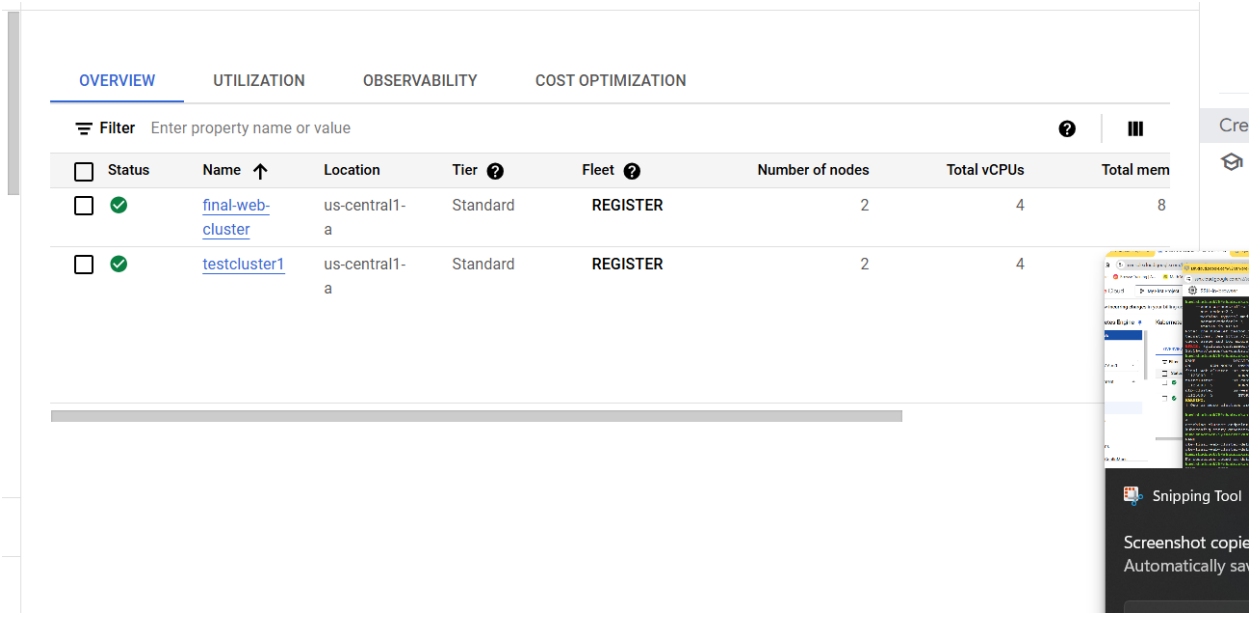
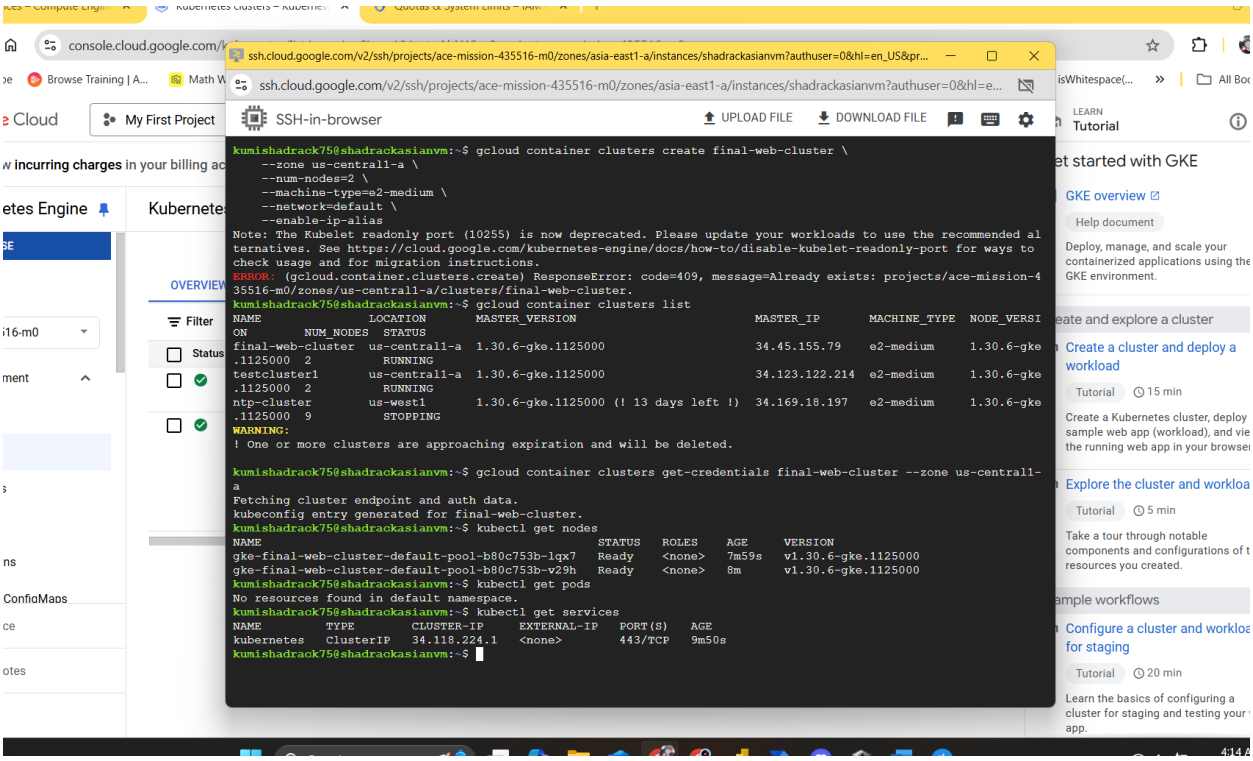
1. Create a GKE Cluster

