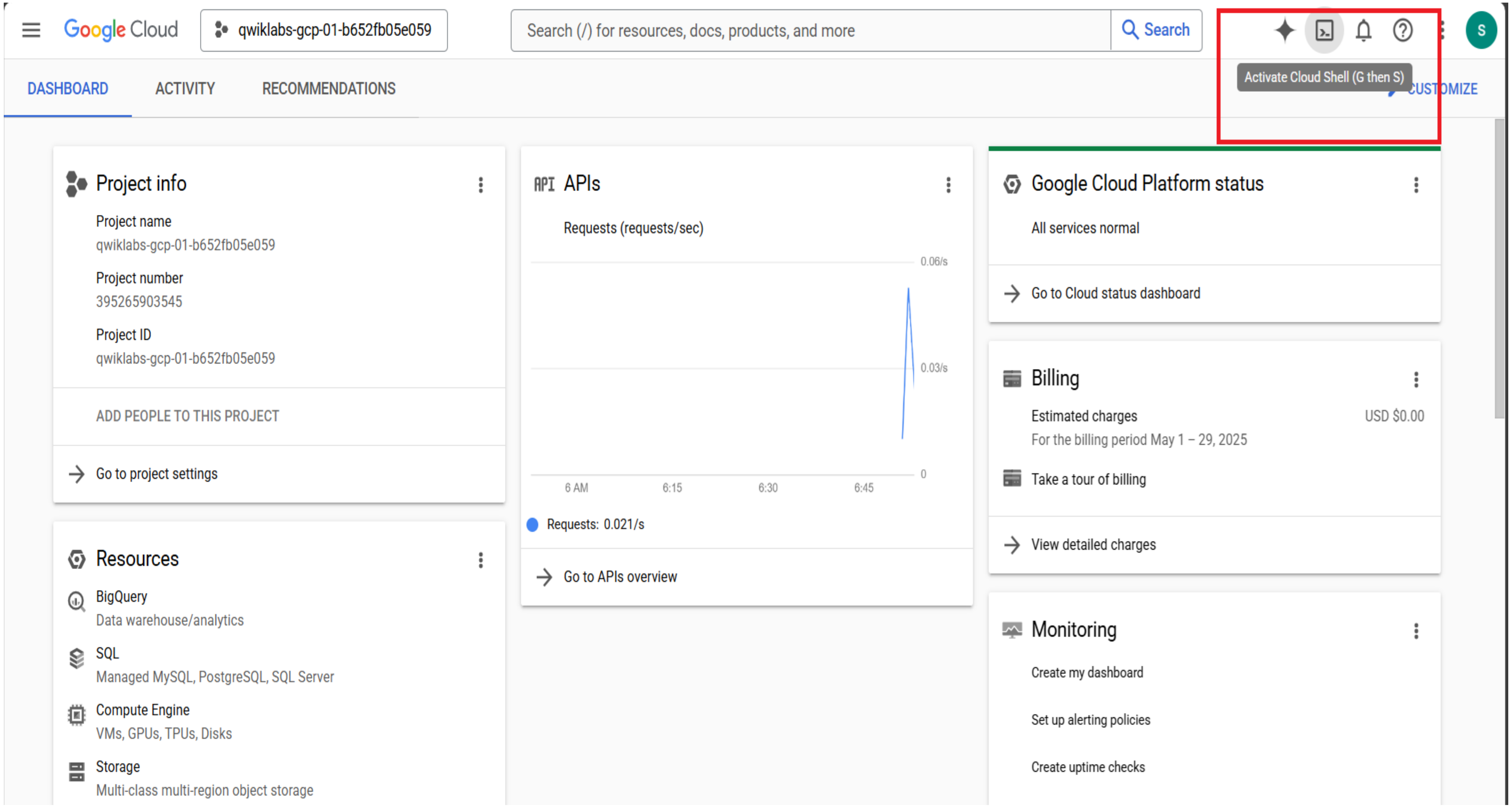
Three common scenarios for heterogeneous deployment are:

- multi-cloud deployments
- fronting on-premises data
- continuous integration/continuous delivery (CI/CD) processes

Refer: GKE overview | Google Kubernetes Engine (GKE) | Google Cloud

STEPS

Activate GCP Cloud Shell


Google Cloud
qwiklabs-gcp-01-b652fb05e059
Search (/) for resources, docs, products, and more
Search
Activate Cloud Shell (G then S)
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE

Project info

Project name
qwiklabs-gcp-01-b652fb05e059

Project number
395265903545

Project ID
qwiklabs-gcp-01-b652fb05e059

ADD PEOPLE TO THIS PROJECT

Go to project settings

APIs

Requests (requests/sec)

0.06/s
0.03/s
0
6 AM
6:15
6:30
6:45

● Requests: 0.021/s

Go to APIs overview

Google Cloud Platform status

All services normal

Go to Cloud status dashboard

Billing

Estimated charges USD $0.00
For the billing period May 1 – 29, 2025

Take a tour of billing

View detailed charges

Resources

BigQuery
Data warehouse/analytics

SQL
Managed MySQL, PostgreSQL, SQL Server

Compute Engine
VMs, GPUs, TPUs, Disks

Storage
Multi-class multi-region object storage

Monitoring

Create my dashboard

Set up alerting policies

Create uptime checks

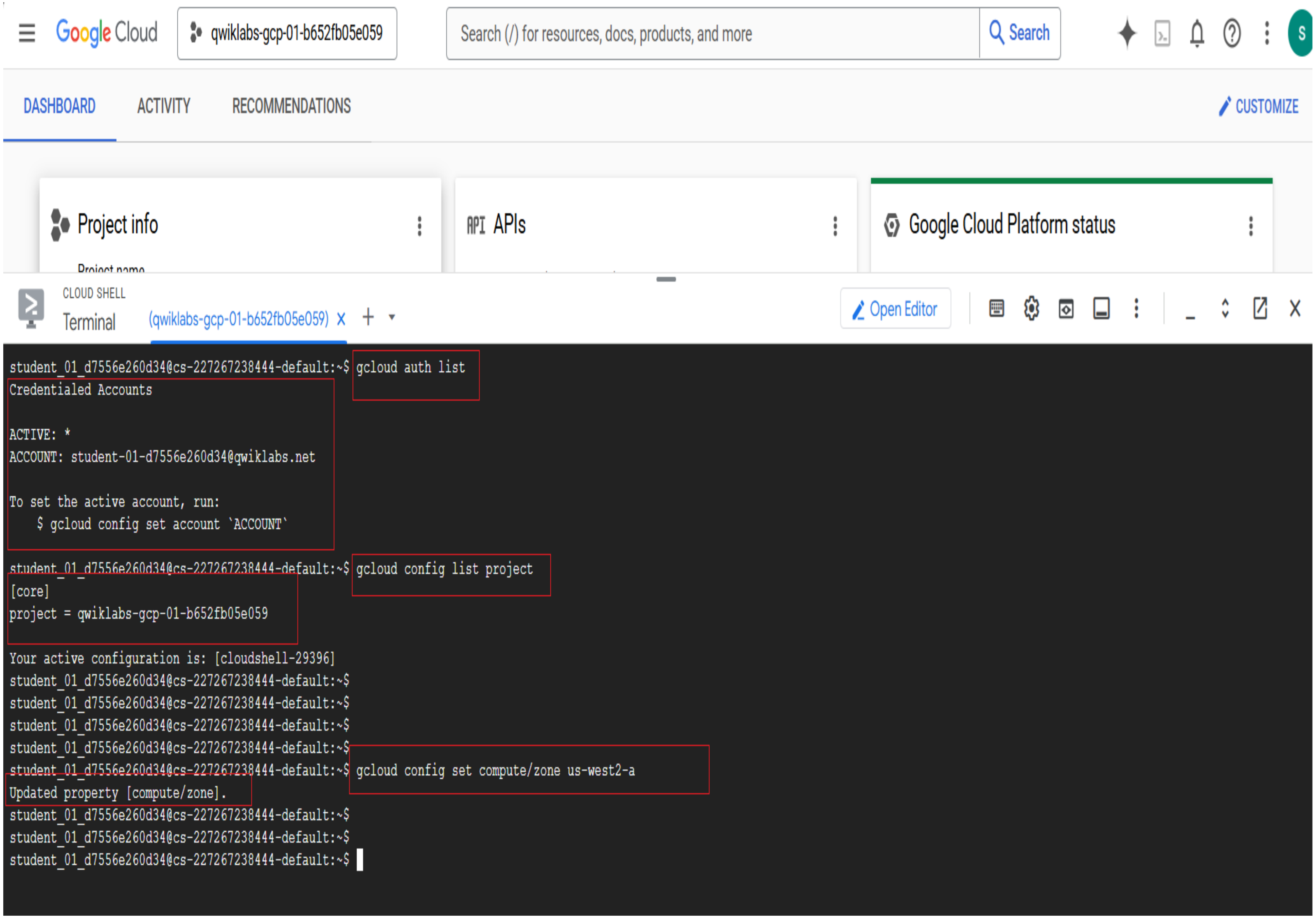
List the active account name: gcloud auth list

Set the active account: gcloud config set account `ACCOUNT`

List the project ID: gcloud config list project

Refer gcloud CLI overview | Google Cloud CLI Documentation

To substituting the local zone as us-west2-a: gcloud config set compute/zone us-west2-a

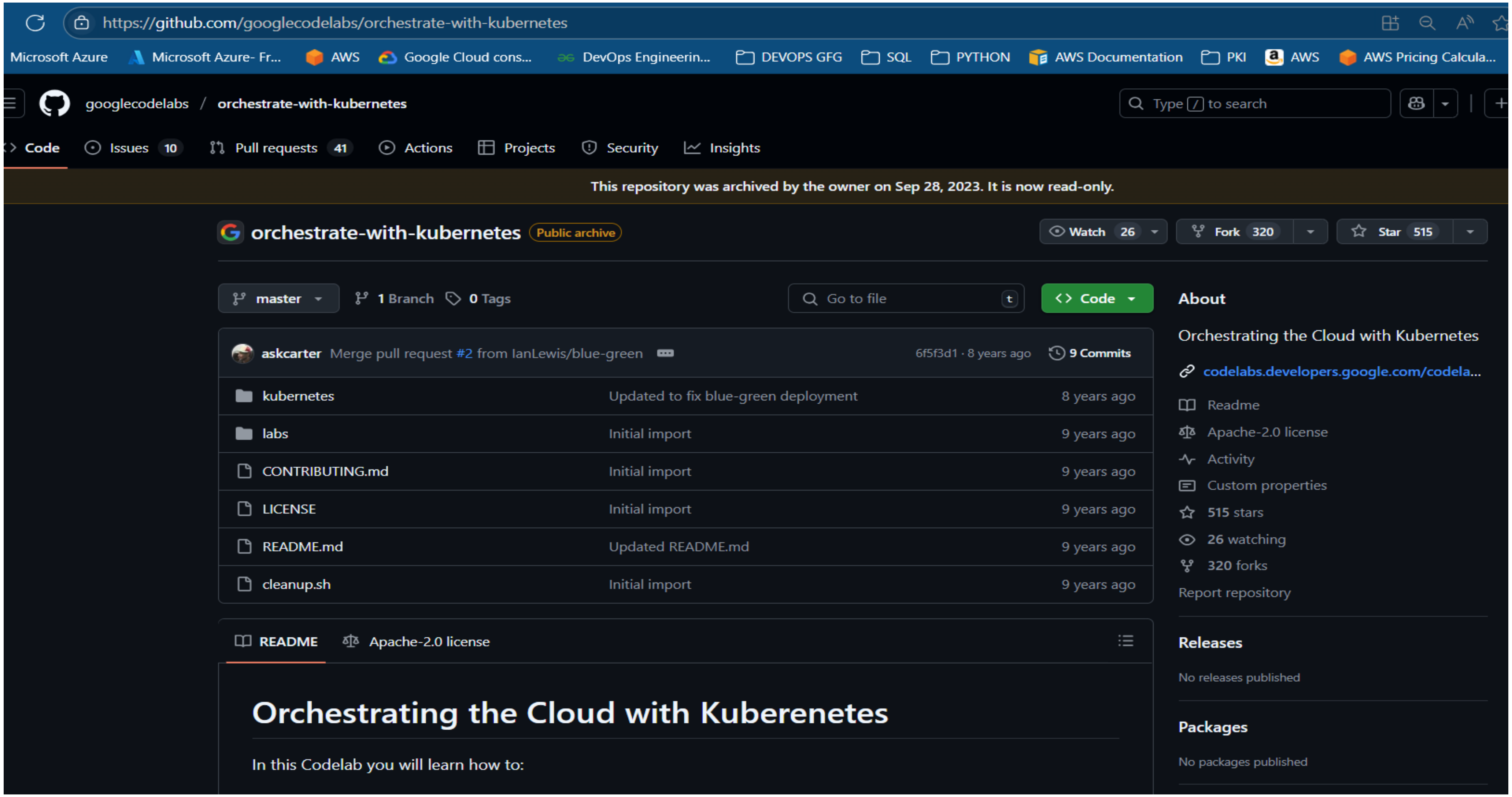

Google Cloud
qwiklabs-gcp-01-b652fb05e059
Search (/) for resources, docs, products, and more
Search
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
Project info
APIs
Google Cloud Platform status
CLOUD SHELL
Terminal
(qwiklabs-gcp-01-b652fb05e059)
Open Editor

```
student_01_d7556e260d34@cs-227267238444-default:~$ gcloud auth list
Credentialed Accounts

ACTIVE: *
ACCOUNT: student-01-d7556e260d34@qwiklabs.net

To set the active account, run:
    $ gcloud config set account `ACCOUNT`

student_01_d7556e260d34@cs-227267238444-default:~$ gcloud config list project
[core]
project = qwiklabs-gcp-01-b652fb05e059

Your active configuration is: [cloudshell-29396]
student_01_d7556e260d34@cs-227267238444-default:~$
student_01_d7556e260d34@cs-227267238444-default:~$
student_01_d7556e260d34@cs-227267238444-default:~$
student_01_d7556e260d34@cs-227267238444-default:~$
student_01_d7556e260d34@cs-227267238444-default:~$ gcloud config set compute/zone us-west2-a
Updated property [compute/zone].
student_01_d7556e260d34@cs-227267238444-default:~$
student_01_d7556e260d34@cs-227267238444-default:~$
student_01_d7556e260d34@cs-227267238444-default:~$
```

Get sample code for this lab

Get the sample code for creating and running containers and deployments: `gsutil -m cp -r gs://spls/gsp053/orchestrate-with-kubernetes .`

This refers to git repo: [googlecodelabs/orchestrate-with-kubernetes: Orchestrating the Cloud with Kubernetes](https://github.com/googlecodelabs/orchestrate-with-kubernetes)


https://github.com/googlecodelabs/orchestrate-with-kubernetes
Microsoft Azure | Microsoft Azure- Fr... | AWS | Google Cloud cons... | DevOps Engineerin... | DEVOPS GFG | SQL | PYTHON | AWS Documentation | PKI | AWS | AWS Pricing Calcula...
googlecodelabs / orchestrate-with-kubernetes
Type / to search
Code | Issues 10 | Pull requests 41 | Actions | Projects | Security | Insights

This repository was archived by the owner on Sep 28, 2023. It is now read-only.

orchestrate-with-kubernetes Public archive

Watch 26 | Fork 320 | Star 515
master | 1 Branch | 0 Tags | Go to file | Code

askcarter Merge pull request #2 from IanLewis/blue-green 6f5f3d1 · 8 years ago 9 Commits

kubernetes	Updated to fix blue-green deployment	8 years ago
labs	Initial import	9 years ago
CONTRIBUTING.md	Initial import	9 years ago
LICENSE	Initial import	9 years ago
README.md	Updated README.md	9 years ago
cleanup.sh	Initial import	9 years ago

About

Orchestrating the Cloud with Kubernetes

codelabs.developers.google.com/codela...

