

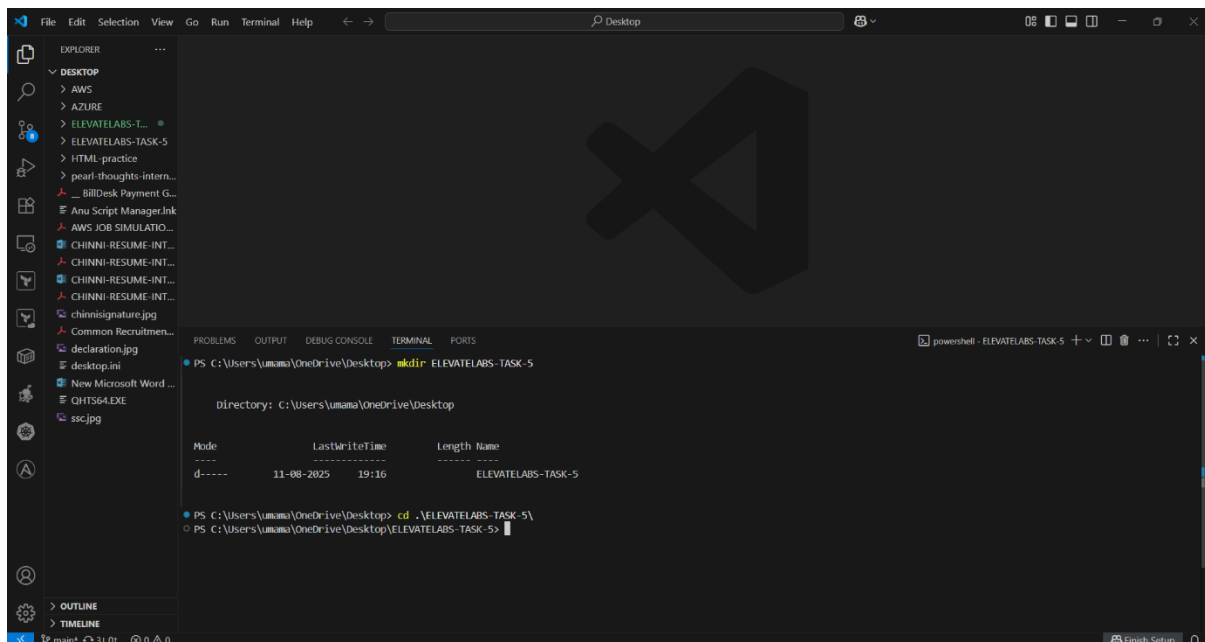
TASK 5: Build a Kubernetes Cluster Locally with Minikube

1. Objective: The objective of this task was to deploy and manage applications in a Kubernetes cluster running locally using Minikube, and to demonstrate Kubernetes concepts such as pods, deployments, services, scaling, and rollback.

2. Tools and Technologies Used

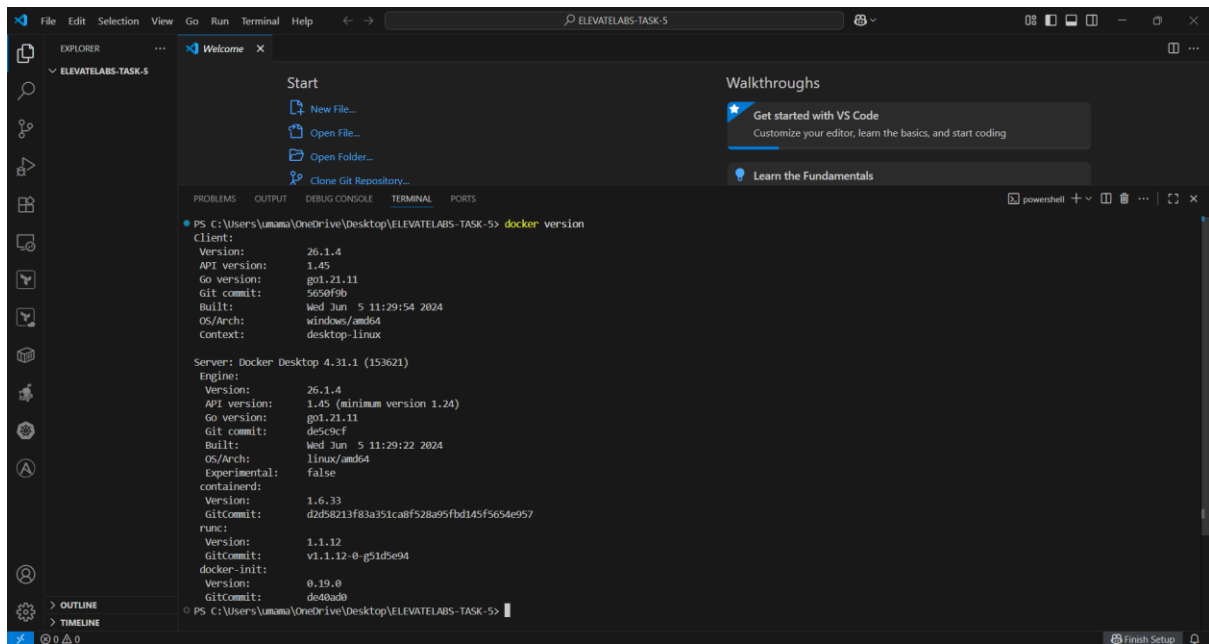
- Minikube – Local Kubernetes cluster
- kubectl – Kubernetes command-line tool
- Docker – Container runtime
- Git & GitHub – Version control and repository hosting