1. **GCP Project Setup**:
    - Navigate to the Google Cloud Console.
    - Create a new project and enable the **Google Kubernetes Engine API**.
2. **Cluster Configuration**:
    - Use the **GKE Console** or **gcloud CLI** to create a cluster:





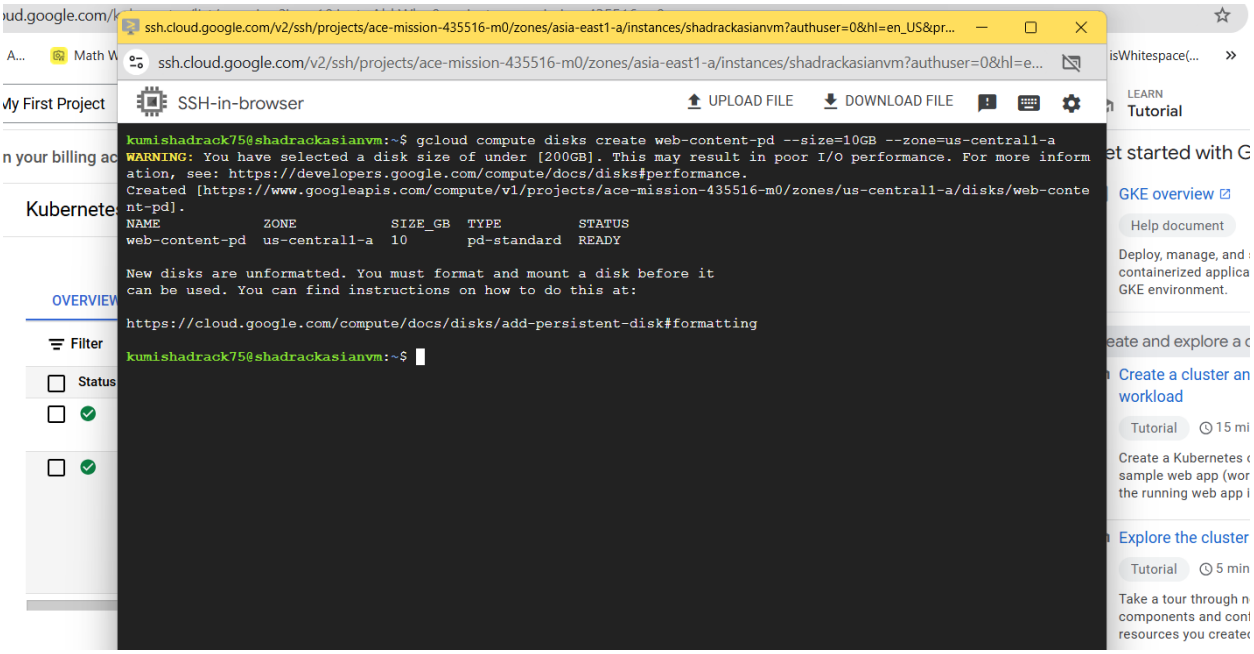
Note the cluster's name, zone (us-central1-a), machine type, and number of nodes.