Readme
Apache-2.0 license
Activity
Custom properties
515 stars
26 watching
320 forks
Report repository

Releases

No releases published

Packages

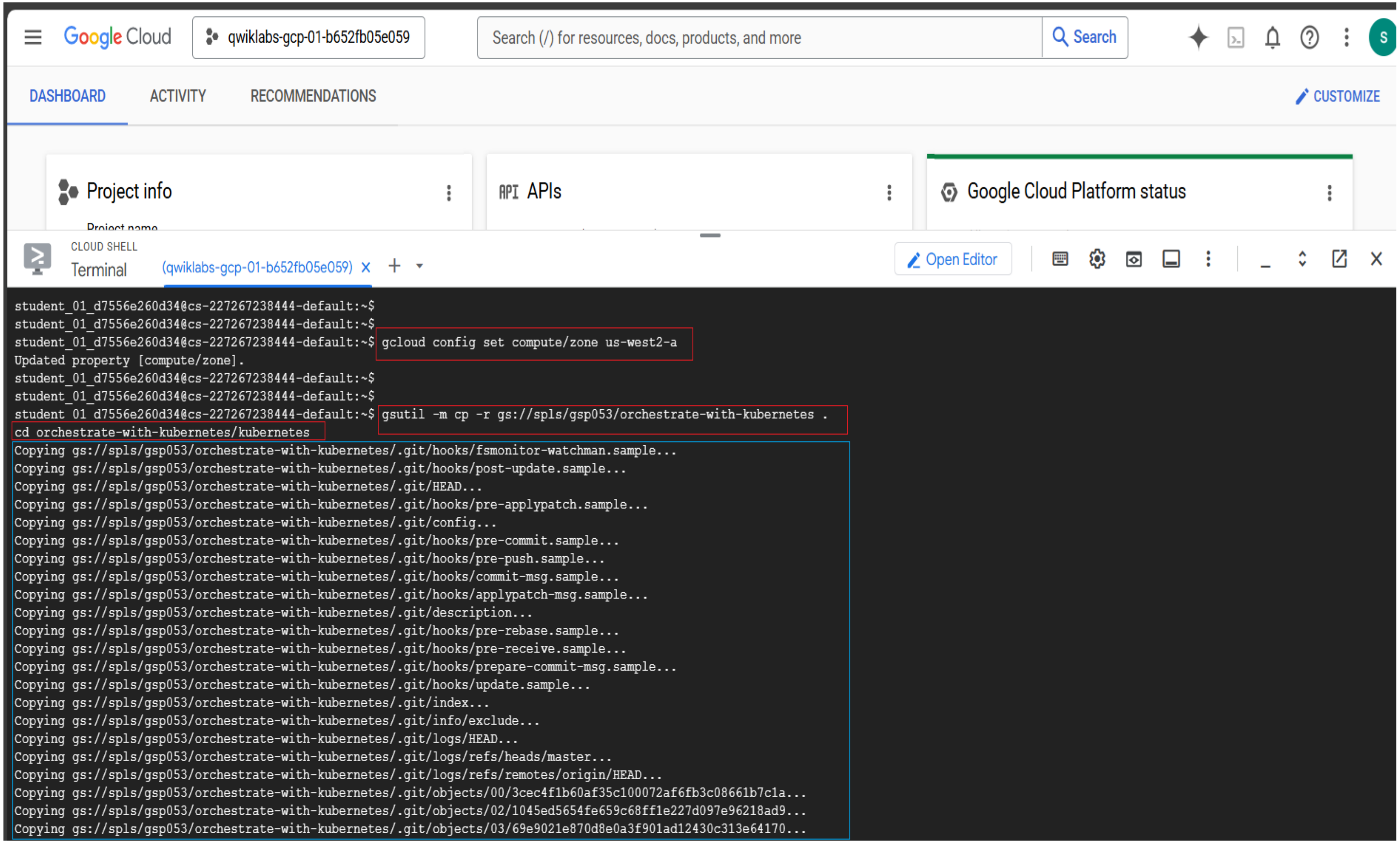
No packages published

README | Apache-2.0 license

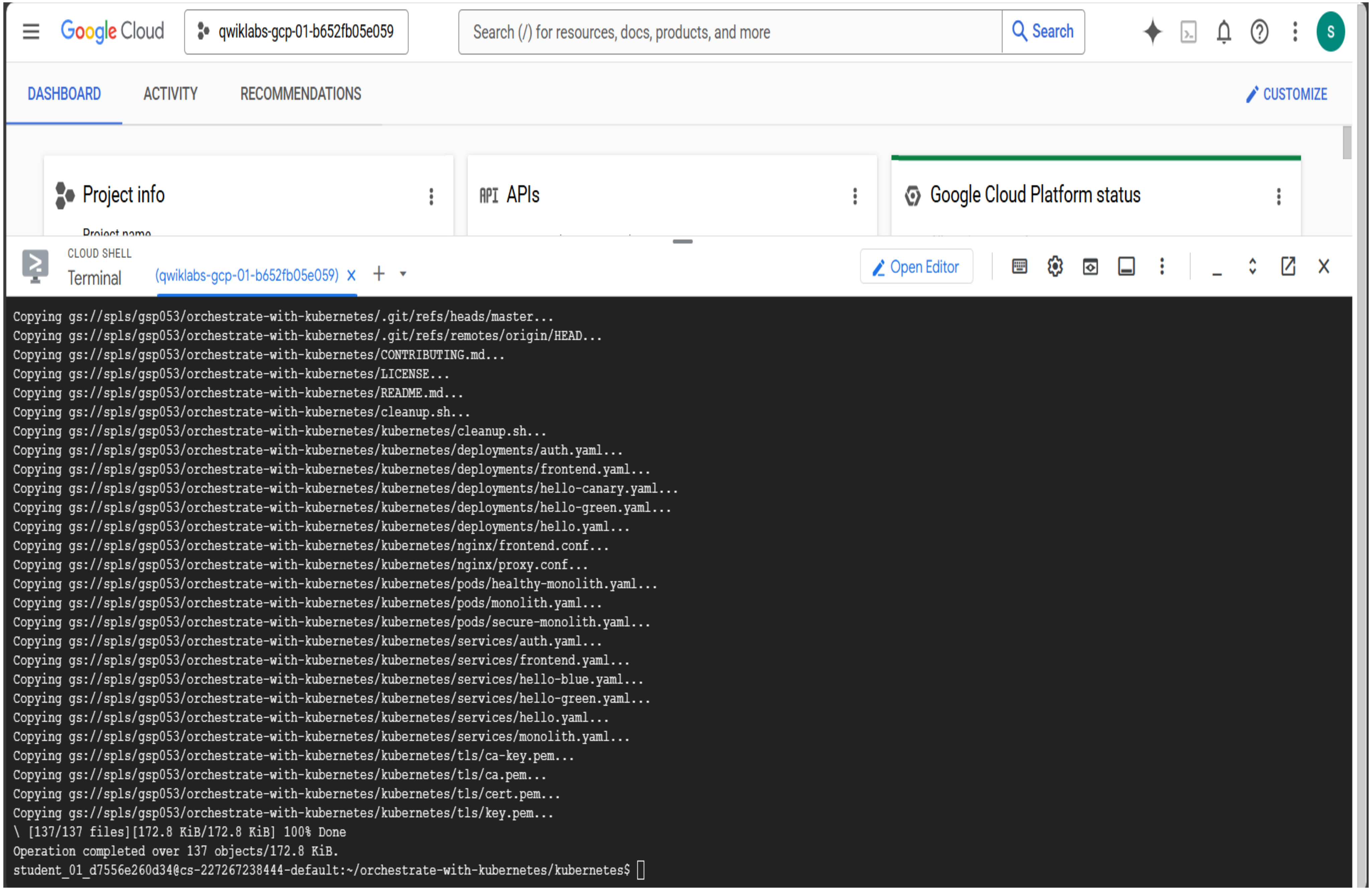
Orchestrating the Cloud with Kuberenetes

In this Codelab you will learn how to:

Then, change directory   cd orchestrate-with-kubernetes/kubernetes


Google Cloud
qwiklabs-gcp-01-b652fb05e059
Search (/) for resources, docs, products, and more
Search
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
Project info
APIs
Google Cloud Platform status
CLOUD SHELL
Terminal
(qwiklabs-gcp-01-b652fb05e059)
Open Editor

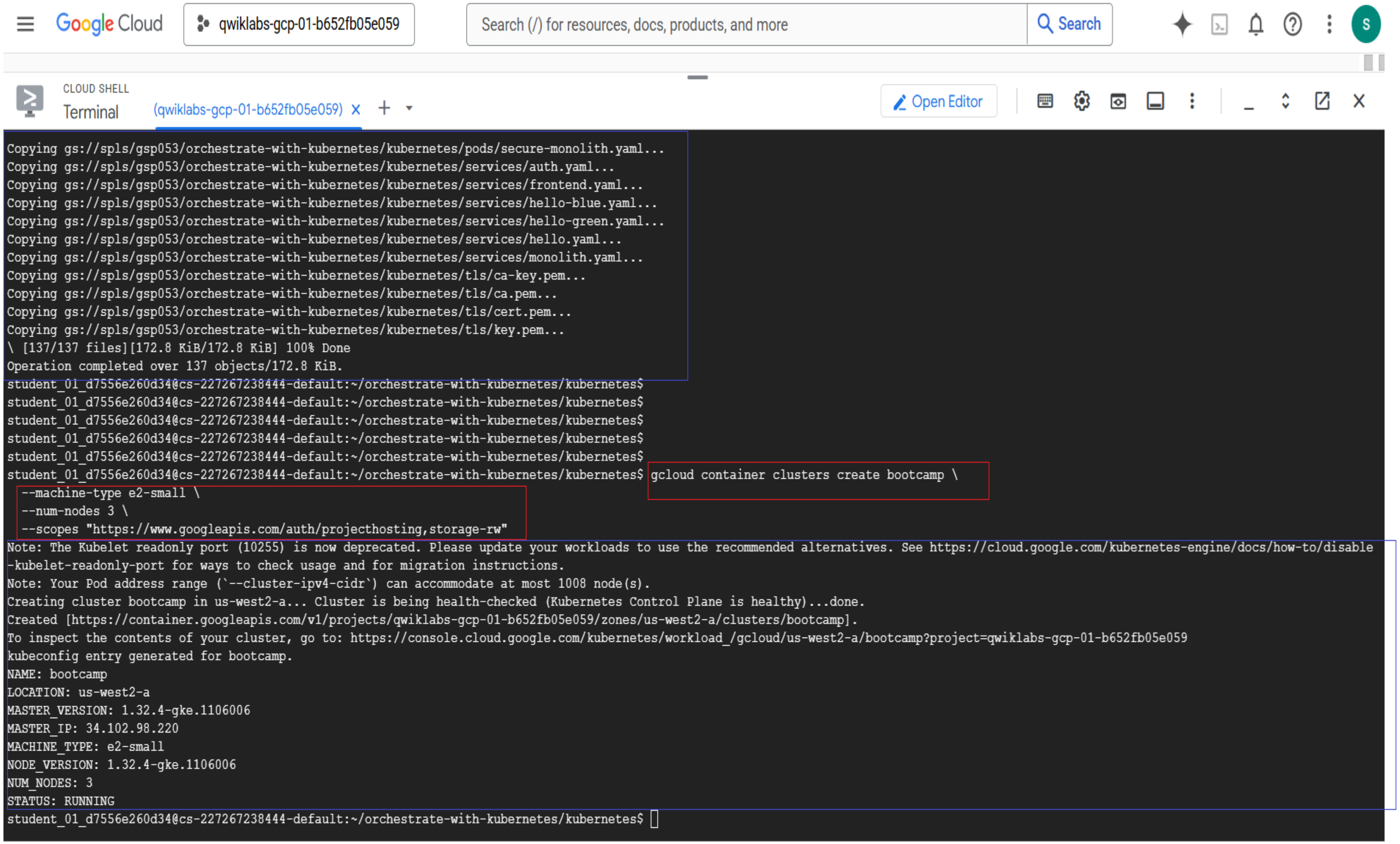
```
student_01_d7556e260d34@cs-227267238444-default:~$
student_01_d7556e260d34@cs-227267238444-default:~$
student_01_d7556e260d34@cs-227267238444-default:~$ gcloud config set compute/zone us-west2-a
Updated property [compute/zone].
student_01_d7556e260d34@cs-227267238444-default:~$
student_01_d7556e260d34@cs-227267238444-default:~$
student_01_d7556e260d34@cs-227267238444-default:~$ gsutil -m cp -r gs://spls/gsp053/orchestrate-with-kubernetes .
cd orchestrate-with-kubernetes/kubernetes
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/fsmonitor-watchman.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/post-update.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/HEAD...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/pre-applypatch.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/config...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/pre-commit.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/pre-push.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/commit-msg.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/applypatch-msg.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/description...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/pre-rebase.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/pre-receive.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/prepare-commit-msg.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/hooks/update.sample...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/index...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/info/exclude...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/logs/HEAD...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/logs/refs/heads/master...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/logs/refs/remotes/origin/HEAD...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/objects/00/3cec4f1b60af35c100072af6fb3c08661b7c1a...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/objects/02/1045ed5654fe659c68ff1e227d097e96218ad9...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/objects/03/69e9021e870d8e0a3f901ad12430c313e64170...
```


Google Cloud
qwiklabs-gcp-01-b652fb05e059
Search (/) for resources, docs, products, and more
Search
DASHBOARD
ACTIVITY
RECOMMENDATIONS
CUSTOMIZE
Project info
Project name
APIs
Google Cloud Platform status
CLOUD SHELL
Terminal
(qwiklabs-gcp-01-b652fb05e059)
Open Editor

```
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/refs/heads/master...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/.git/refs/remotes/origin/HEAD...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/CONTRIBUTING.md...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/LICENSE...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/README.md...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/cleanup.sh...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/cleanup.sh...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/deployments/auth.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/deployments/frontend.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/deployments/hello-canary.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/deployments/hello-green.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/deployments/hello.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/nginx/frontend.conf...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/nginx/proxy.conf...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/pods/healthy-monolith.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/pods/monolith.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/pods/secure-monolith.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/auth.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/frontend.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/hello-blue.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/hello-green.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/hello.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/monolith.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/tls/ca-key.pem...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/tls/ca.pem...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/tls/cert.pem...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/tls/key.pem...
\ [137/137 files][172.8 KiB/172.8 KiB] 100% Done
Operation completed over 137 objects/172.8 KiB.
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Create a cluster with 3 nodes:

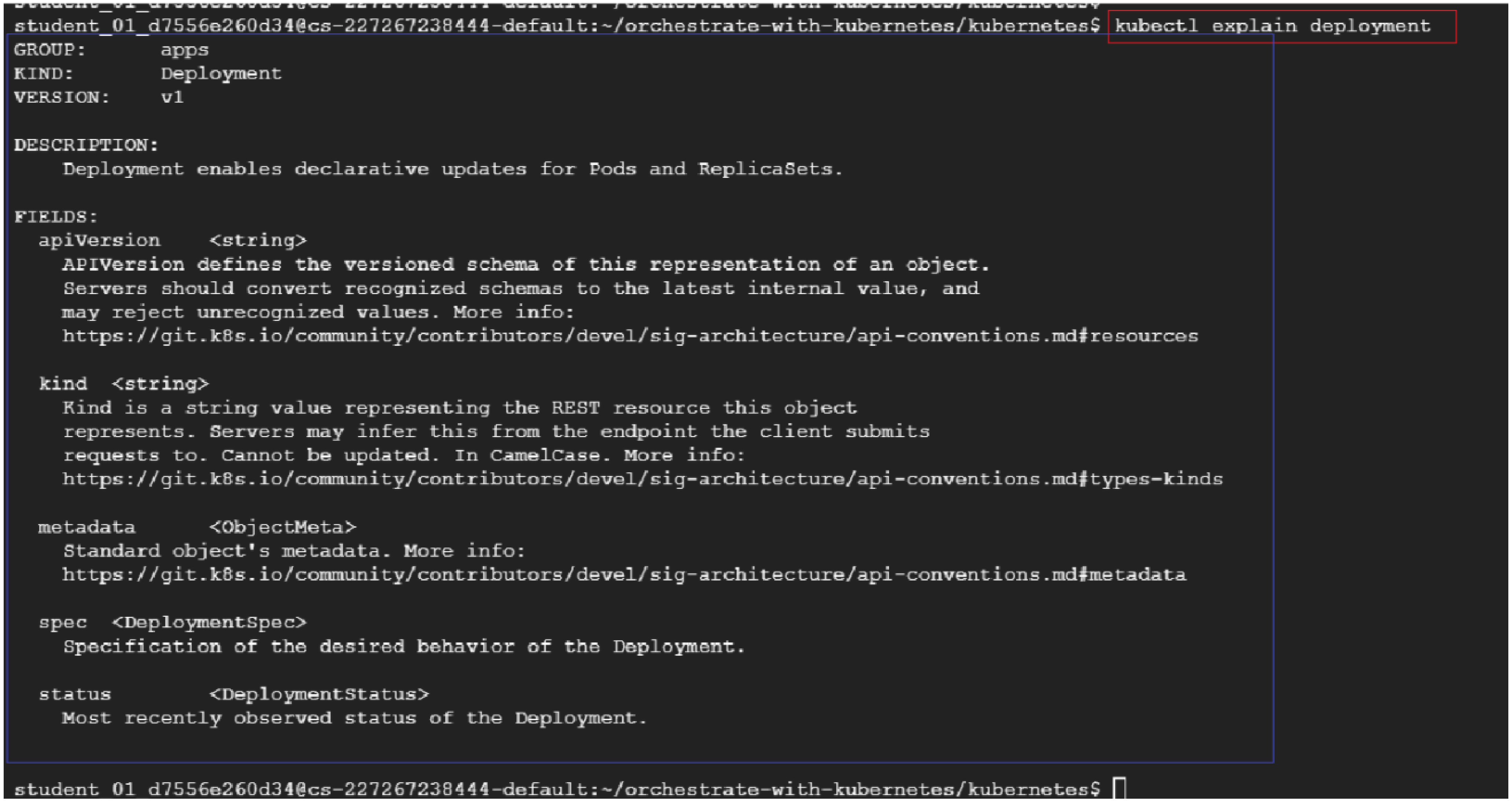
```
gcloud container clusters create bootcamp \
  --machine-type e2-small \
  --num-nodes 3 \
  --scopes
"https://www.googleapis.com/auth/projecthosting,storage-rw"
```


Google Cloud
qwiklabs-gcp-01-b652fb05e059
Search (/) for resources, docs, products, and more
Search
CLOUD SHELL
Terminal
(qwiklabs-gcp-01-b652fb05e059)
Open Editor