3. Project folder creation locally: using mkdir command

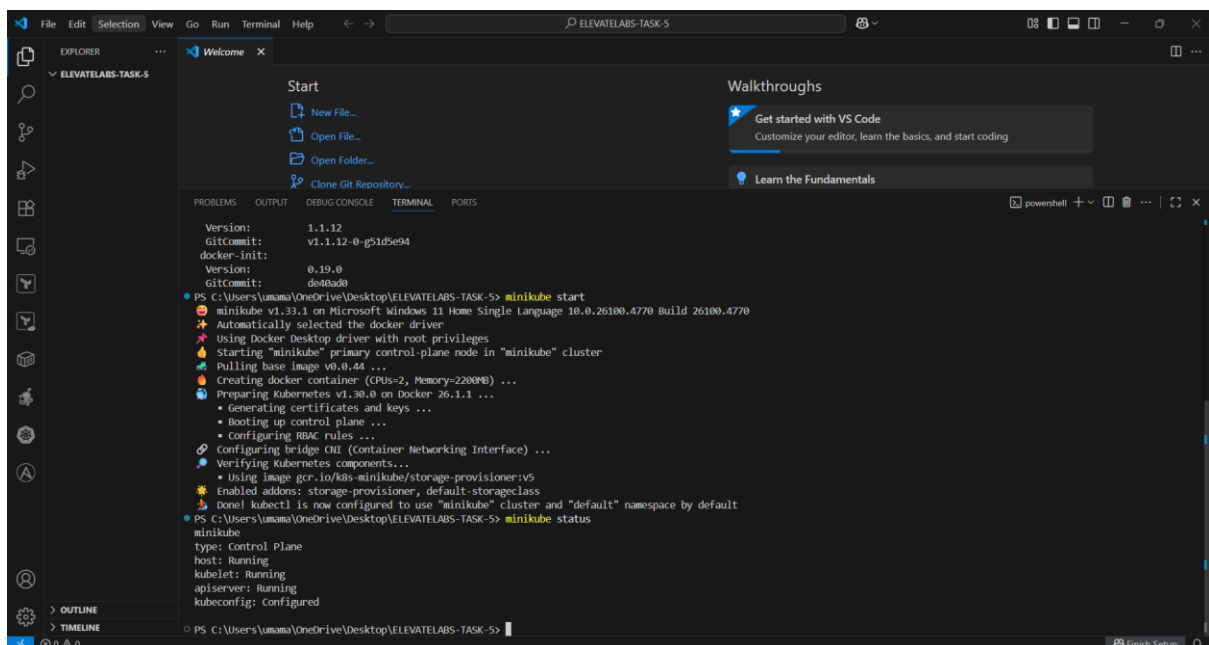


4. Step-by-Step Implementation:

- start docker-engine
- Check docker version: `docker version`

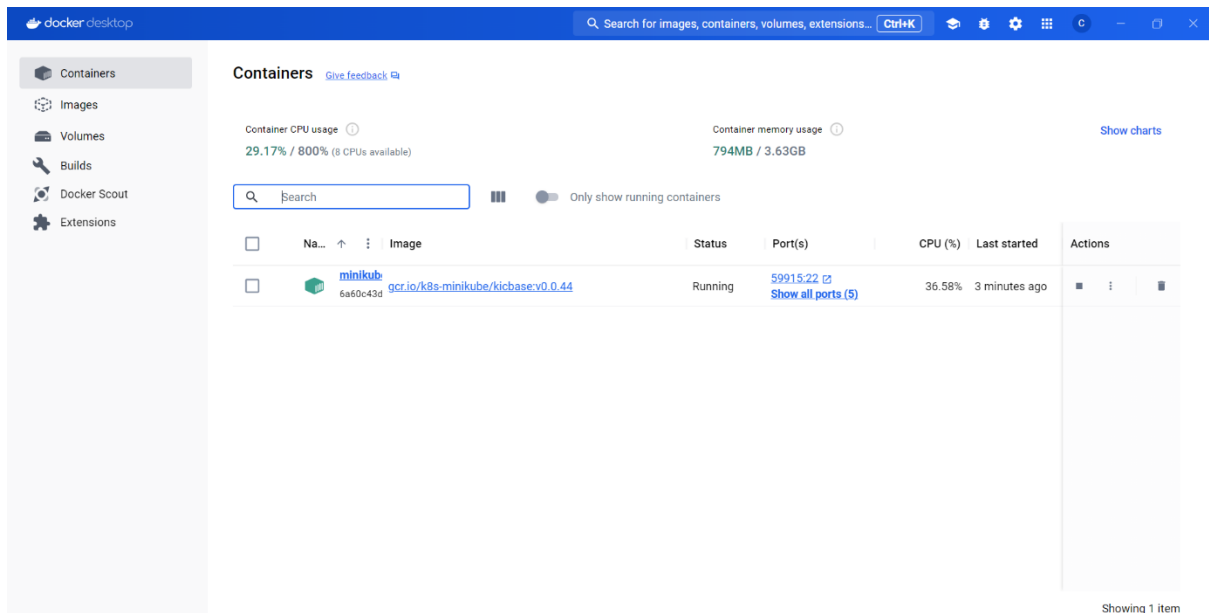


- Start minikube : `minikube start`
- Check status: `minikube status`



We see that minikube container and image are running in docker desktop engine