Firewall Rules:

- Ensure the default VPC network has firewall rules to allow external traffic on the web server's port (e.g., HTTP/80).

2. Create a Persistent Disk

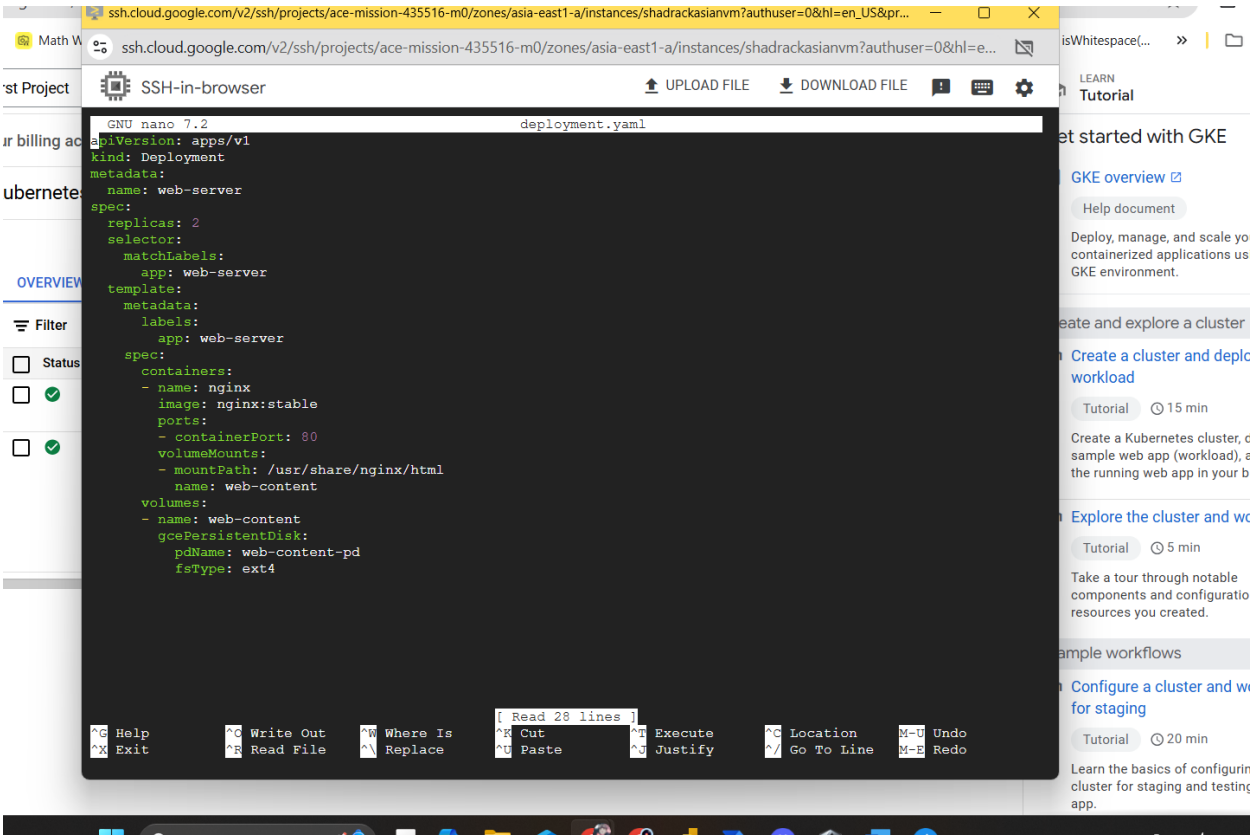
1. **Persistent Disk Setup**:
    - Create a 10GB Persistent Disk:



Document its name (web-content-pd), size, and zone.