```
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/pods/secure-monolith.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/auth.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/frontend.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/hello-blue.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/hello-green.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/hello.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/services/monolith.yaml...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/tls/ca-key.pem...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/tls/ca.pem...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/tls/cert.pem...
Copying gs://spls/gsp053/orchestrate-with-kubernetes/kubernetes/tls/key.pem...
\ [137/137 files][172.8 KiB/172.8 KiB] 100% Done
Operation completed over 137 objects/172.8 KiB.
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ gcloud container clusters create bootcamp \
  --machine-type e2-small \
  --num-nodes 3 \
  --scopes "https://www.googleapis.com/auth/projecthosting,storage-rw"
Note: The Kubelet readonly port (10255) is now deprecated. Please update your workloads to use the recommended alternatives. See https://cloud.google.com/kubernetes-engine/docs/how-to/disable
-kubelet-readonly-port for ways to check usage and for migration instructions.
Note: Your Pod address range (`--cluster-ipv4-cidr`) can accommodate at most 1008 node(s).
Creating cluster bootcamp in us-west2-a... Cluster is being health-checked (Kubernetes Control Plane is healthy)...done.
Created [https://container.googleapis.com/v1/projects/qwiklabs-gcp-01-b652fb05e059/zones/us-west2-a/clusters/bootcamp].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload_/gcloud/us-west2-a/bootcamp?project=qwiklabs-gcp-01-b652fb05e059
kubeconfig entry generated for bootcamp.
NAME: bootcamp
LOCATION: us-west2-a
MASTER_VERSION: 1.32.4-gke.1106006
MASTER_IP: 34.102.98.220
MACHINE_TYPE: e2-small
NODE_VERSION: 1.32.4-gke.1106006
NUM_NODES: 3
STATUS: RUNNING
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Get the information about deployment object

The explain command in kubectl can tell us about the deployment object: kubectl explain deployment



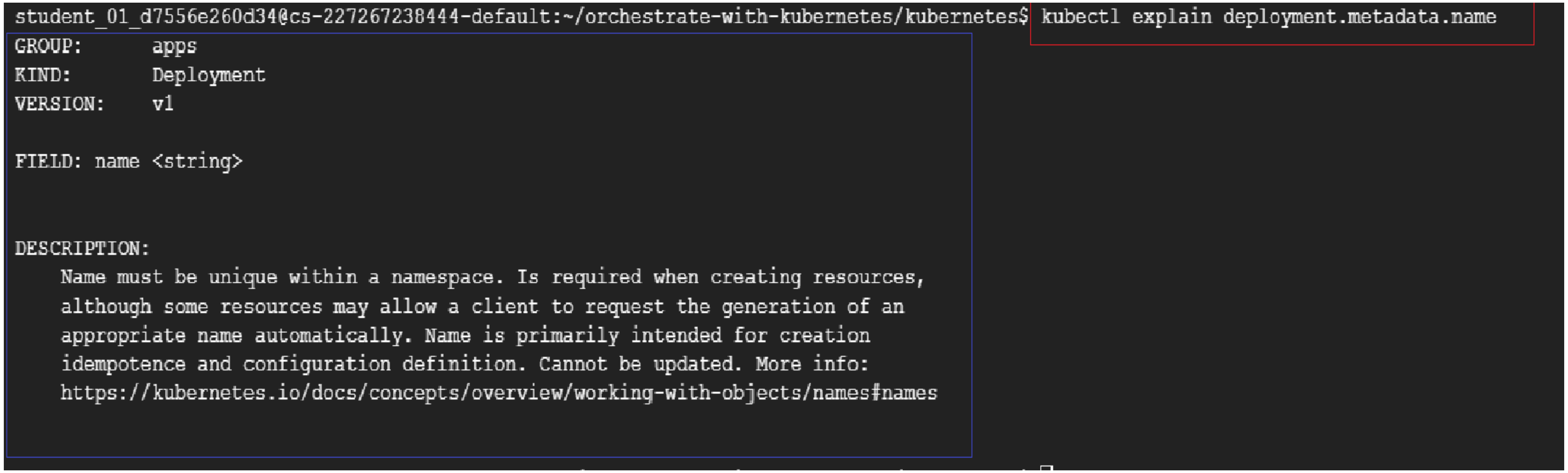
```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl explain deployment
GROUP:      apps
KIND:       Deployment
VERSION:    v1

DESCRIPTION:
    Deployment enables declarative updates for Pods and ReplicaSets.

FIELDS:
  apiVersion    <string>
    APIVersion defines the versioned schema of this representation of an object.
    Servers should convert recognized schemas to the latest internal value, and
    may reject unrecognized values. More info:
    https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources

  kind  <string>
    Kind is a string value representing the REST resource this object
    represents. Servers may infer this from the endpoint the client submits
    requests to. Cannot be updated. In CamelCase. More info:
    https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds

  metadata      <ObjectMeta>
    Standard object's metadata. More info:
    https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

  spec  <DeploymentSpec>
    Specification of the desired behavior of the Deployment.

  status        <DeploymentStatus>
    Most recently observed status of the Deployment.


student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ 
```

You can also see all of the fields using the --recursive option:

kubectl explain deployment –recursive

You can use explain command as you go through the lab to help you understand the structure of a deployment object and understand what the individual fields do:

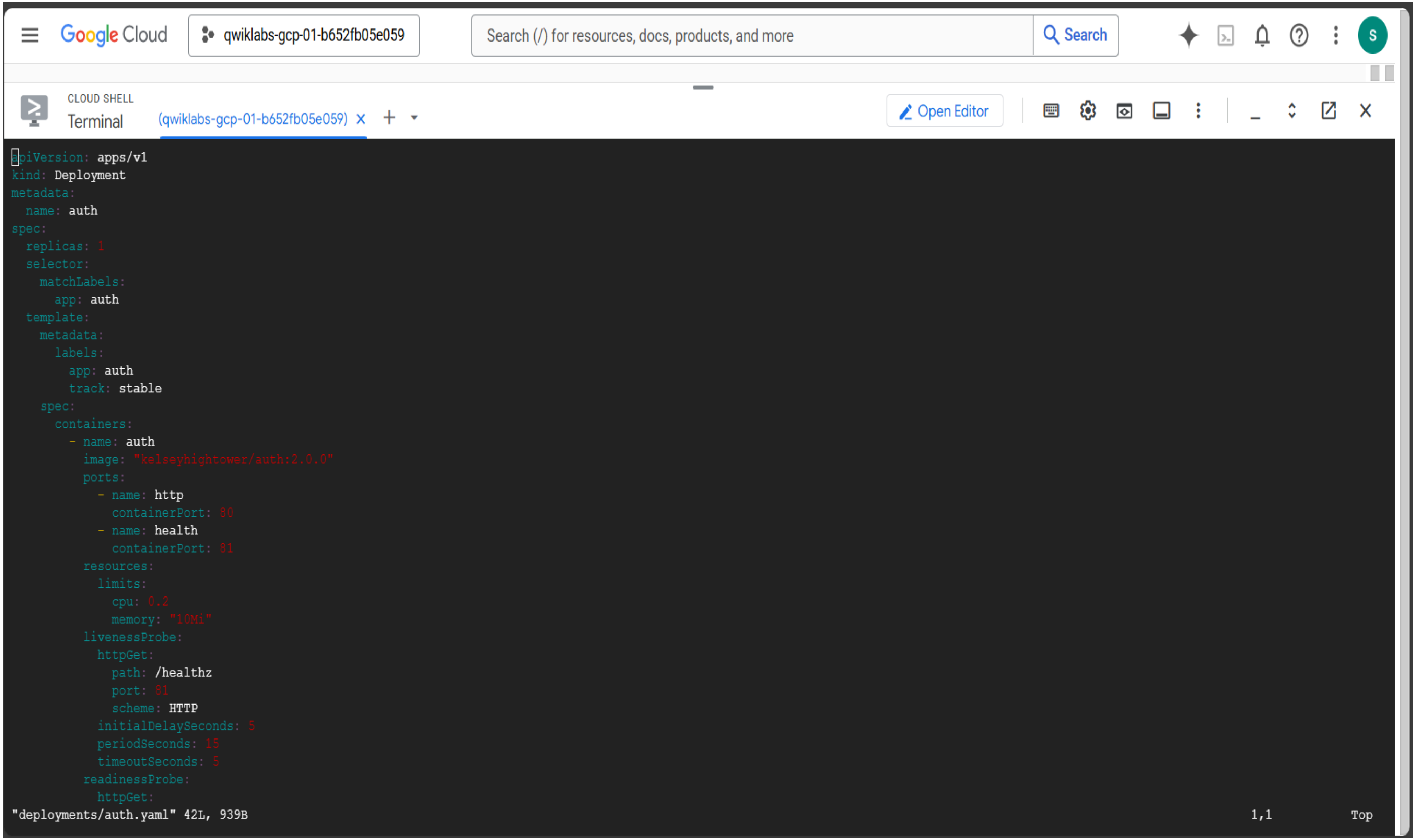
kubectl explain deployment.metadata.name



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl explain deployment.metadata.name
GROUP:      apps
KIND:       Deployment
VERSION:    v1

FIELD: name <string>


DESCRIPTION:
    Name must be unique within a namespace. Is required when creating resources,
    although some resources may allow a client to request the generation of an
    appropriate name automatically. Name is primarily intended for creation
    idempotence and configuration definition. Cannot be updated. More info:
    https://kubernetes.io/docs/concepts/overview/working-with-objects/names#names
```

Create a deployment

Update the deployments/auth.yaml configuration file: vi deployments/auth.yaml

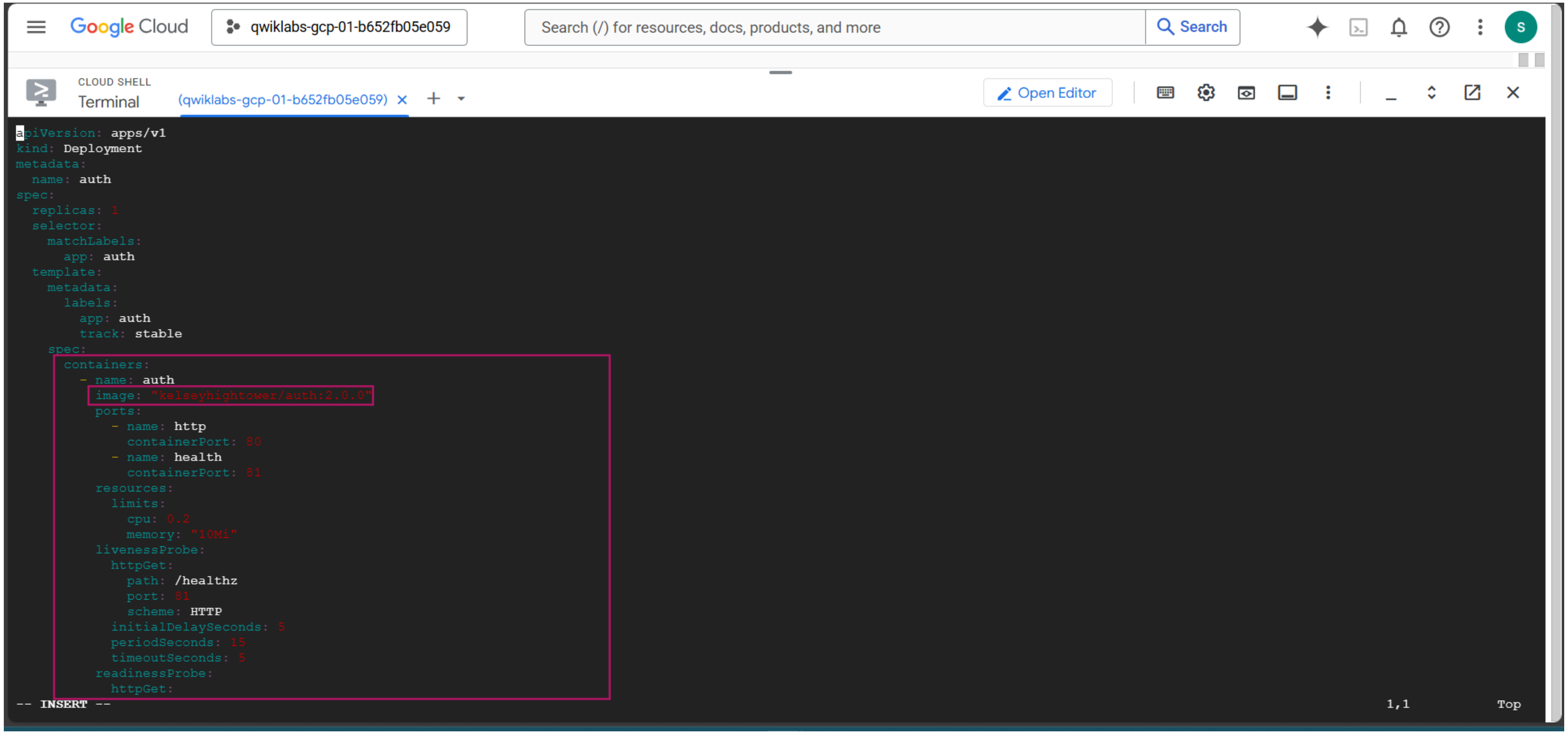

Google Cloud
qwiklabs-gcp-01-b652fb05e059
Search (/) for resources, docs, products, and more
Search
CLOUD SHELL
Terminal
(qwiklabs-gcp-01-b652fb05e059)
Open Editor

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: auth
spec:
  replicas: 1
  selector:
    matchLabels:
      app: auth
  template:
    metadata:
      labels:
        app: auth
        track: stable
    spec:
      containers:
        - name: auth
          image: "kelseyhightower/auth:2.0.0"
          ports:
            - name: http
              containerPort: 80
            - name: health
              containerPort: 81
          resources:
            limits:
              cpu: 0.2
              memory: "10Mi"
          livenessProbe:
            httpGet:
              path: /healthz
              port: 81
              scheme: HTTP
            initialDelaySeconds: 5
            periodSeconds: 15
            timeoutSeconds: 5
          readinessProbe:
            httpGet:
"deployments/auth.yaml" 42L, 939B                                                          1,1           Top
```

Start the editor:

press "i" for Insert mode in vi editor

Change the image in the containers section of the deployment to the following:


Google Cloud
qwiklabs-gcp-01-b652fb05e059
Search (/) for resources, docs, products, and more
Search
CLOUD SHELL
Terminal
(qwiklabs-gcp-01-b652fb05e059)
Open Editor