5. Create Deployment YAML

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

replicas: 2

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

image: nginx:latest

ports:

- containerPort: 80

resources:

requests:

memory: "64Mi"

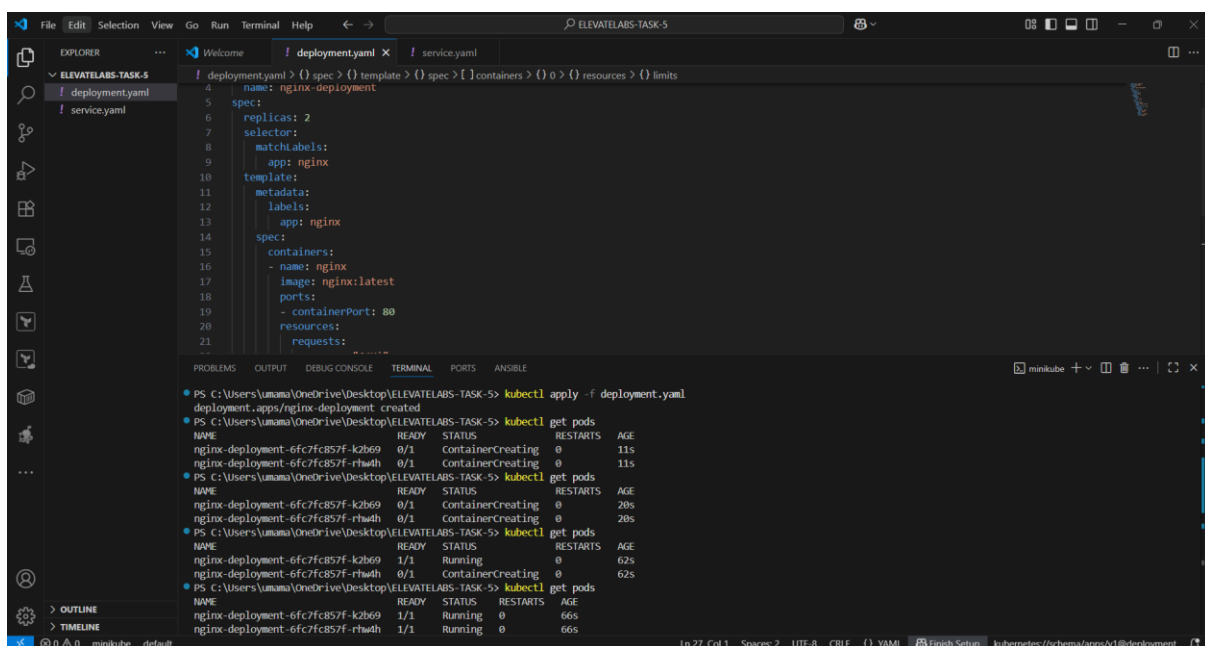
cpu: "250m"

limits:

memory: "128Mi"

cpu: "500m"