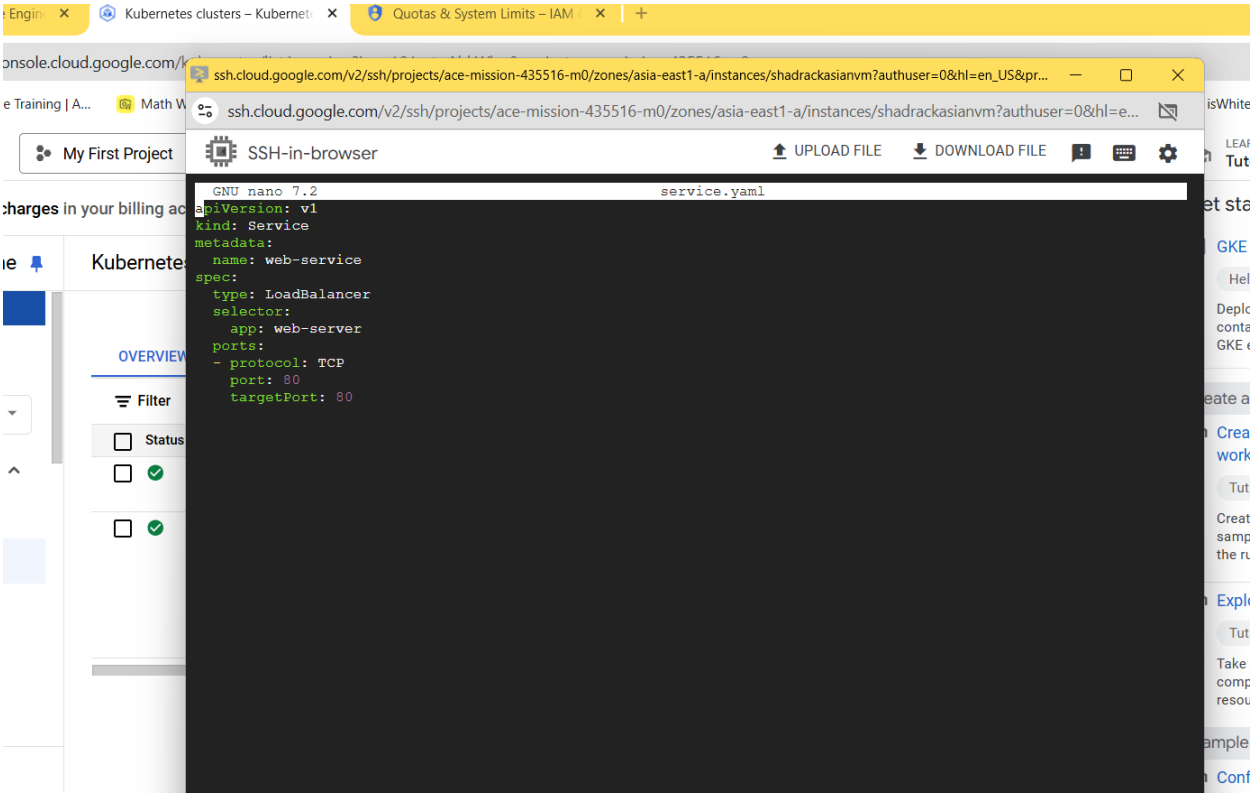
3. Deploy the Web Server

1. **Choose Container Image**:
    - Use a pre-built Nginx image: nginx:stable.
2. **Create Deployment Manifest**:
    - Save the following to deployment.yaml



Create Service Manifest:

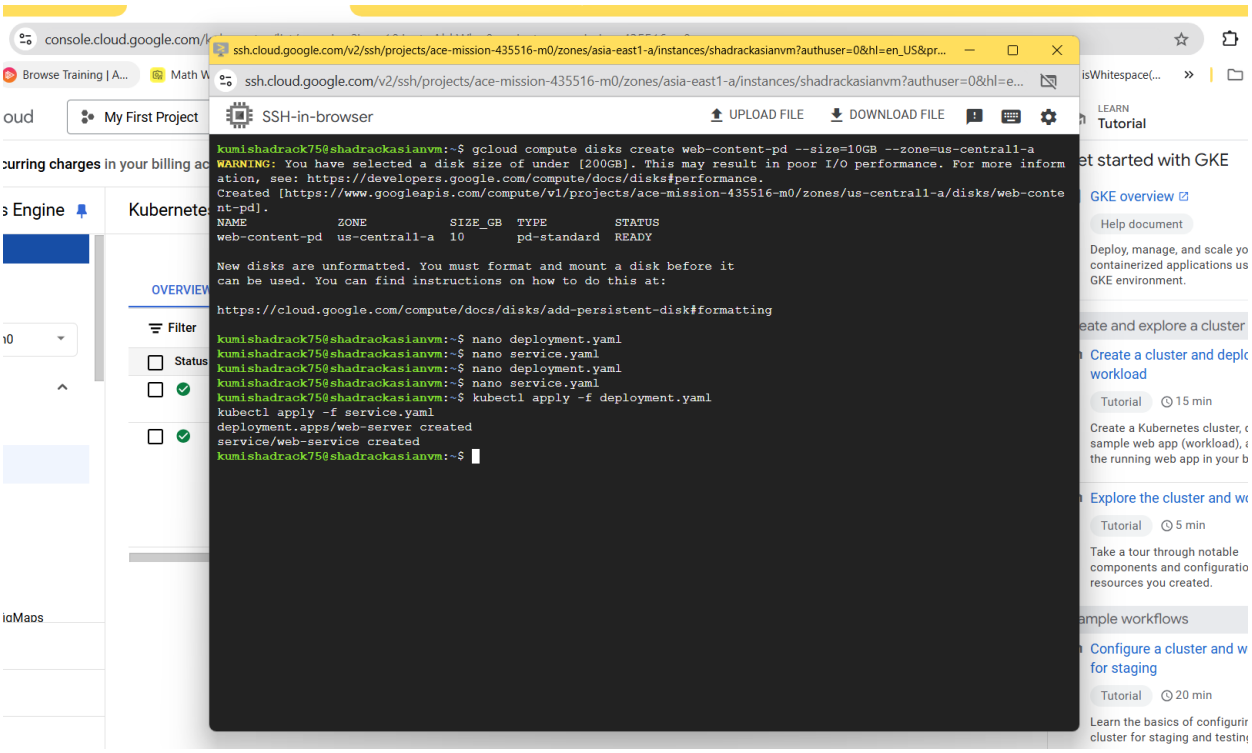
- Save the following to service.yaml



Deploy Resources:

kubectl apply -f deployment.yaml

kubectl apply -f service.yaml

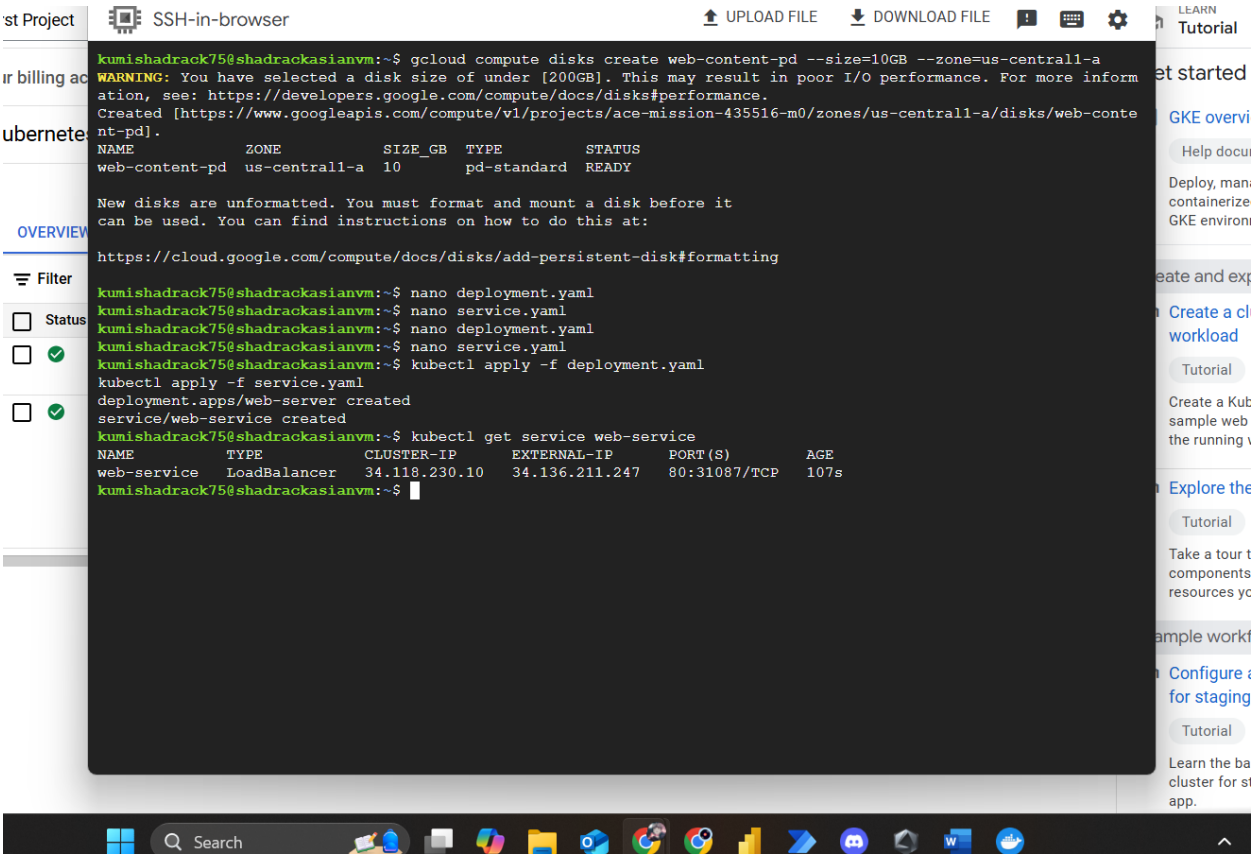


Verify Deployment:

- Get the external IP address:

kubectl get service web-service

Test access using a browser or curl.



4. Configure Monitoring

1. **Enable GKE Monitoring**:
    - Use **Metrics Explorer** in the Cloud Console to monitor:
        - CPU and memory usage.
        - HTTP request metrics.
        - Disk I/O performance.
2. **Set Alerts**:
    - Configure alerts for critical metrics:
        - Navigate to **Monitoring** > **Alerting** in Cloud Console.
        - Add policies for thresholds (e.g., CPU > 80%, HTTP errors).

3. **Security Monitoring**:
    - Enable **Cloud Security Command Center** for threat detection.

5. Test the Web Server

1. **Verify Content Serving**:
    - Check the web server by accessing its external IP

curl http://<EXTERNAL_IP>

Simulate Failures:

- Kill a pod to test high availability

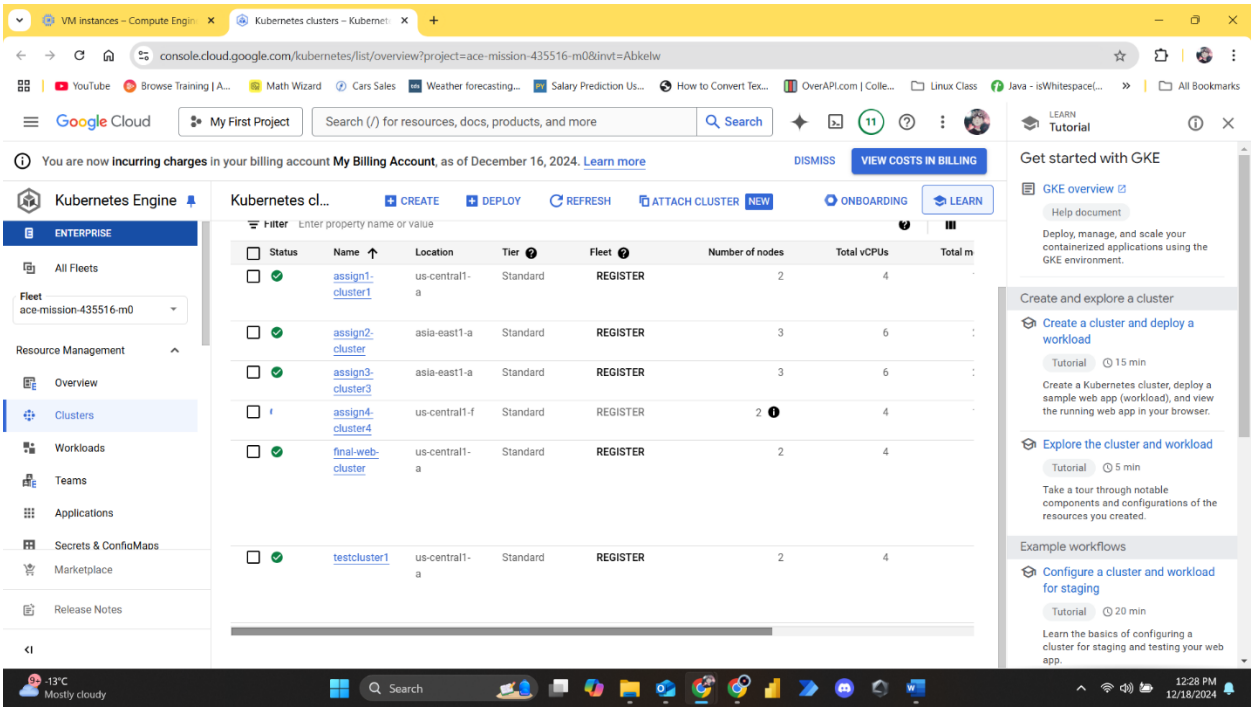
kubectl delete pod <pod-name>

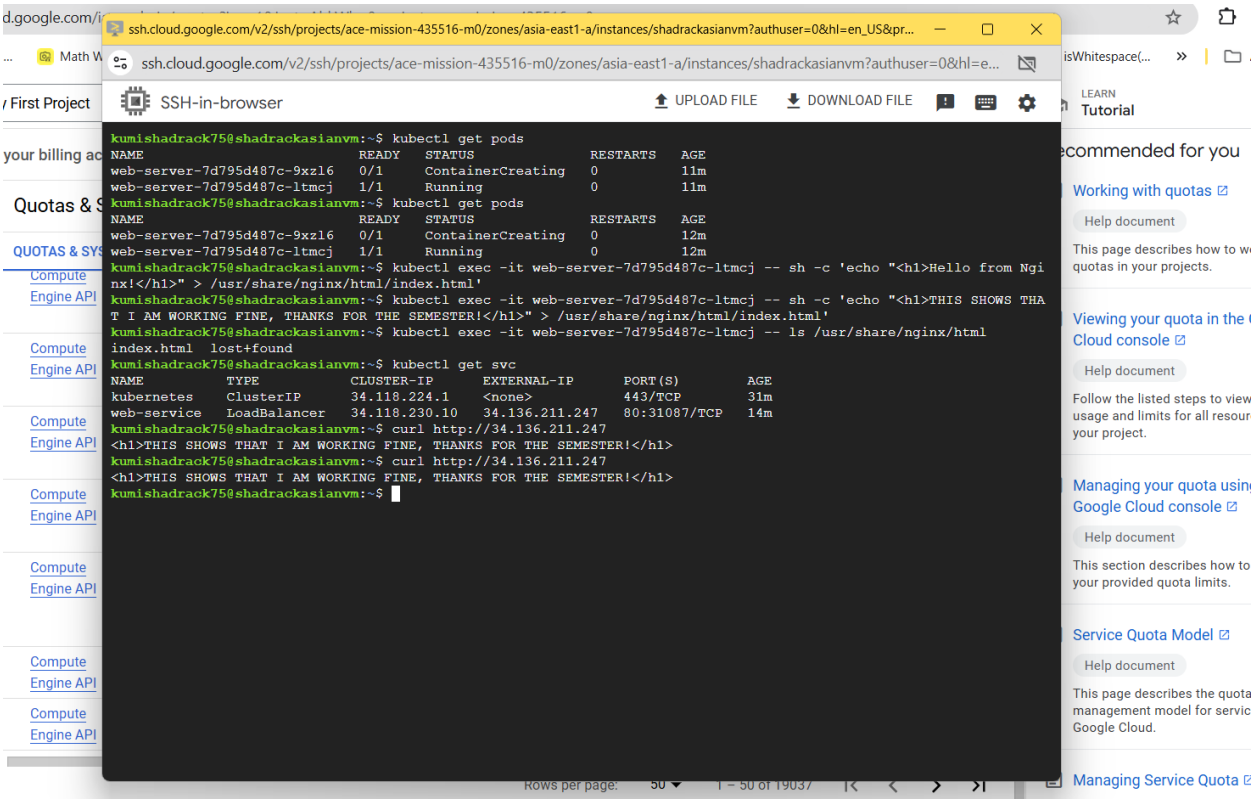
Verify traffic is routed to healthy pods

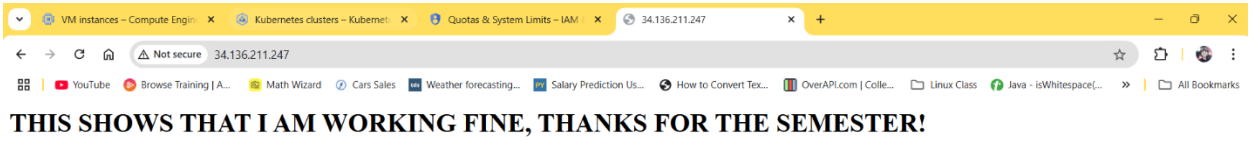
Trigger Alerts:

- Stress-test resources to ensure alerts work:

kubectl exec <pod-name> -- stress --cpu 4 --timeout 60


VM instances – Compute Engine
Kubernetes clusters – Kubernetes
console.cloud.google.com/kubernetes/list/overview?project=ace-mission-435516-m0&invt=Abkelw
Google Cloud
My First Project
Search (/) for resources, docs, products, and more
Search
You are now incurring charges in your billing account My Billing Account, as of December 16, 2024. Learn more
DISMISS
VIEW COSTS IN BILLING
Kubernetes Engine
Kubernetes cl...
CREATE
DEPLOY
REFRESH
ATTACH CLUSTER
NEW
ONBOARDING
LEARN
ENTERPRISE
All Fleets
Fleet
ace-mission-435516-m0
Resource Management
Overview
Clusters
Workloads
Teams
Applications
Secrets & ConfigMaps
Marketplace
Release Notes
Filter
Enter property name or value
Status
Name
Location
Tier
Fleet
Number of nodes
Total vCPUs
assign1-cluster1
us-central1-a
Standard
REGISTER
2
4
assign2-cluster
asia-east1-a
Standard
REGISTER
3
6
assign3-cluster3
asia-east1-a
Standard
REGISTER
3
6
assign4-cluster4
us-central1-f
Standard
REGISTER
2
4
final-web-cluster
us-central1-a
Standard
REGISTER
2
4
testcluster1
us-central1-a
Standard
REGISTER
2
4
LEARN
Tutorial
Get started with GKE
GKE overview
Help document
Deploy, manage, and scale your containerized applications using the GKE environment.
Create and explore a cluster
Create a cluster and deploy a workload
Tutorial
15 min
Create a Kubernetes cluster, deploy a sample web app (workload), and view the running web app in your browser.
Explore the cluster and workload
Tutorial
5 min
Take a tour through notable components and configurations of the resources you created.
Example workflows
Configure a cluster and workload for staging
Tutorial
20 min
Learn the basics of configuring a cluster for staging and testing your web app.
-13°C
Mostly cloudy
Search
12:28 PM
12/18/2024


ssh.cloud.google.com/v2/ssh/projects/ace-mission-435516-m0/zones/asia-east1-a/instances/shadrackasianvm?authuser=0&hl=en_US&pr...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
kumishadrack75@shadrackasianvm:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
web-server-7d795d487c-9xzl6 0/1 ContainerCreating 0 11m
web-server-7d795d487c-ltmcj 1/1 Running 0 11m
kumishadrack75@shadrackasianvm:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
web-server-7d795d487c-9xzl6 0/1 ContainerCreating 0 12m
web-server-7d795d487c-ltmcj 1/1 Running 0 12m
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-7d795d487c-ltmcj -- sh -c 'echo "<h1>Hello from Nginx!</h1>" > /usr/share/nginx/html/index.html'
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-7d795d487c-ltmcj -- sh -c 'echo "<h1>THIS SHOWS THAT I AM WORKING FINE, THANKS FOR THE SEMESTER!</h1>" > /usr/share/nginx/html/index.html'
kumishadrack75@shadrackasianvm:~$ kubectl exec -it web-server-7d795d487c-ltmcj -- ls /usr/share/nginx/html
index.html lost+found
kumishadrack75@shadrackasianvm:~$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
kubernetes ClusterIP 34.118.224.1 <none> 443/TCP 31m
web-service LoadBalancer 34.118.230.10 34.136.211.247 80:31087/TCP 14m
kumishadrack75@shadrackasianvm:~$ curl http://34.136.211.247
<h1>THIS SHOWS THAT I AM WORKING FINE, THANKS FOR THE SEMESTER!</h1>
kumishadrack75@shadrackasianvm:~$ curl http://34.136.211.247
<h1>THIS SHOWS THAT I AM WORKING FINE, THANKS FOR THE SEMESTER!</h1>
kumishadrack75@shadrackasianvm:~$


Not secure 34.136.211.247
THIS SHOWS THAT I AM WORKING FINE, THANKS FOR THE SEMESTER!