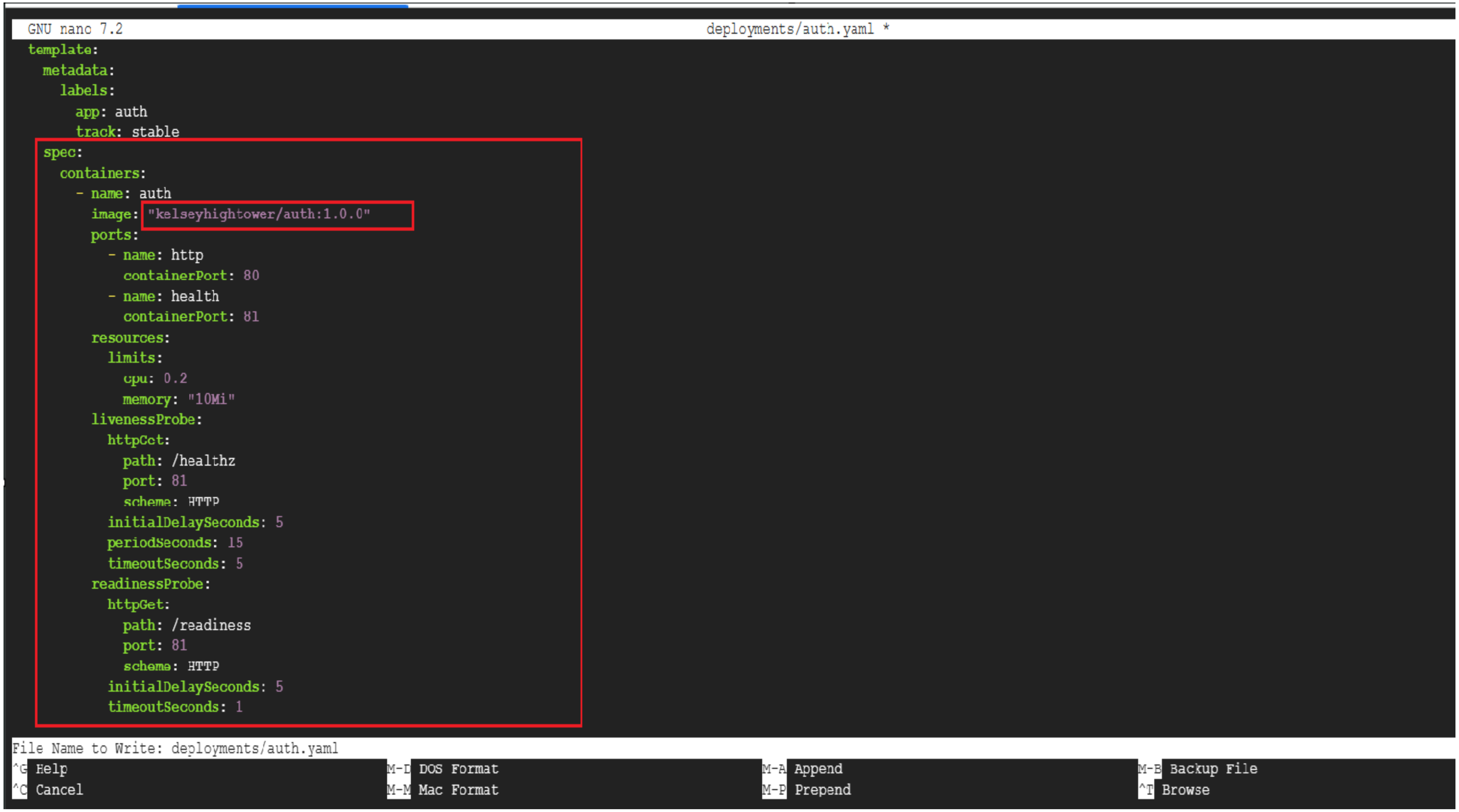
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: auth
spec:
  replicas: 1
  selector:
    matchLabels:
      app: auth
  template:
    metadata:
      labels:
        app: auth
        track: stable
    spec:
      containers:
        - name: auth
          image: "kelseyhightower/auth:2.0.0"
          ports:
            - name: http
              containerPort: 80
            - name: health
              containerPort: 81
          resources:
            limits:
              cpu: 0.2
              memory: "10Mi"
          livenessProbe:
            httpGet:
              path: /healthz
              port: 81
              scheme: HTTP
            initialDelaySeconds: 5
            periodSeconds: 15
            timeoutSeconds: 5
          readinessProbe:
            httpGet:
-- INSERT --                                                    1,1           Top
```

Save the auth.yaml file: press <Esc> then type:

:wq

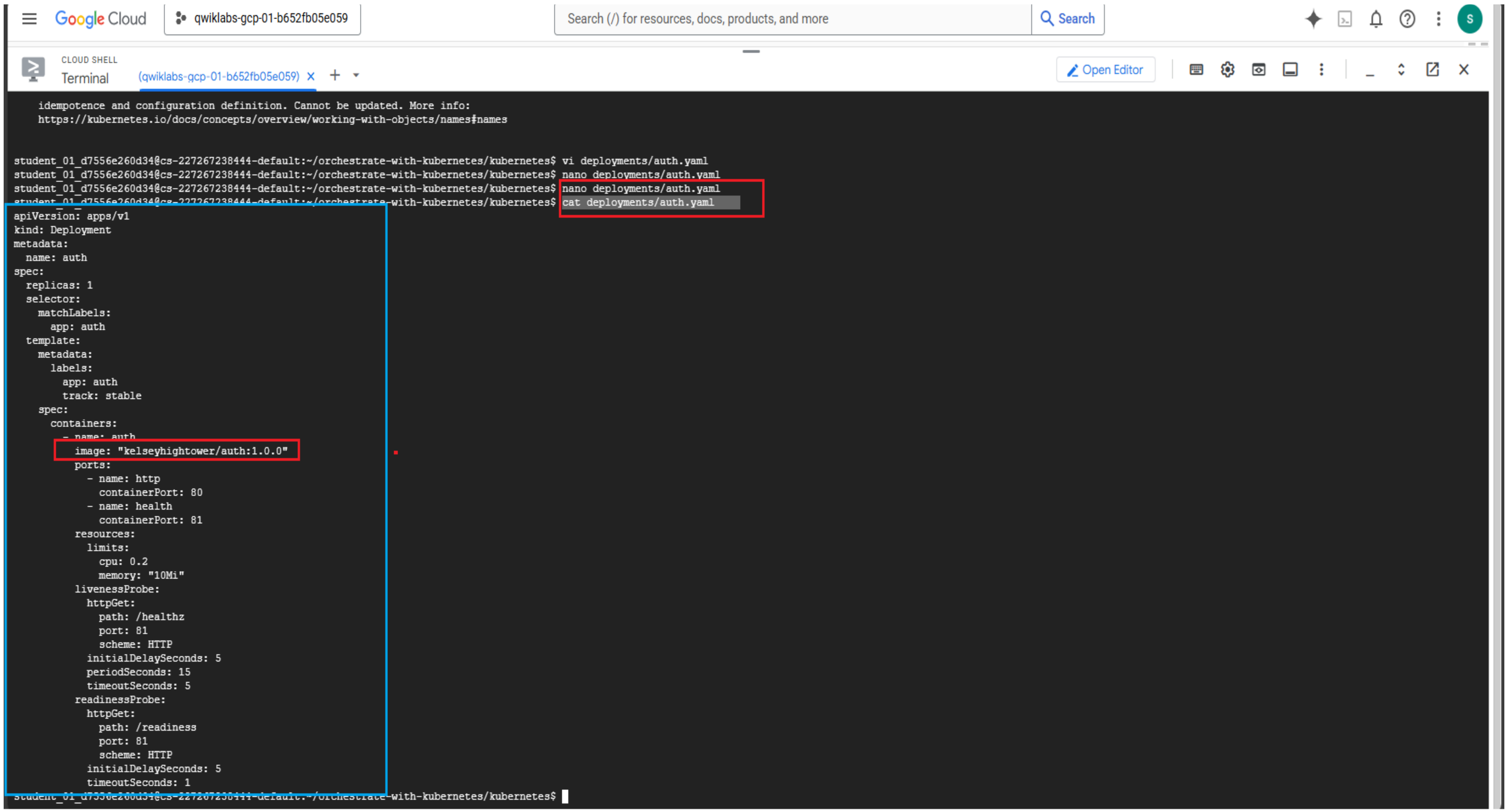
Press <Enter>. Now create a simple deployment.

I used nano editor to make changes:



```
  GNU nano 7.2                                                                    deployments/auth.yaml *
  template:
    metadata:
      labels:
        app: auth
        track: stable
    spec:
      containers:
        - name: auth
          image: "kelseyhightower/auth:1.0.0"
          ports:
            - name: http
              containerPort: 80
            - name: health
              containerPort: 81
          resources:
            limits:
              cpu: 0.2
              memory: "10Mi"
          livenessProbe:
            httpGet:
              path: /healthz
              port: 81
              scheme: HTTP
            initialDelaySeconds: 5
            periodSeconds: 15
            timeoutSeconds: 5
          readinessProbe:
            httpGet:
              path: /readiness
              port: 81
              scheme: HTTP
            initialDelaySeconds: 5
            timeoutSeconds: 1
File Name to Write: deployments/auth.yaml
^G Help                  M-D DOS Format           M-A Append               M-B Backup File
^C Cancel                M-M Mac Format           M-P Prepend              ^T Browse
```

Examine the deployment configuration file: cat deployments/auth.yaml


Google Cloud
qwiklabs-gcp-01-b652fb05e059
Search (/) for resources, docs, products, and more
Search
CLOUD SHELL
Terminal
(qwiklabs-gcp-01-b652fb05e059)
Open Editor

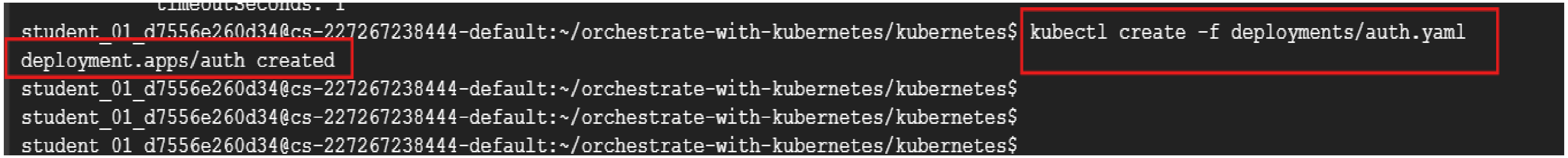
```
    idempotence and configuration definition. Cannot be updated. More info:
    https://kubernetes.io/docs/concepts/overview/working-with-objects/names#names


student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ vi deployments/auth.yaml
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ nano deployments/auth.yaml
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ nano deployments/auth.yaml
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ cat deployments/auth.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: auth
spec:
  replicas: 1
  selector:
    matchLabels:
      app: auth
  template:
    metadata:
      labels:
        app: auth
        track: stable
    spec:
      containers:
        - name: auth
          image: "kelseyhightower/auth:1.0.0"
          ports:
            - name: http
              containerPort: 80
            - name: health
              containerPort: 81
          resources:
            limits:
              cpu: 0.2
              memory: "10Mi"
          livenessProbe:
            httpGet:
              path: /healthz
              port: 81
              scheme: HTTP
            initialDelaySeconds: 5
            periodSeconds: 15
            timeoutSeconds: 5
          readinessProbe:
            httpGet:
              path: /readiness
              port: 81
              scheme: HTTP
            initialDelaySeconds: 5
            timeoutSeconds: 1
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

The deployment is creating one replica and it's using version 1.0.0 of the auth container.

Kubectl create command is used to create the auth deployment.

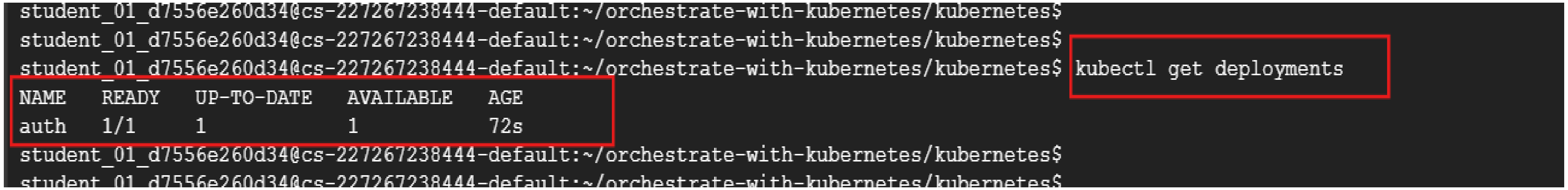
It makes 1 pod that conforms to the data in the deployment manifest. You can scale the number of Pods by changing the number specified in the replicas field.

Create your deployment object using kubectl create: kubectl create -f deployments/auth.yaml



```
            timeoutSeconds: 1
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl create -f deployments/auth.yaml
deployment.apps/auth created
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

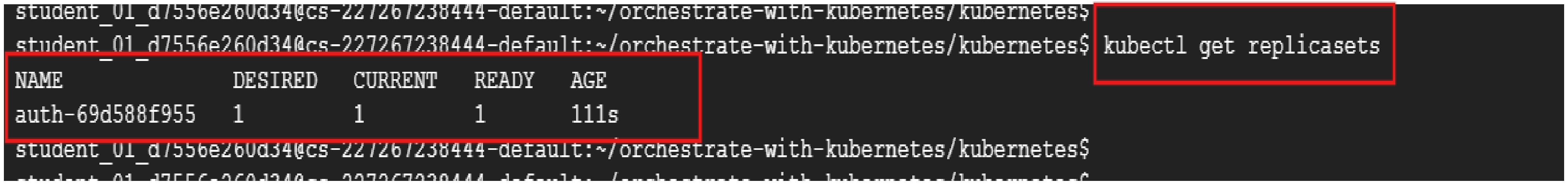
Verify the deployment creation: kubectl get deployments



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl get deployments
NAME   READY   UP-TO-DATE   AVAILABLE   AGE
auth   1/1     1            1           72s
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Once the deployment is created, Kubernetes will create a ReplicaSet for the deployment.

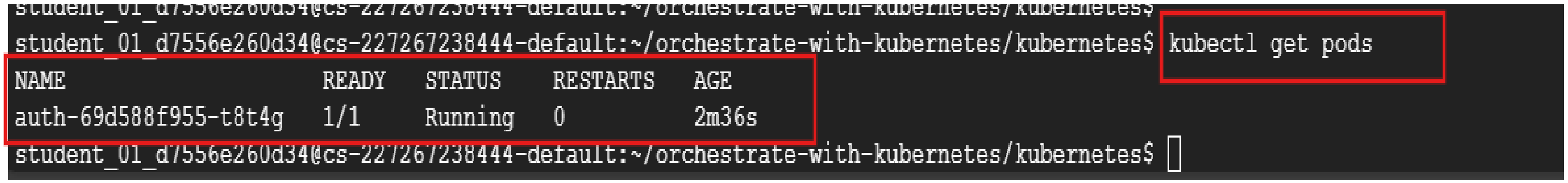
Verify that a ReplicaSet was created for the deployment: kubectl get replicasets



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl get replicasets
NAME              DESIRED   CURRENT   READY   AGE
auth-69d588f955   1         1         1       111s
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

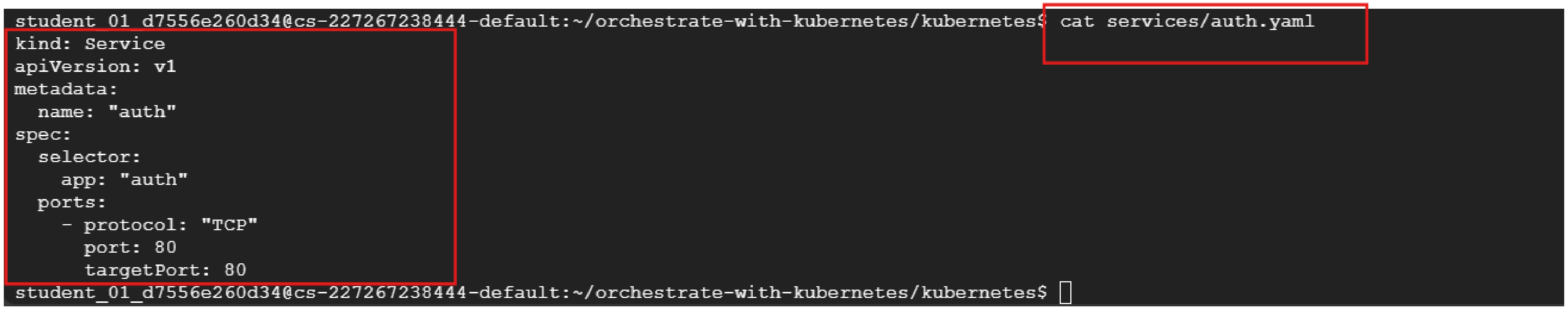
ReplicaSet is named as auth-xxxxxxx. The single Pod is created by the Kubernetes when the ReplicaSet is created.

View the Pods: kubectl get pods



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl get pods
NAME                    READY   STATUS    RESTARTS   AGE
auth-69d588f955-t8t4g   1/1     Running   0          2m36s
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Use the kubectl create command to create the auth service for the auth deployment: kubectl create -f services/auth.yaml

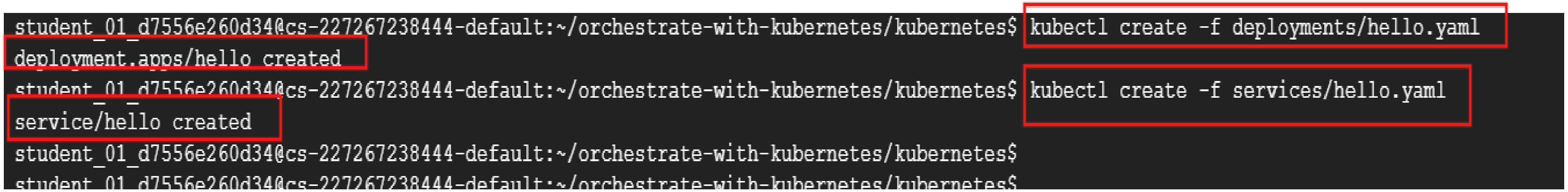


```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ cat services/auth.yaml
kind: Service
apiVersion: v1
metadata:
  name: "auth"
spec:
  selector:
    app: "auth"
  ports:
    - protocol: "TCP"
      port: 80
      targetPort: 80
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Create and expose the hello deployment:

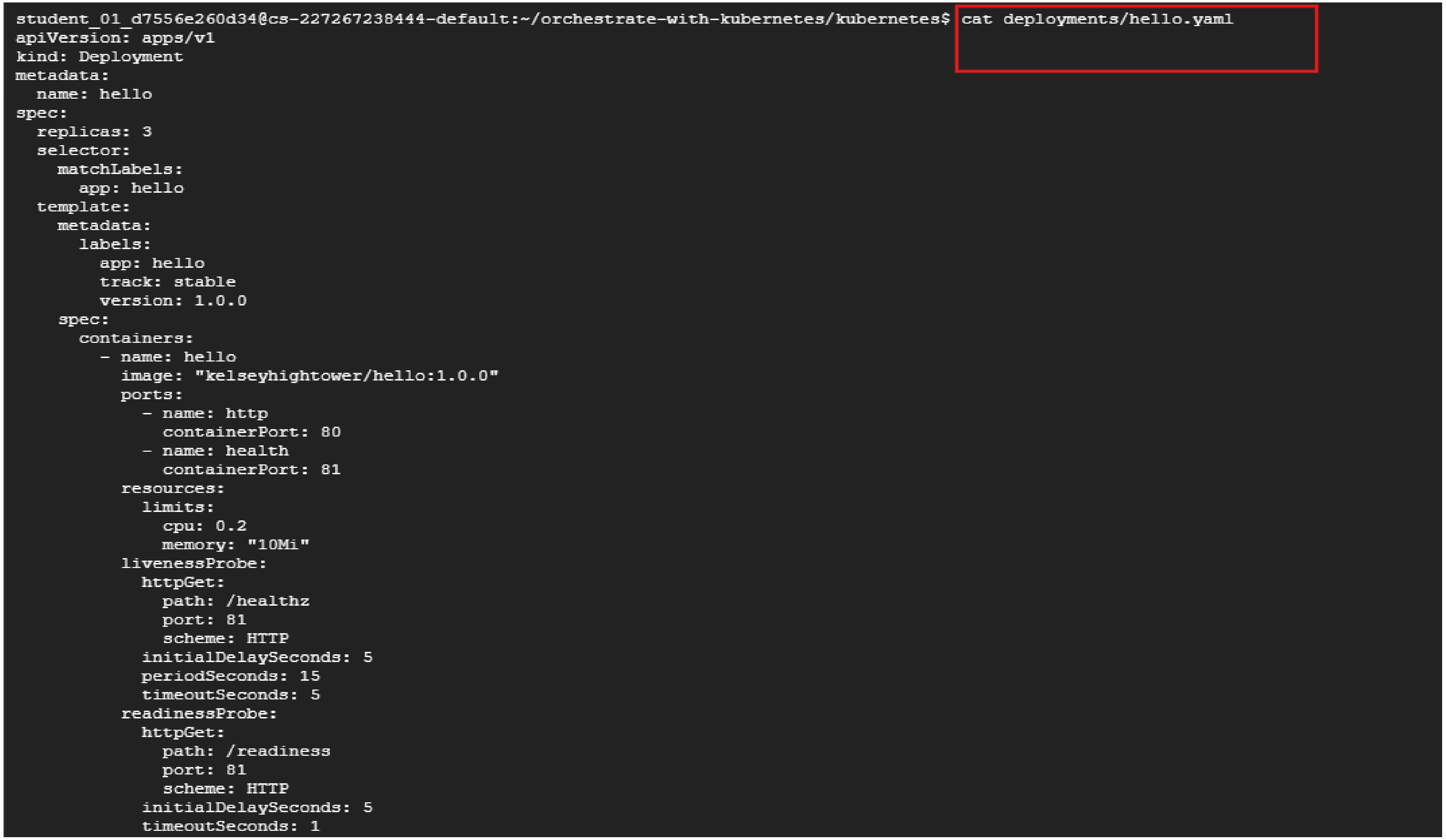
kubectl create -f deployments/hello.yaml

kubectl create -f services/hello.yaml



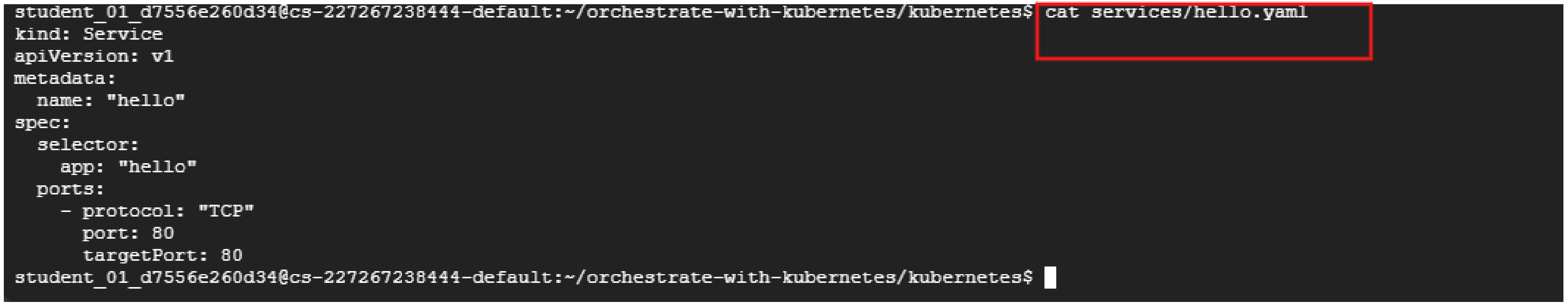
```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl create -f deployments/hello.yaml
deployment.apps/hello created
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl create -f services/hello.yaml
service/hello created
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

cat deployments/hello.yaml



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ cat deployments/hello.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello
spec:
  replicas: 3
  selector:
    matchLabels:
      app: hello
  template:
    metadata:
      labels:
        app: hello
        track: stable
        version: 1.0.0
    spec:
      containers:
        - name: hello
          image: "kelseyhightower/hello:1.0.0"
          ports:
            - name: http
              containerPort: 80
            - name: health
              containerPort: 81
          resources:
            limits:
              cpu: 0.2
              memory: "10Mi"
          livenessProbe:
            httpGet:
              path: /healthz
              port: 81
              scheme: HTTP
            initialDelaySeconds: 5
            periodSeconds: 15
            timeoutSeconds: 5
          readinessProbe:
            httpGet:
              path: /readiness
              port: 81
              scheme: HTTP
            initialDelaySeconds: 5
            timeoutSeconds: 1
```

cat services/hello.yaml



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ cat services/hello.yaml
kind: Service
apiVersion: v1
metadata:
  name: "hello"
spec:
  selector:
    app: "hello"
  ports:
    - protocol: "TCP"
      port: 80
      targetPort: 80
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

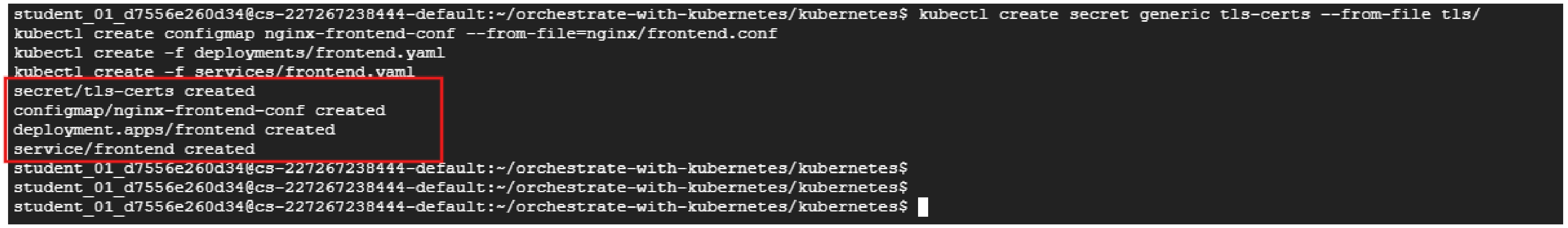
Create and expose the frontend deployment:

kubectl create secret generic tls-certs --from-file tls/

kubectl create configmap nginx-frontend-conf --from-file=nginx/frontend.conf

kubectl create -f deployments/frontend.yaml

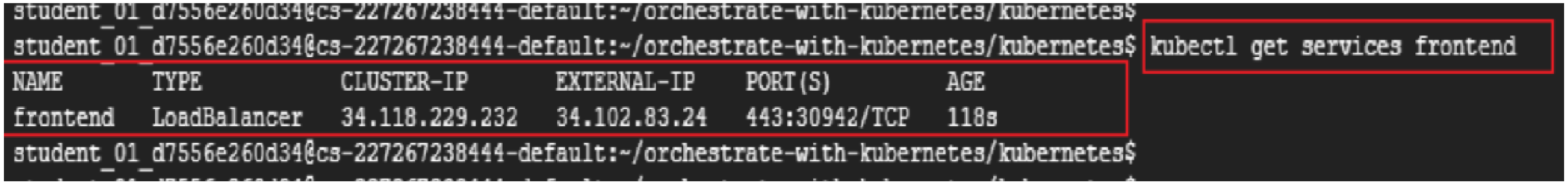
kubectl create -f services/frontend.yaml



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl create secret generic tls-certs --from-file tls/
kubectl create configmap nginx-frontend-conf --from-file=nginx/frontend.conf
kubectl create -f deployments/frontend.yaml
kubectl create -f services/frontend.yaml
secret/tls-certs created
configmap/nginx-frontend-conf created
deployment.apps/frontend created
service/frontend created
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Note: You created a ConfigMap for the frontend.

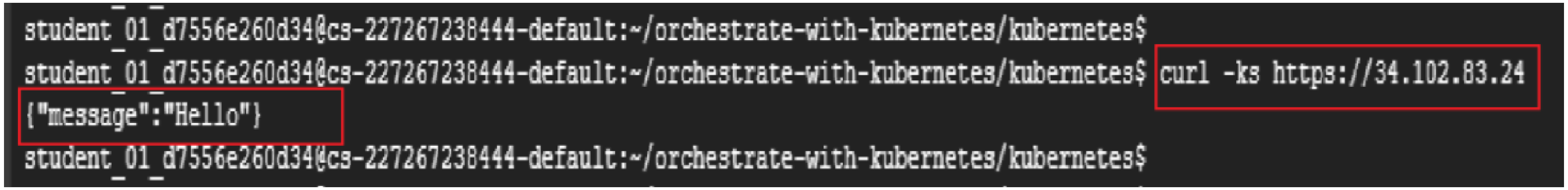
Interact with the frontend by grabbing its external IP and then curling to it: kubectl get services frontend



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl get services frontend
NAME       TYPE           CLUSTER-IP       EXTERNAL-IP    PORT(S)         AGE
frontend   LoadBalancer   34.118.229.232   34.102.83.24   443:30942/TCP   118s
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

curl -ks https://<EXTERNAL-IP>

curl -ks https://34.102.83.24

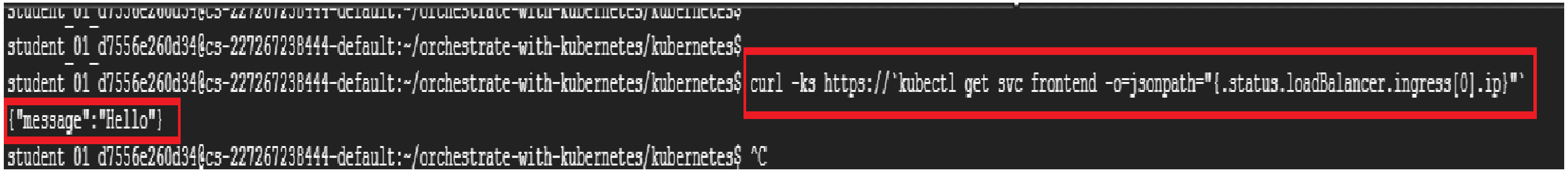


```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ curl -ks https://34.102.83.24
{"message":"Hello"}
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Use the output templating feature of kubectl to use curl as a one-liner:

curl -ks https://`kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"`

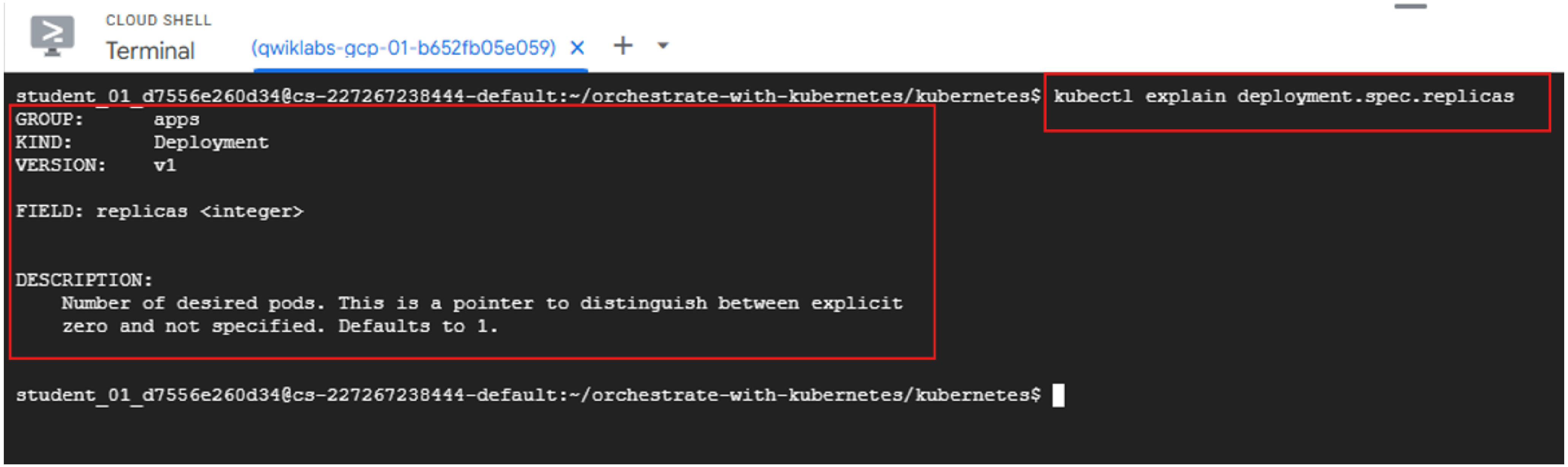
curl -ks https://`kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"`



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ curl -ks https://`kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"`
{"message":"Hello"}
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ ^C
```

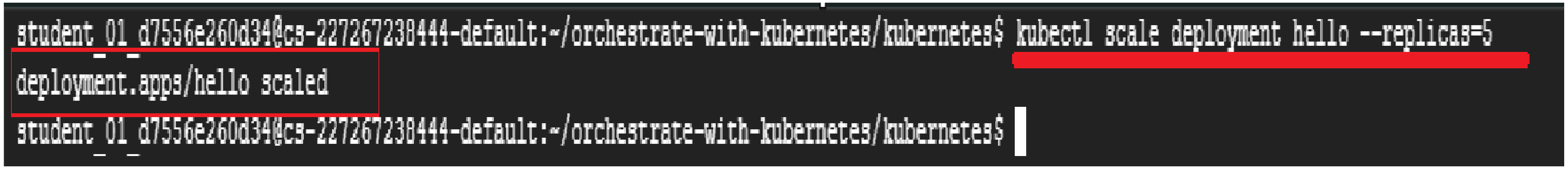
Scale a deployment

Scale the deployment by updating the spec.replicas field: kubectl explain deployment.spec.replicas



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl explain deployment.spec.replicas
GROUP:      apps
KIND:       Deployment
VERSION:    v1

FIELD: replicas <integer>


DESCRIPTION:
    Number of desired pods. This is a pointer to distinguish between explicit
    zero and not specified. Defaults to 1.

student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

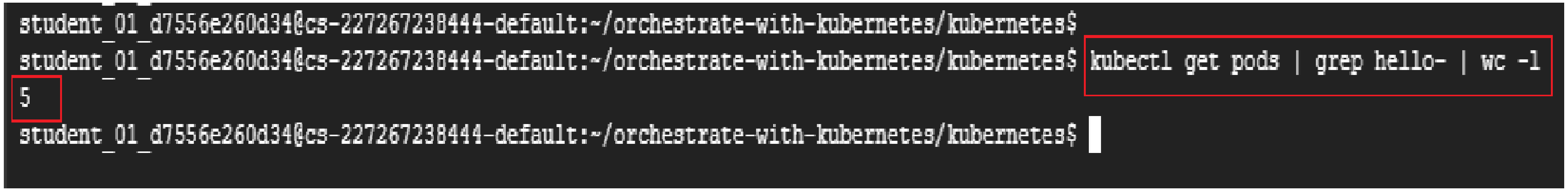
Update replicas field using the kubectl scale command: kubectl scale deployment hello --replicas=5



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl scale deployment hello --replicas=5
deployment.apps/hello scaled
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

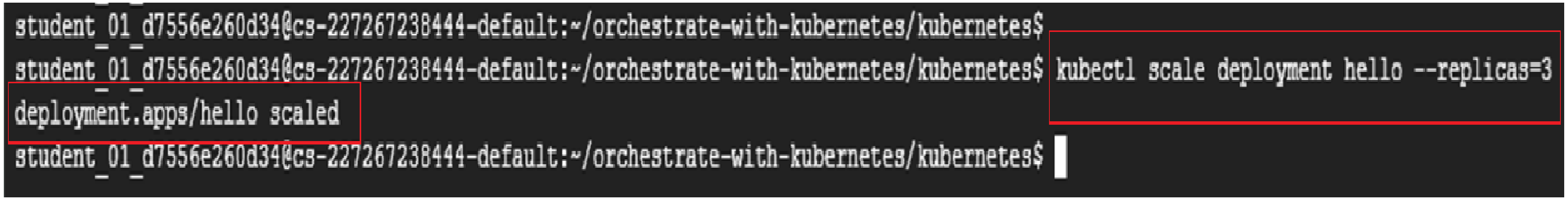
After the deployment is updated, Kubernetes will automatically update the associated ReplicaSet and start new Pods to make the total number of Pods equal 5.