- **apiVersion, kind, metadata** → Identifies the resource as a Deployment named nginx-deployment.
- **replicas: 2** → Runs two identical pods for load balancing and high availability.
- **selector & labels** → Ensures the Deployment manages only pods labeled app: nginx.
- **template** → Pod definition specifying the nginx:latest image, listening on port 80.
- **resources** → Requests (minimum) and limits (maximum) CPU & memory to control resource usage.



The screenshot shows a Visual Studio Code editor with a file named `deployment.yaml` open. The file contains a Kubernetes Deployment manifest for nginx. The terminal at the bottom shows the following commands and output:

```
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl apply -f deployment.yaml
deployment.apps/nginx-deployment created
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6fc7fc857f-k2b69   0/1     ContainerCreating   0           11s
nginx-deployment-6fc7fc857f-rh4th   0/1     ContainerCreating   0           11s
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6fc7fc857f-k2b69   0/1     ContainerCreating   0           20s
nginx-deployment-6fc7fc857f-rh4th   0/1     ContainerCreating   0           20s
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6fc7fc857f-k2b69   1/1     Running      0           62s
nginx-deployment-6fc7fc857f-rh4th   0/1     ContainerCreating   0           62s
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6fc7fc857f-k2b69   1/1     Running      0           66s
nginx-deployment-6fc7fc857f-rh4th   1/1     Running      0           66s
```

6. Create Service YAML

apiVersion: v1

kind: Service

metadata:

name: nginx-service

spec:

selector:

app: nginx

ports:

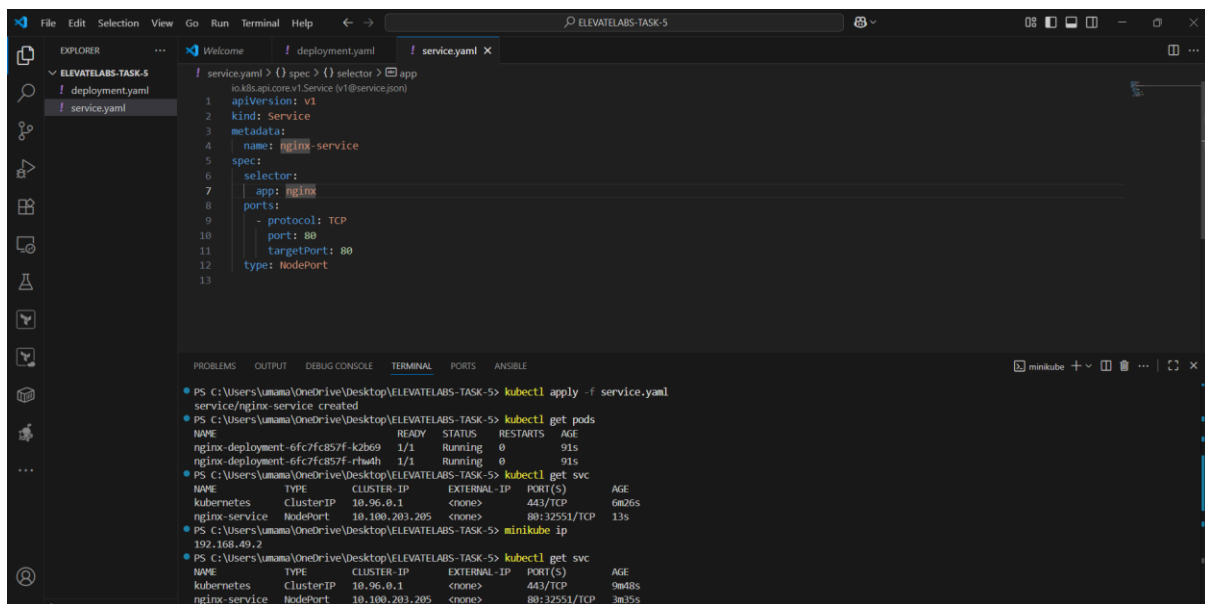
- protocol: TCP

port: 80

targetPort: 80

type: NodePort

- **apiVersion, kind, metadata** → Identifies the resource as a Service named nginx-service.
- **selector** → Matches pods with the label app: nginx (so it routes traffic to them).
- **ports** → **port: 80** → Port on the service itself.
- **targetPort: 80** → Port on the container to forward traffic to.
- **protocol: TCP** → Communication protocol.
- **type: NodePort** → Exposes the service externally via a high port (30000–32767) on each node's IP, allowing access from outside the cluster.



The screenshot shows a Visual Studio Code editor with a file explorer on the left showing 'ELEVATELABS-TASK-5' containing 'deployment.yaml' and 'service.yaml'. The 'service.yaml' file is open, showing a Kubernetes Service manifest for 'nginx-service'. Below the editor, the 'TERMINAL' panel shows the following commands and output:

```
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl apply -f service.yaml
service/nginx-service created
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6fc7fc857f-k2b69   1/1     Running   0           91s
nginx-deployment-6fc7fc857f-rh4th   1/1     Running   0           91s
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl get svc
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes  ClusterIP   10.96.0.1     <none>        443/TCP          6m26s
nginx-service  NodePort    10.100.203.205 <none>        80:32551/TCP     13s
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> minikube ip
192.168.49.2
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl get svc
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes  ClusterIP   10.96.0.1     <none>        443/TCP          9m48s
nginx-service  NodePort    10.100.203.205 <none>        80:32551/TCP     3m35s
```

7. Access the Application:

<http://192.168.49.2:32551>

Get inside Minikube VM shell: minikube ssh

curl http://localhost:32551 inside Minikube works perfectly → nginx is reachable at NodePort 32551 inside the Minikube VM.

Exit Minikube shell : exit

The screenshot shows a VS Code editor with a file named `service.yaml` open. The file contains a Kubernetes `Service` definition for `nginx`. The terminal window shows the output of running `minikube ssh` and `curl` commands, displaying the HTML content served by the nginx service.

```
service.yaml: {} spec: {} selector: {} app
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: nginx-service
5 spec:
6   selector:
7     app: nginx
8   ports:
9     - protocol: TCP
10     port: 80
11     targetPort: 80
12   type: NodePort
13
```

```
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> minikube ssh
docker@minikube:~$ curl http://localhost:80
curl: (7) Failed to connect to localhost port 80 after 76 ms: Connection refused
docker@minikube:~$ curl http://localhost:32551
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
docker@minikube:~$ exit
logout
working. Further configuration is required.</p>
```

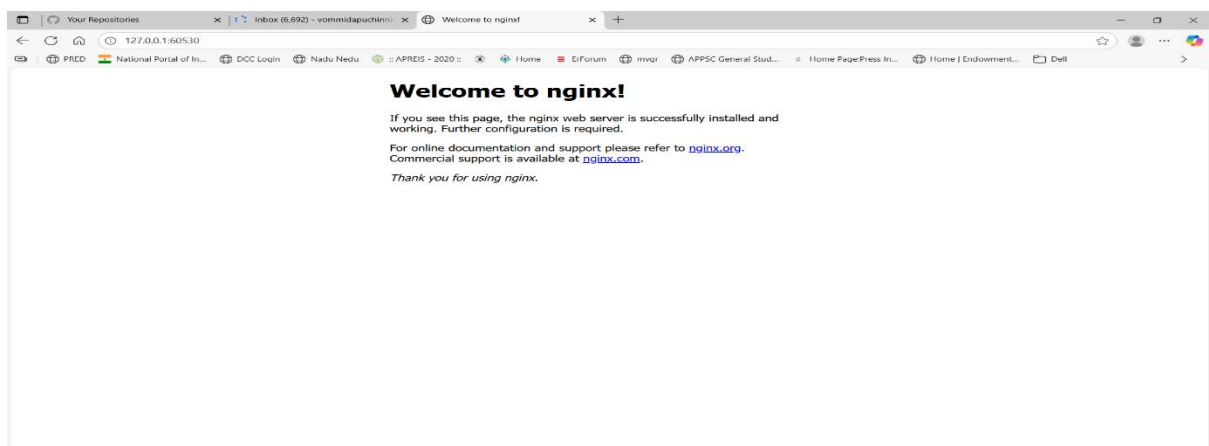
Alternate Quick Check

- Run: `minikube service nginx-service --url`
- It will print the full URL you can open in browser.