Verify that there are now 5 hello Pods running: kubectl get pods | grep hello- | wc -l



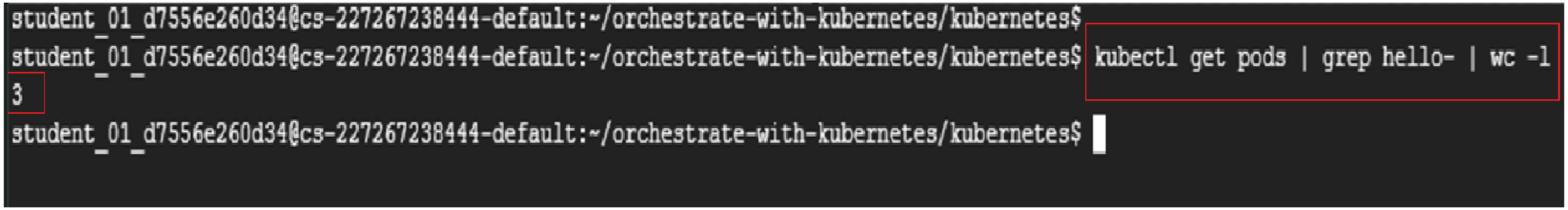
```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl get pods | grep hello- | wc -l
5
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Scale back the application: kubectl scale deployment hello --replicas=3



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl scale deployment hello --replicas=3
deployment.apps/hello scaled
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Verify the correct number of Pods: kubectl get pods | grep hello- | wc -l

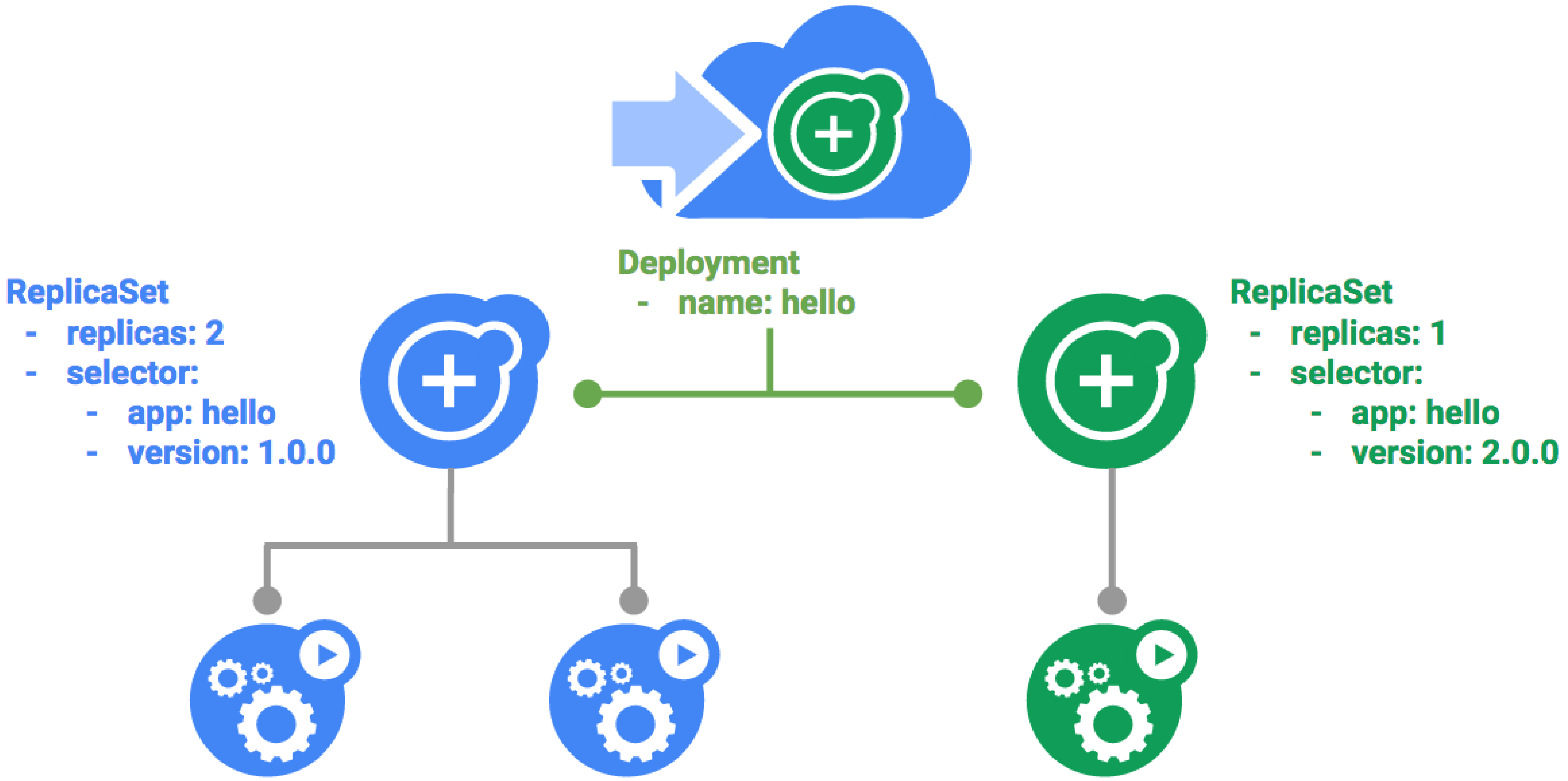


```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl get pods | grep hello- | wc -l
3
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Rolling update

Deployments support updating images to a new version through a rolling update mechanism.

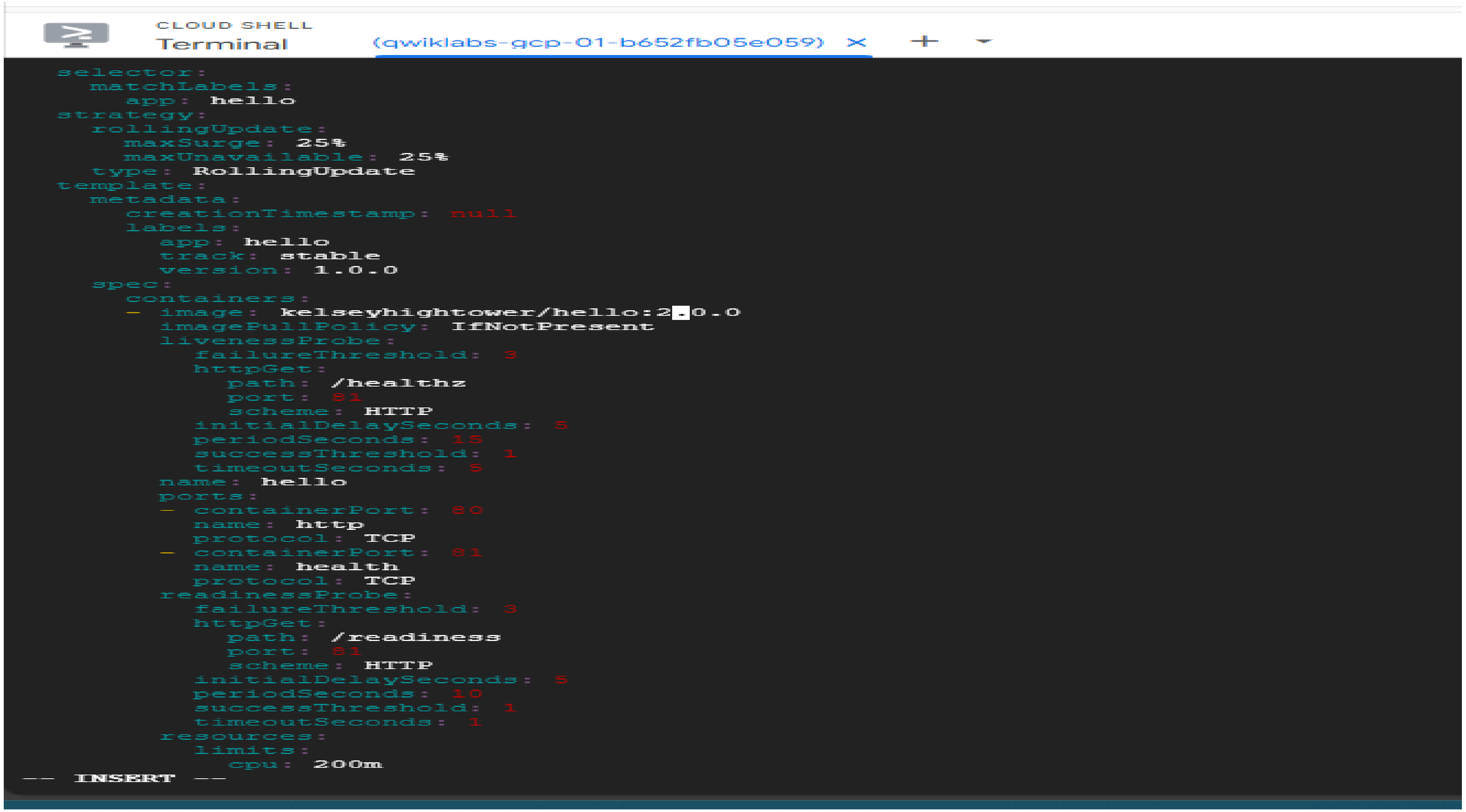
When a deployment is updated with a new version, it creates a new ReplicaSet and slowly increases the number of replicas in the new ReplicaSet as it decreases the replicas in the old ReplicaSet.



Trigger a rolling update

Update your deployment: kubectl edit deployment hello

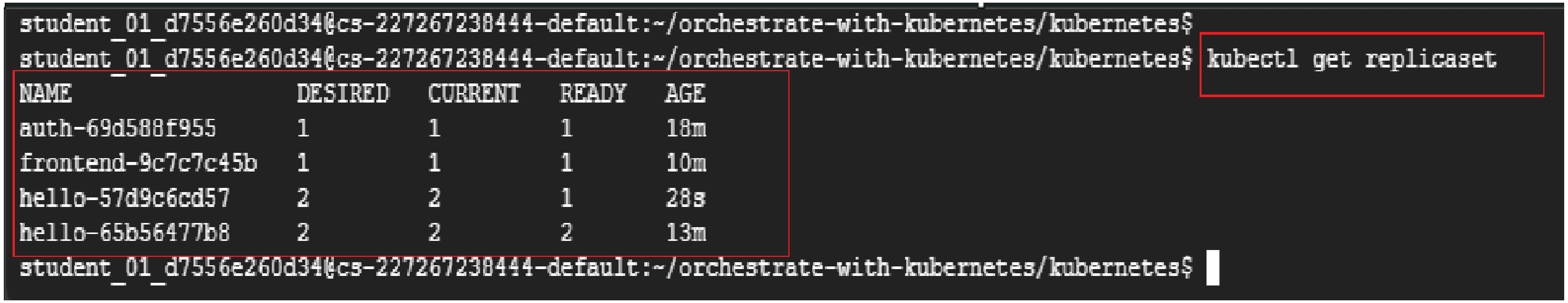
Change the image in the containers section of the deployment to the following:



```
  selector:
    matchLabels:
      app: hello
  strategy:
    rollingUpdate:
      maxSurge: 25%
      maxUnavailable: 25%
    type: RollingUpdate
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: hello
        track: stable
        version: 1.0.0
    spec:
      containers:
      - image: kelseyhightower/hello:2.0.0
        imagePullPolicy: IfNotPresent
        livenessProbe:
          failureThreshold: 3
          httpGet:
            path: /healthz
            port: 81
            scheme: HTTP
          initialDelaySeconds: 5
          periodSeconds: 15
          successThreshold: 1
          timeoutSeconds: 5
        name: hello
        ports:
        - containerPort: 80
          name: http
          protocol: TCP
        - containerPort: 81
          name: health
          protocol: TCP
        readinessProbe:
          failureThreshold: 3
          httpGet:
            path: /readiness
            port: 81
            scheme: HTTP
          initialDelaySeconds: 5
          periodSeconds: 10
          successThreshold: 1
          timeoutSeconds: 1
        resources:
          limits:
            cpu: 200m
-- INSERT --
```

Save and exit. The updated deployment will be saved to your cluster and Kubernetes will begin a rolling update.

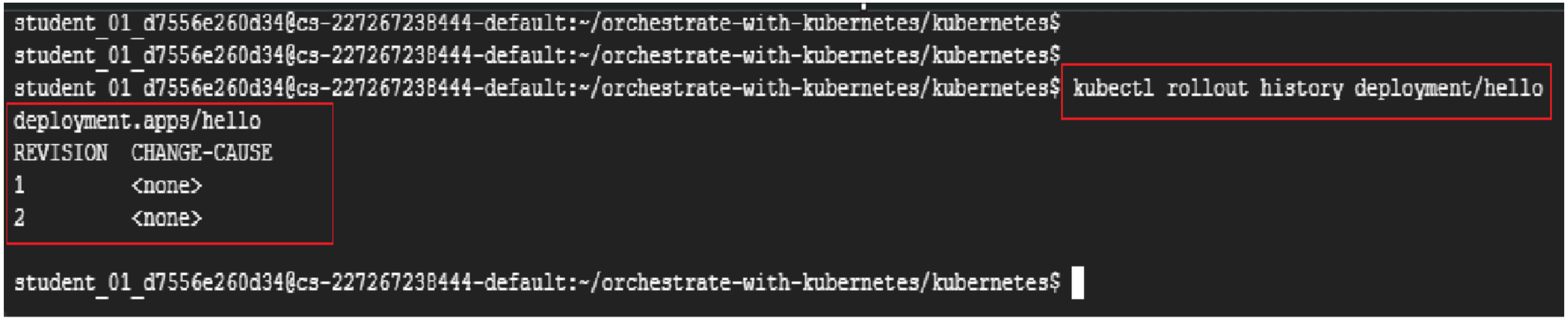
New ReplicaSet created by Kubernetes: Kubectl get replicaset



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl get replicaset
NAME                  DESIRED   CURRENT   READY   AGE
auth-69d588f955       1         1         1       18m
frontend-9c7c7c45b    1         1         1       10m
hello-57d9c6cd57      2         2         1       28s
hello-65b56477b8      2         2         2       13m
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

View the changes in the rollout history:

kubectl rollout history deployment/hello

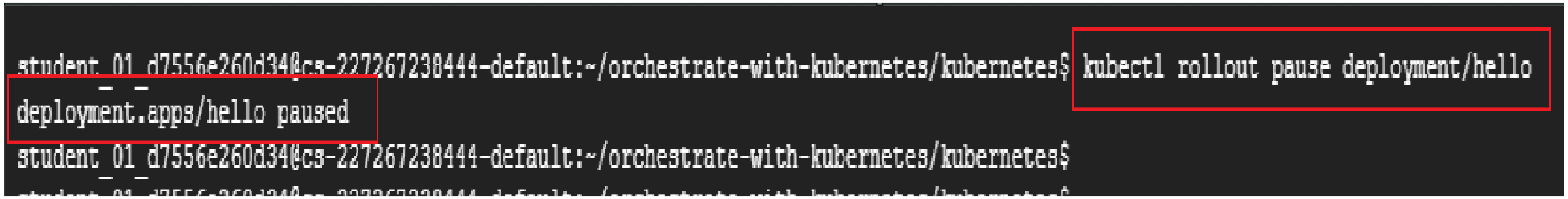


```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl rollout history deployment/hello
deployment.apps/hello
REVISION  CHANGE-CAUSE
1         <none>
2         <none>

student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Pause a rolling update

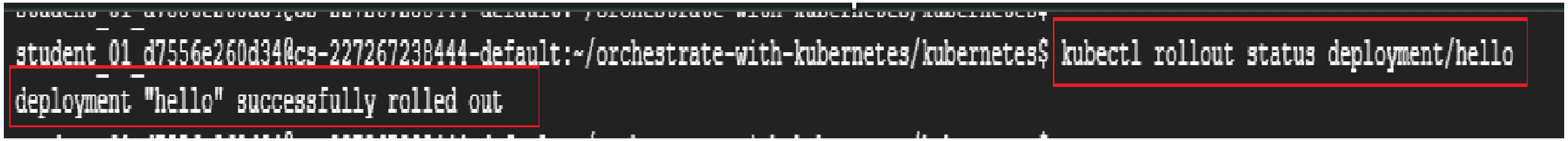
If you detect problems with a running rollout, pause it to stop the update.

Pause the rollout: kubectl rollout pause deployment/hello



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl rollout pause deployment/hello
deployment.apps/hello paused
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

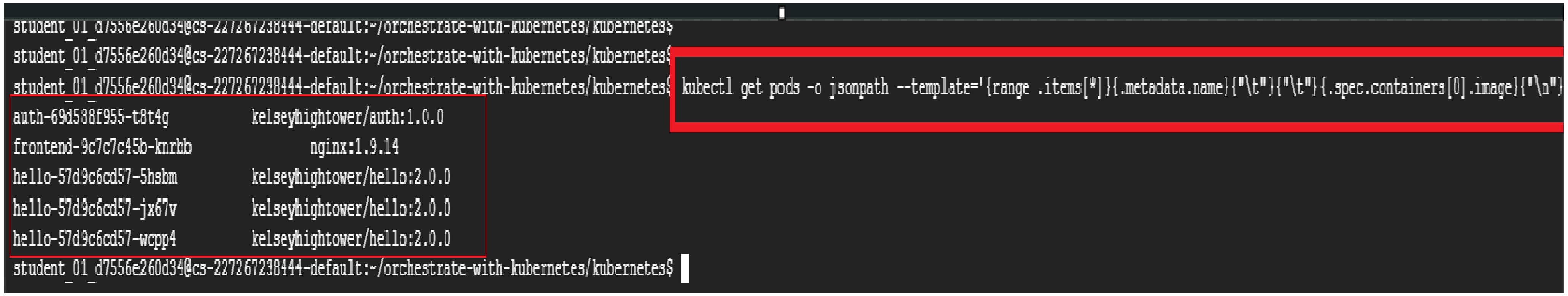
Verify the current state of the rollout: kubectl rollout status deployment/hello



```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl rollout status deployment/hello
deployment "hello" successfully rolled out
```

You can also verify this on the Pods directly:

```
kubectl get pods -o jsonpath --template='{range .items[*]}{.metadata.name}{"\t"}{"\t"}{.spec.containers[0].image}{"\n"}{end}'
```

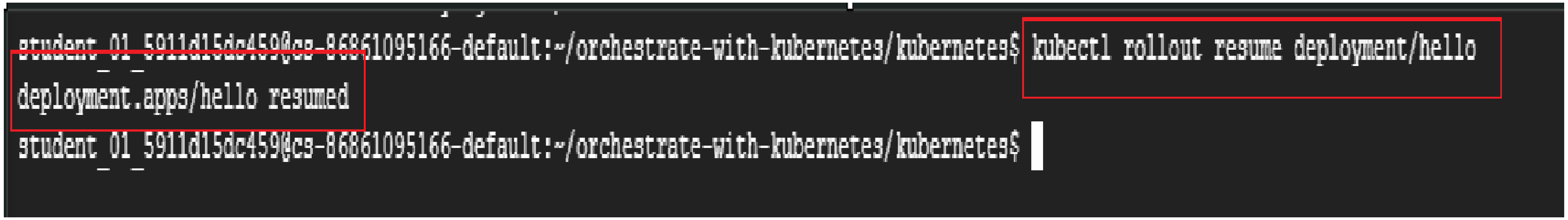


```
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl get pods -o jsonpath --template='{range .items[*]}{.metadata.name}{"\t"}{"\t"}{.spec.containers[0].image}{"\n"}
auth-69d588f955-t8t4g           kelseyhightower/auth:1.0.0
frontend-9c7c7c45b-knrbb                nginx:1.9.14
hello-57d9c6cd57-5hsbm          kelseyhightower/hello:2.0.0
hello-57d9c6cd57-jx67v          kelseyhightower/hello:2.0.0
hello-57d9c6cd57-wcpp4          kelseyhightower/hello:2.0.0
student_01_d7556e260d34@cs-227267238444-default:~/orchestrate-with-kubernetes/kubernetes$
```

Resume a rolling update

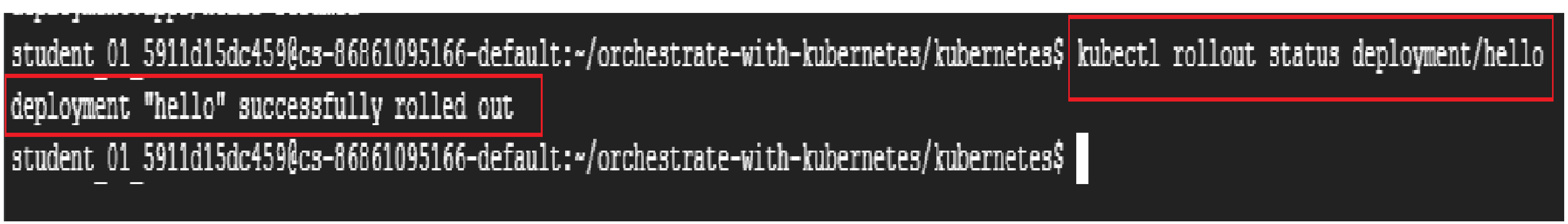
The rollout is paused which means that some pods are at the new version and some pods are at the older version.

Continue the rollout: kubectl rollout resume deployment/hello



```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl rollout resume deployment/hello
deployment.apps/hello resumed
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

Rollout is complete: kubectl rollout status deployment/hello

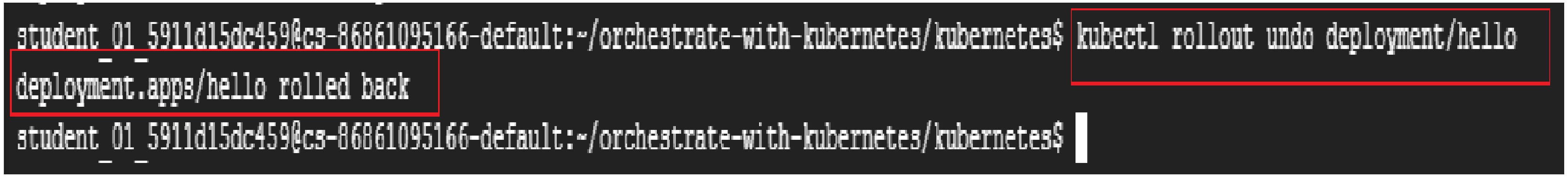


```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl rollout status deployment/hello
deployment "hello" successfully rolled out
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

Roll back an update

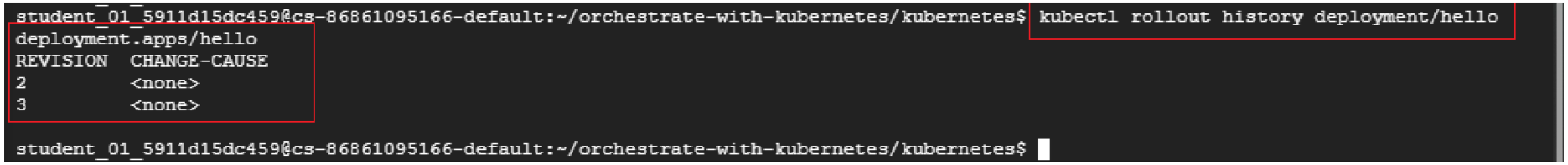
Assume that a bug was detected in your new version. Since the new version is presumed to have problems, any users connected to the new Pods will experience those issues.You will want to roll back to the previous version so you can investigate and then release a version that is fixed properly.

Roll back to the previous version: kubectl rollout undo deployment/hello



```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl rollout undo deployment/hello
deployment.apps/hello rolled back
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

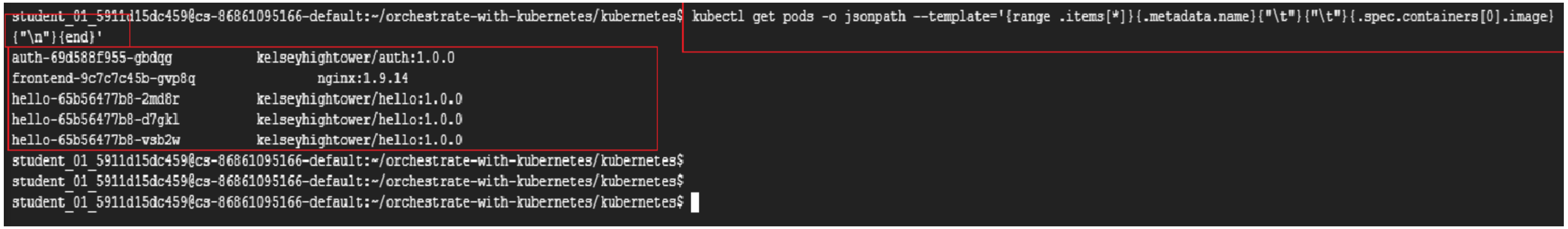
Verify the roll back in the history: kubectl rollout history deployment/hello



```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl rollout history deployment/hello
deployment.apps/hello
REVISION  CHANGE-CAUSE
2         <none>
3         <none>

student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

Verify that all the Pods have rolled back to their previous versions:

```
kubectl get pods -o jsonpath --template='{range .items[*]}{.metadata.name}{"\t"}{"\t"}{.spec.containers[0].image}{"\n"}{end}'
```



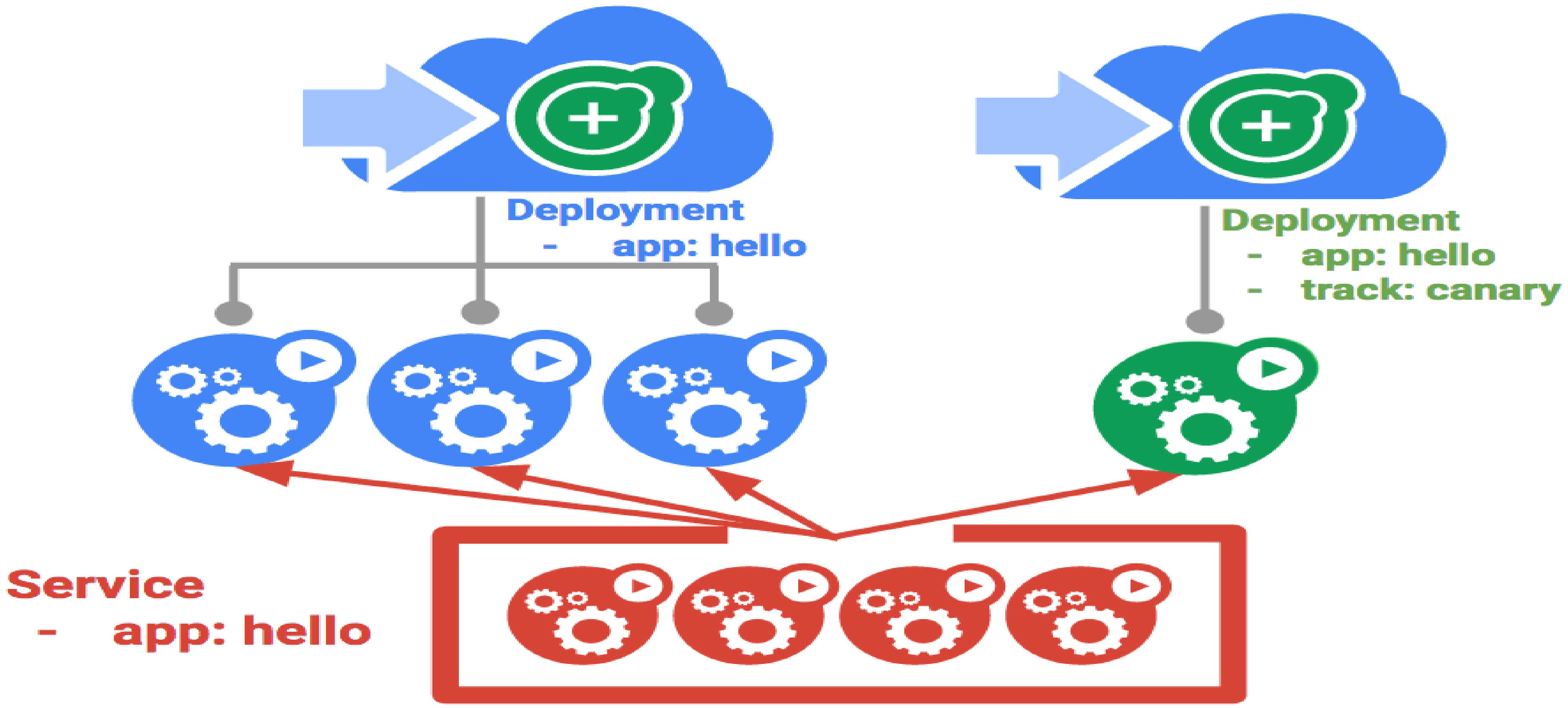
```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl get pods -o jsonpath --template='{range .items[*]}{.metadata.name}{"\t"}{"\t"}{.spec.containers[0].image}
{"\n"}{end}'
auth-69d588f955-gbdqg           kelseyhightower/auth:1.0.0
frontend-9c7c7c45b-gvp8q                nginx:1.9.14
hello-65b56477b8-2md8r          kelseyhightower/hello:1.0.0
hello-65b56477b8-d7gkl          kelseyhightower/hello:1.0.0
hello-65b56477b8-vsb2w          kelseyhightower/hello:1.0.0
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

Canary deployments

When you want to test a new deployment in production with a subset of your users, use a canary deployment.

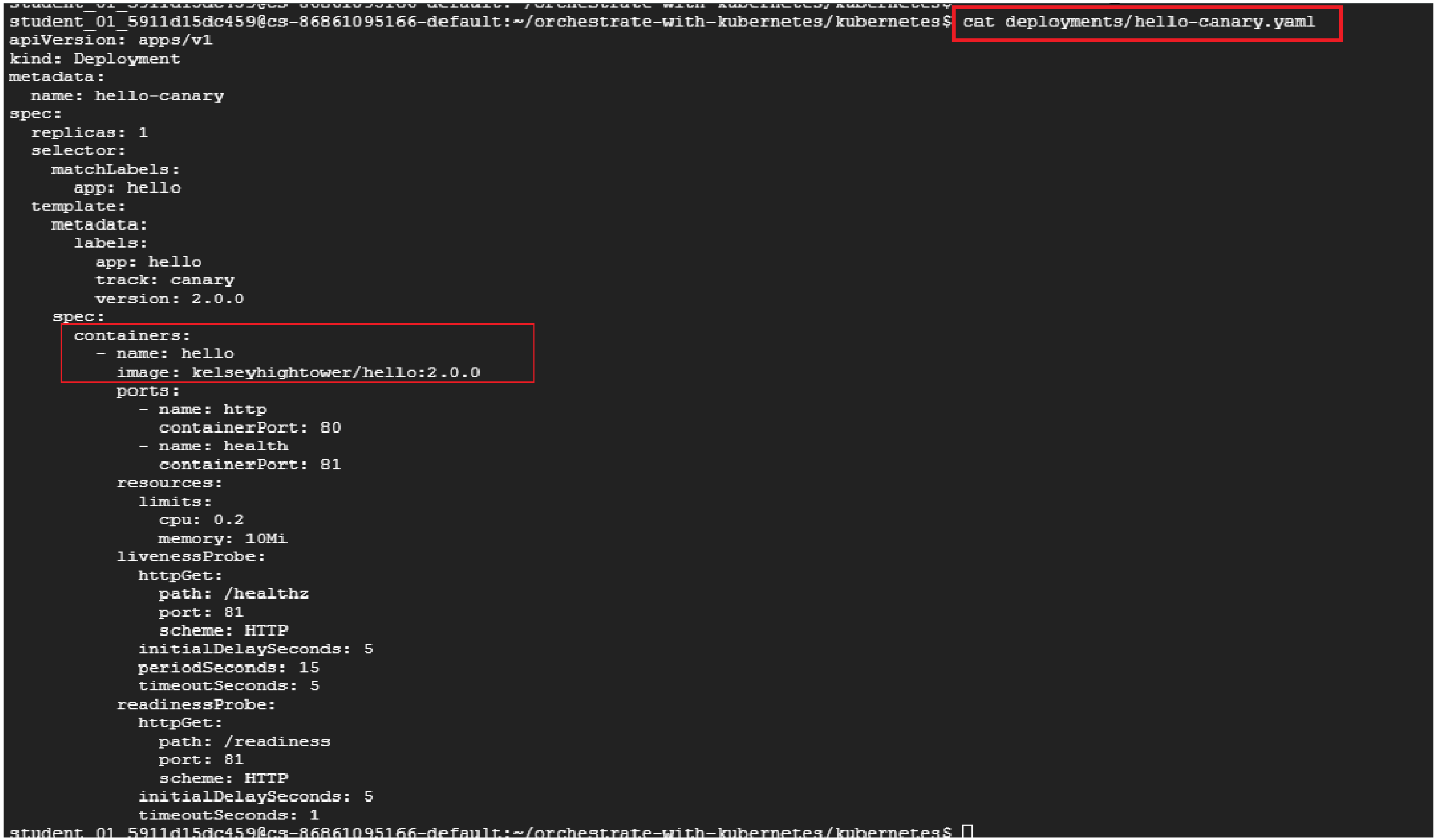
Canary deployments allow you to release a change to a small subset of your users to mitigate risk associated with new releases.

A canary deployment consists of a separate deployment with your new version and a service that targets both your normal, stable deployment as well as your canary deployment.



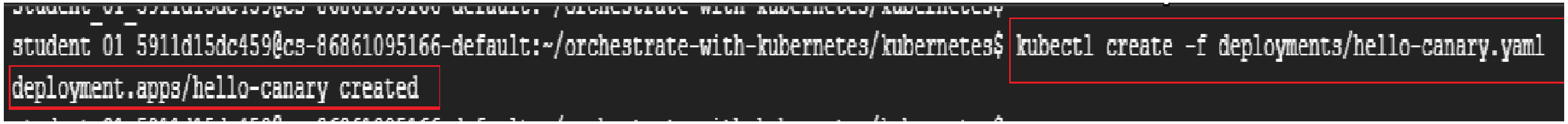
Create a Canary Deployment

Create a new canary deployment for the new version: cat deployments/hello-canary.yaml



```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ cat deployments/hello-canary.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hello-canary
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hello
  template:
    metadata:
      labels:
        app: hello
        track: canary
        version: 2.0.0
    spec:
      containers:
        - name: hello
          image: kelseyhightower/hello:2.0.0
          ports:
            - name: http
              containerPort: 80
            - name: health
              containerPort: 81
          resources:
            limits:
              cpu: 0.2
              memory: 10Mi
          livenessProbe:
            httpGet:
              path: /healthz
              port: 81
              scheme: HTTP
            initialDelaySeconds: 5
            periodSeconds: 15
            timeoutSeconds: 5
          readinessProbe:
            httpGet:
              path: /readiness
              port: 81
              scheme: HTTP
            initialDelaySeconds: 5
            timeoutSeconds: 1
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

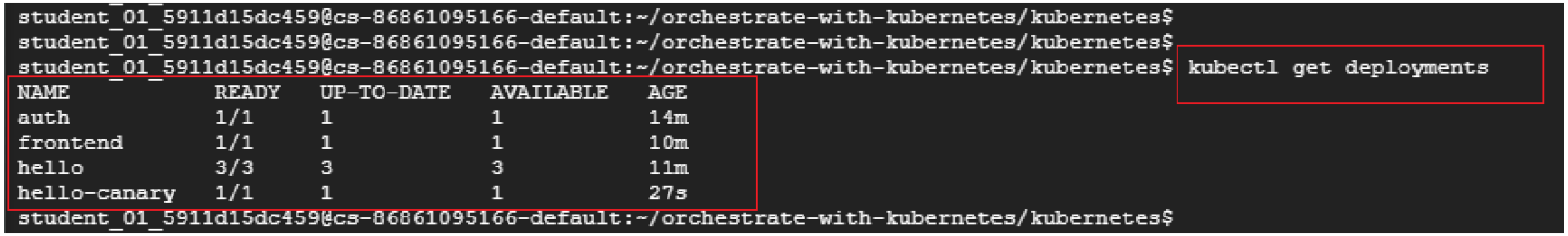
Create the canary deployment: Kubectl create -f deployments/hello-canary.yaml



```
student 01 5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl create -f deployments/hello-canary.yaml
deployment.apps/hello-canary created
```

After the canary deployment is created, you should have two deployments, hello and hello-canary.

Verify it with this kubectl command: kubectl get deployments



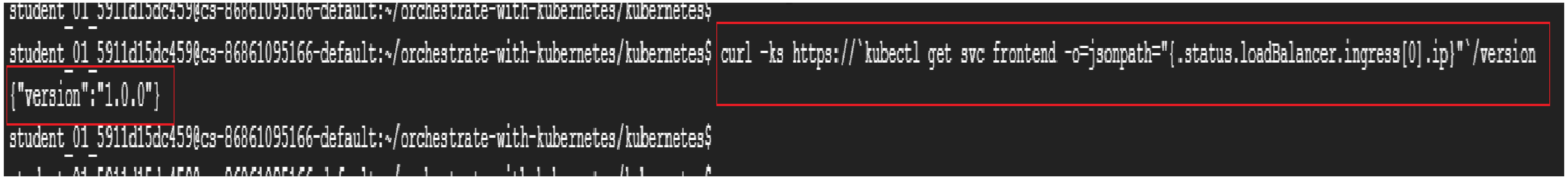
```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl get deployments
NAME           READY   UP-TO-DATE   AVAILABLE   AGE
auth           1/1     1            1           14m
frontend       1/1     1            1           10m
hello          3/3     3            3           11m
hello-canary   1/1     1            1           27s
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

On the hello service, the app:hello selector will match pods in both the prod deployment and canary deployment. However, because the canary deployment has a fewer number of pods, it will be visible to fewer users.

Verify the canary deployment

Verify the hello version being served by the request:

```
curl -ks https://`kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"`/version
```



```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ curl -ks https://`kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"`/version
{"version":"1.0.0"}
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

Run this several times and you should see that some of the requests are served by hello 1.0.0 and a small subset (1/4 = 25%) are served by 2.0.0.

Canary deployments in production - session affinity

In this project, each request sent to the Nginx service had a chance to be served by the canary deployment.

But what if you wanted to ensure that a user didn't get served by the canary deployment?

A use case could be that the UI for an application changed, and you don't want to confuse the user. In a case like this, you want the user to "stick" to one deployment or the other.

You can do this by creating a service with session affinity. This way the same user will always be served from the same version.

In the example below, the service is the same as before, but a new sessionAffinity field has been added, and set to ClientIP. All clients with the same IP address will have their requests sent to the same version of the hello application.

```
kind: Service
apiVersion: v1
metadata:
  name: "hello"
spec:
  sessionAffinity: ClientIP
  selector:
    app: "hello"
  ports:
    - protocol: "TCP"
      port: 80
      targetPort: 80
```

Blue-green deployments

Rolling updates are ideal because they allow you to deploy an application slowly with minimal overhead, minimal performance impact, and minimal downtime.

There are instances where it is beneficial to modify the load balancers to point to that new version only after it has been fully deployed.

In this case, blue-green deployments are the way to go.

Kubernetes achieves this by creating two separate deployments;

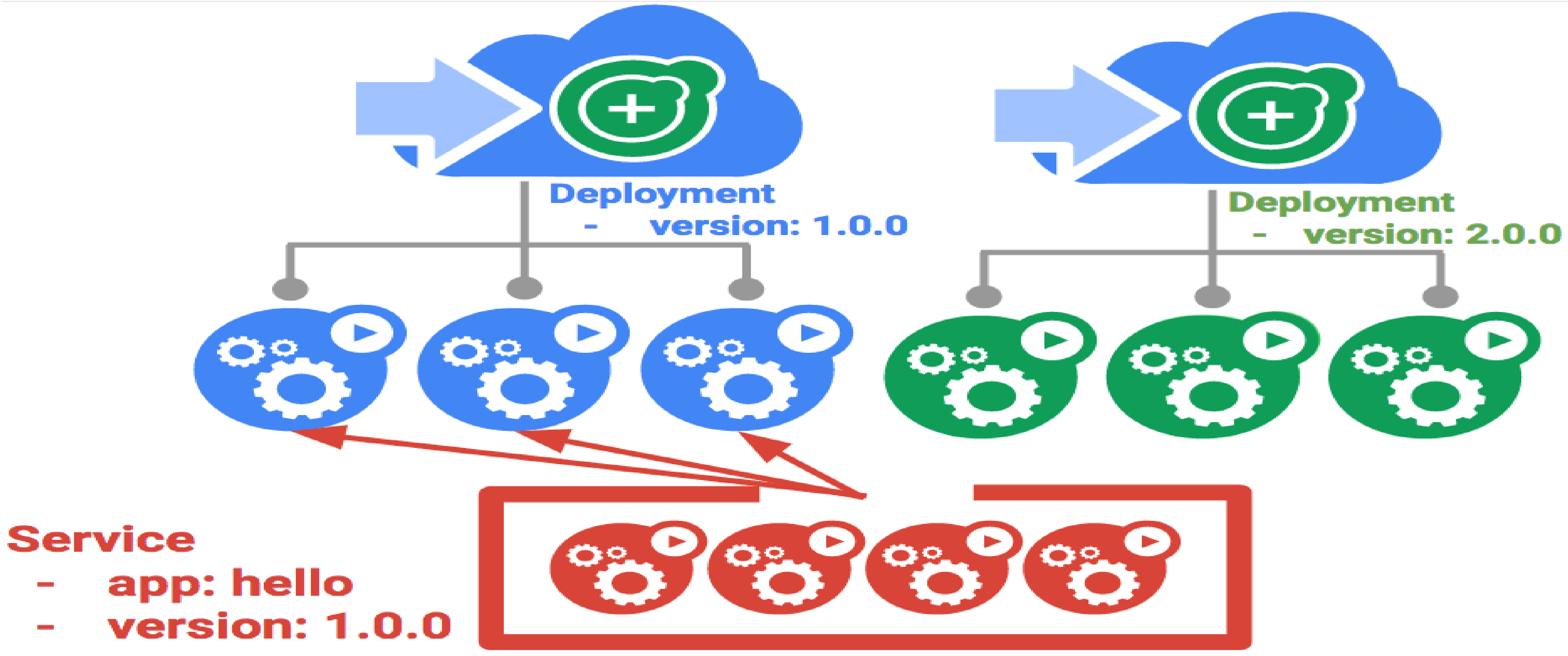
one for the old "blue" version

one for the new "green" version.

Use your existing hello deployment for the "blue" version.

The deployments will be accessed via a service which will act as the router.

Once the new "green" version is up and running, you'll switch over to using that version by updating the service.



Disadvantage of blue-green deployments: At least 2x the resources are needed in your cluster necessary to host your application.

The service

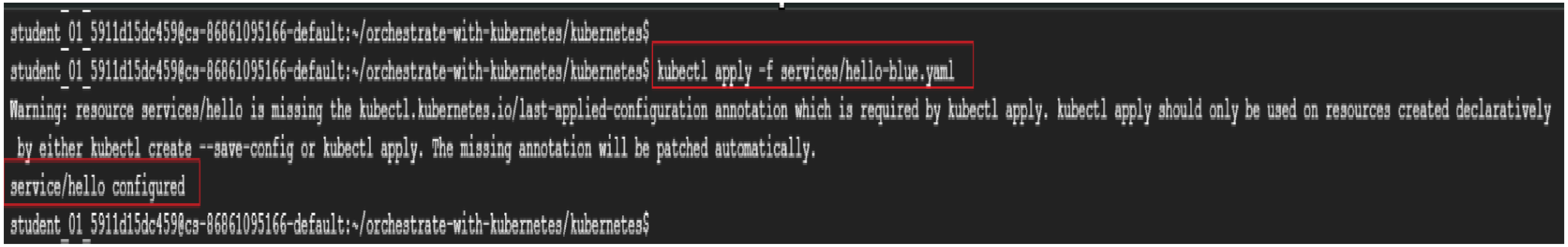
Use the existing hello service, but update it so that it has a selector app:hello, version: 1.0.0.

The selector will match the existing "blue" deployment.

But it will not match the "green" deployment because it will use a different version.

Update the service:

```
Kubectl apply -f services/hello-blue.yaml
```



```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl apply -f services/hello-blue.yaml
Warning: resource services/hello is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resources created declaratively
 by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.
service/hello configured
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

Updating using Blue-Green deployment

In order to support a blue-green deployment style, you will create a new "green" deployment for the new version.

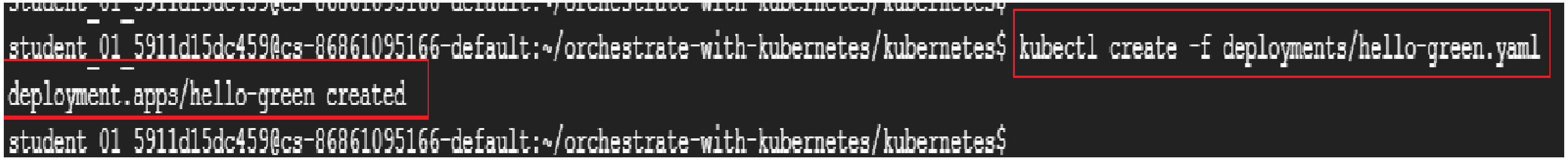
The green deployment updates the version label and the image path.

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: hello-green
spec:
 replicas: 3
 selector:
  matchLabels:
   app: hello
 template:
  metadata:
   labels:
    app: hello
    track: stable
    version: 2.0.0
  spec:
   containers:
    - name: hello
     image: kelseyhightower/hello:2.0.0
```

```
ports:
 - name: http
   containerPort: 80
 - name: health
   containerPort: 81
resources:
 limits:
   cpu: 0.2
   memory: 10Mi
livenessProbe:
 httpGet:
   path: /healthz
   port: 81
   scheme: HTTP
 initialDelaySeconds: 5
 periodSeconds: 15
 timeoutSeconds: 5
readinessProbe:
 httpGet:
   path: /readiness
   port: 81
   scheme: HTTP
 initialDelaySeconds: 5
 timeoutSeconds: 1
```

Create the green deployment:

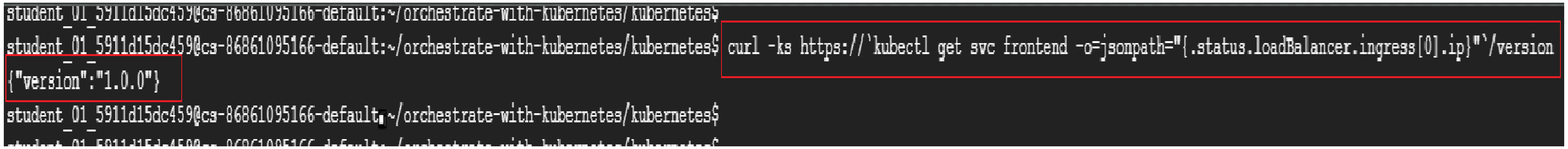
kubectl create -f deployments/hello-green.yaml



```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl create -f deployments/hello-green.yaml
deployment.apps/hello-green created
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

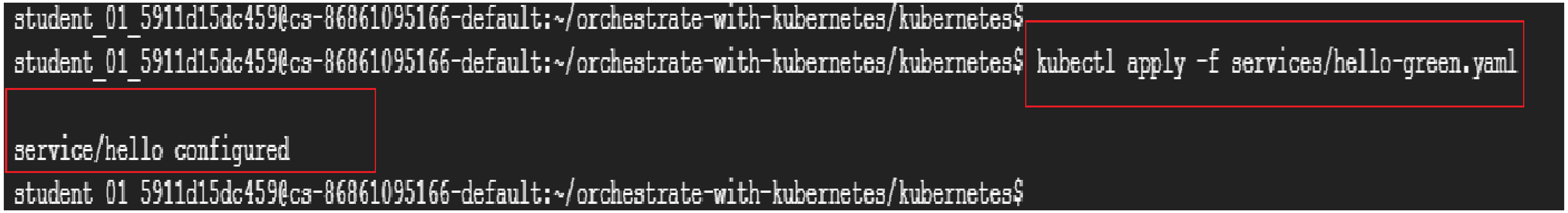
Verify that the current version of 1.0.0 is still being used:

curl -ks https://`kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"`/version



```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ curl -ks https://`kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"`/version
{"version":"1.0.0"}
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

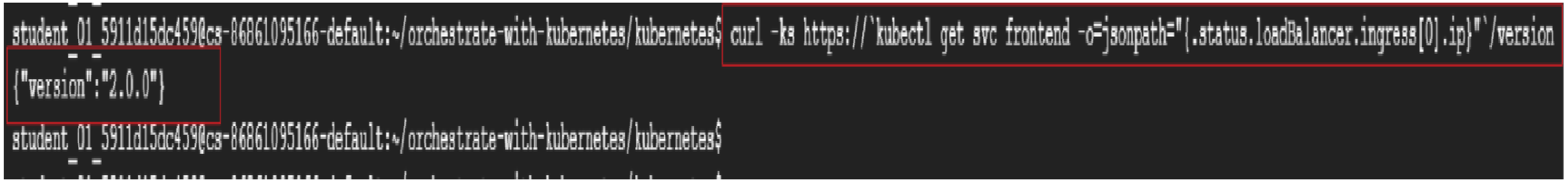
Update the service to point to the new version: kubectl apply -f services/hello-green.yaml



```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl apply -f services/hello-green.yaml

service/hello configured
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

When the service is updated, the "green" deployment will be used immediately. You can now verify that the new version is always being used:

curl -ks https://`kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"`/version

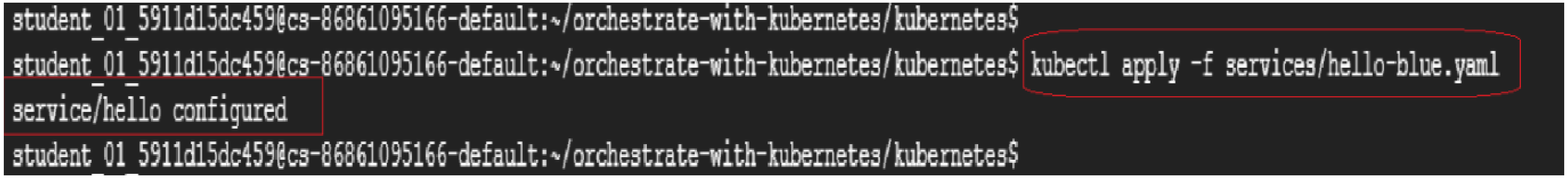


```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ curl -ks https://`kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"`/version
{"version":"2.0.0"}
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

Blue-Green rollback to older version

While the "blue" deployment is still running, just update the service back to the old version:

kubectl apply -f services/hello-blue.yaml

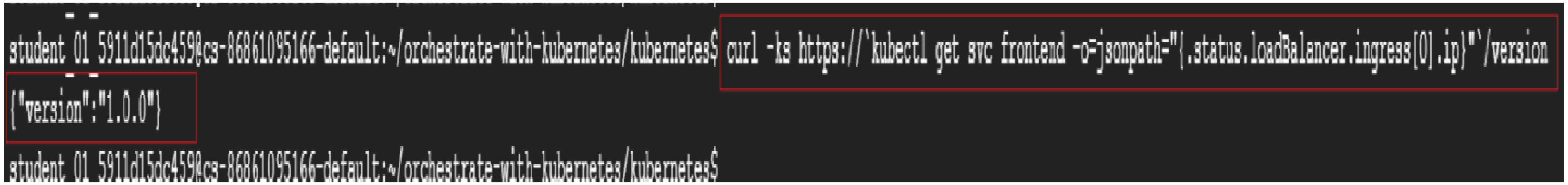


```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ kubectl apply -f services/hello-blue.yaml
service/hello configured
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```

Once you have updated the service, your rollback will have been successful.

Verify that the right version is now being used:

curl -ks https://` kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"` /version



```
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$ curl -ks https://`kubectl get svc frontend -o=jsonpath="{.status.loadBalancer.ingress[0].ip}"`/version
{"version":"1.0.0"}
student_01_5911d15dc459@cs-86861095166-default:~/orchestrate-with-kubernetes/kubernetes$
```