The screenshot shows the same VS Code editor with the `service.yaml` file. The terminal window now shows the output of running `minikube service nginx-service --url`, which prints the full URL `http://127.0.0.1:60530`.

```
service.yaml: {} spec: {} selector: {} app
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: nginx-service
5 spec:
6   selector:
7     app: nginx
8   ports:
9     - protocol: TCP
10     port: 80
11     targetPort: 80
12   type: NodePort
13
```

```
logout
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> minikube service nginx-service --url
http://127.0.0.1:60530
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

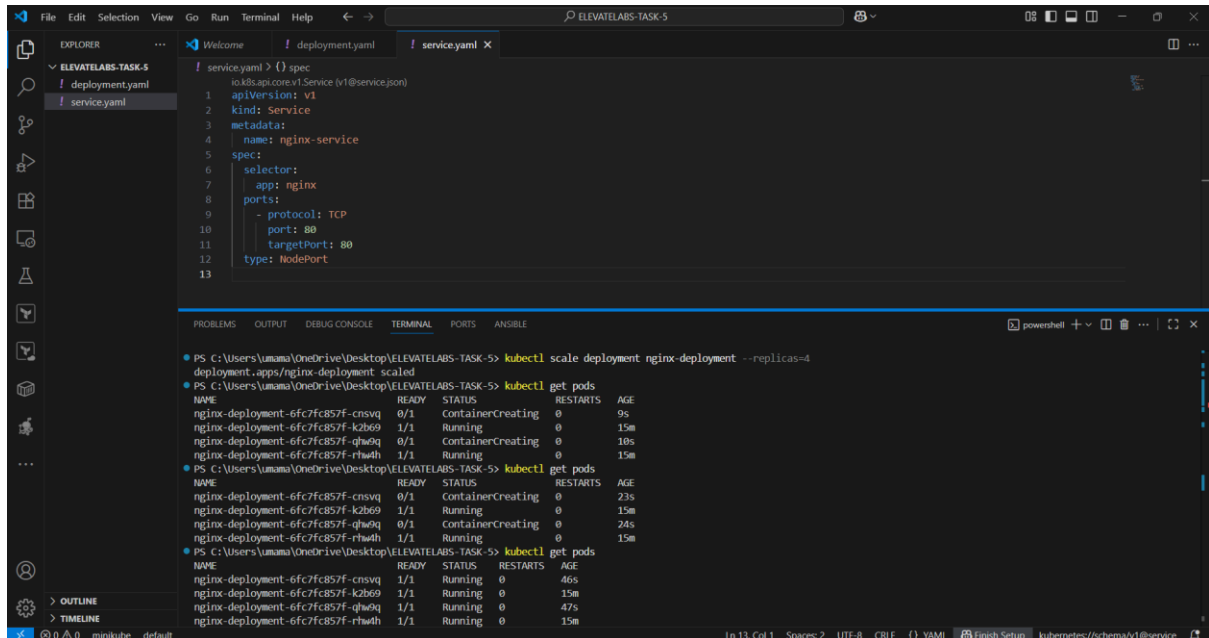


8. Scale Deployment

Scale pods from 2 to 4

kubectl scale deployment nginx-deployment --replicas=4

Check pods again: kubectl get pods



The screenshot shows a VS Code editor with a file explorer on the left containing 'deployment.yaml' and 'service.yaml'. The main editor displays the 'service.yaml' file with the following content:

```
1 service.yaml > {} spec
2 io.k8s.api.core.v1.Service (v1@service.json)
3   apiVersion: v1
4   kind: Service
5   metadata:
6     name: nginx-service
7   spec:
8     selector:
9       app: nginx
10    ports:
11      - protocol: TCP
12        port: 80
13        targetPort: 80
14    type: NodePort
```

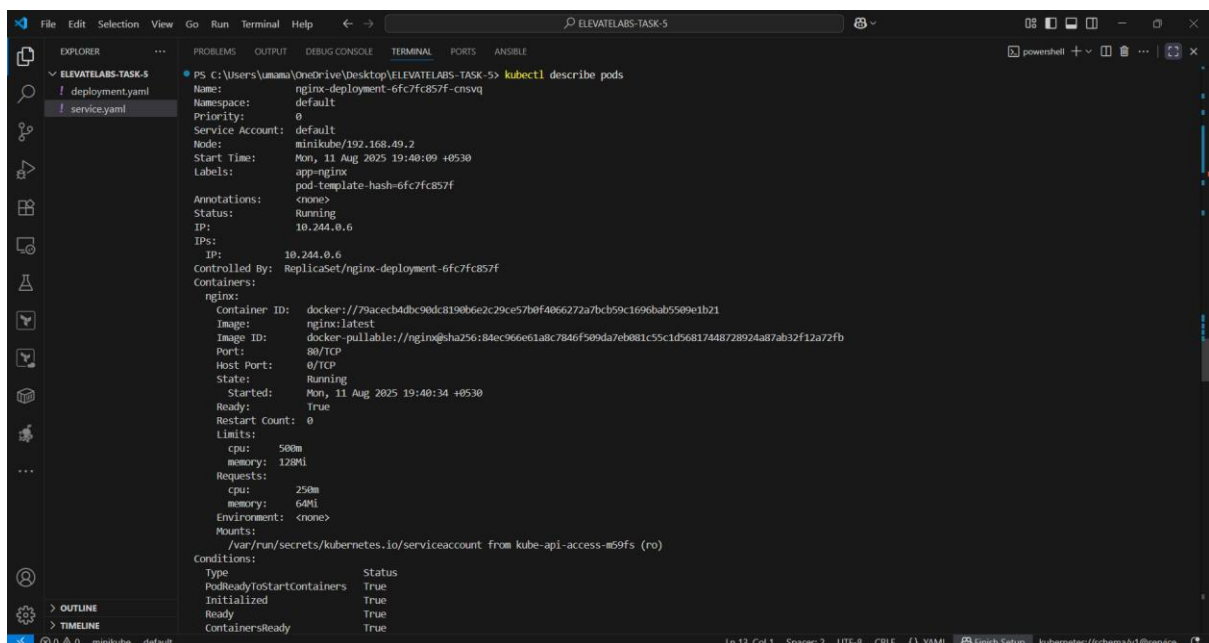
The terminal window shows the following commands and output:

```
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl scale deployment nginx-deployment --replicas=4
deployment.apps/nginx-deployment scaled
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6fc7fc857f-cnsvq   0/1     ContainerCreating   0           9s
nginx-deployment-6fc7fc857f-kzb69   1/1     Running           0           15m
nginx-deployment-6fc7fc857f-qhw9q   0/1     ContainerCreating   0           10s
nginx-deployment-6fc7fc857f-rh4th   1/1     Running           0           15m
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6fc7fc857f-cnsvq   0/1     ContainerCreating   0           23s
nginx-deployment-6fc7fc857f-kzb69   1/1     Running           0           15m
nginx-deployment-6fc7fc857f-qhw9q   0/1     ContainerCreating   0           24s
nginx-deployment-6fc7fc857f-rh4th   1/1     Running           0           15m
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6fc7fc857f-cnsvq   1/1     Running           0           46s
nginx-deployment-6fc7fc857f-kzb69   1/1     Running           0           15m
nginx-deployment-6fc7fc857f-qhw9q   1/1     Running           0           47s
nginx-deployment-6fc7fc857f-rh4th   1/1     Running           0           15m
```

9. Describe and Logs

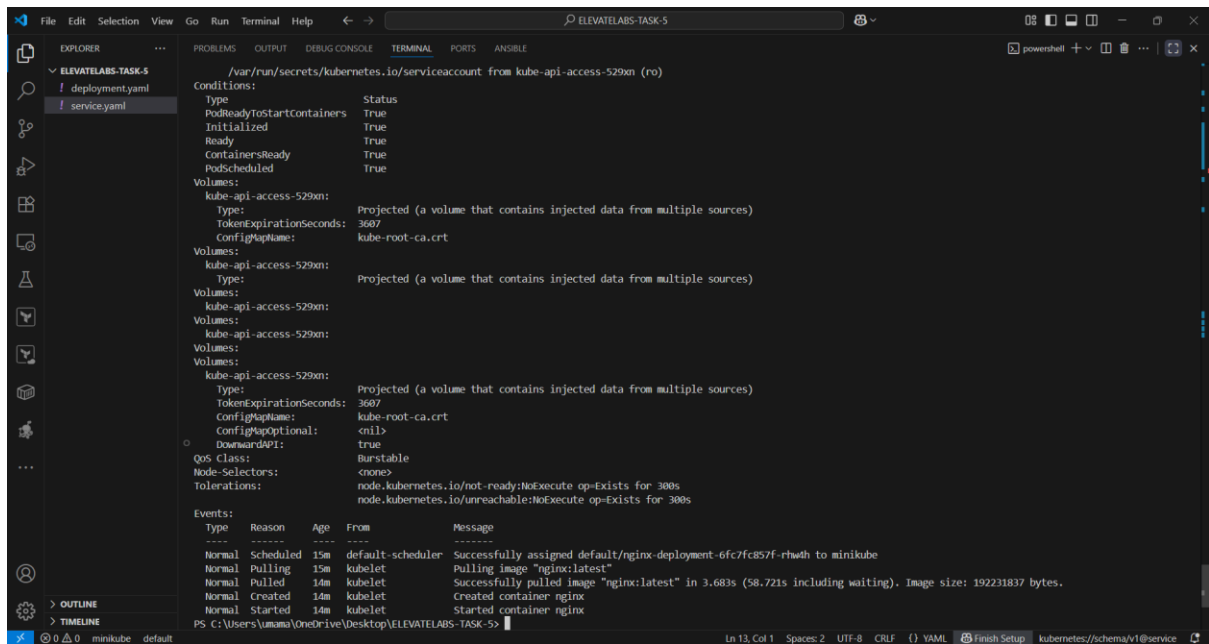
Describe pods for details: kubectl describe pods

Check logs of any pod: kubectl logs <pod-name>

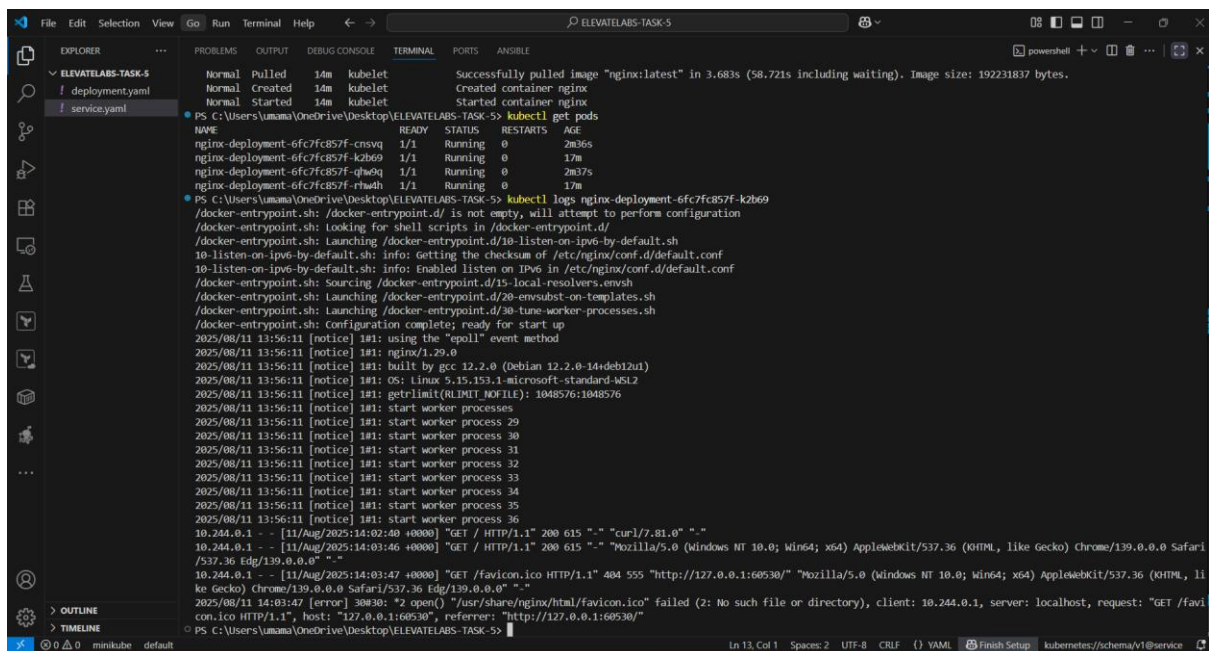


The screenshot shows the VS Code editor with the terminal window displaying the output of the command 'kubectl describe pods nginx-deployment-6fc7fc857f-cnsvq'. The output provides detailed information about the pod, including its namespace, priority, service account, node, start time, labels, annotations, status, IP, and the container details.

```
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl describe pods
Name: nginx-deployment-6fc7fc857f-cnsvq
Namespace: default
Priority: 0
Service Account: default
Node: minikube/192.168.49.2
Start Time: Mon, 11 Aug 2025 19:40:09 +0530
Labels: app=nginx, pod-template-hash=6fc7fc857f
Annotations: <none>
Status: Running
IP: 10.244.0.6
IPs: 10.244.0.6
Controlled By: ReplicaSet/nginx-deployment-6fc7fc857f
Containers:
  nginx:
    Container ID: docker://79aceb4db90dc8190b6e2c29ce57b0f4066272a7cb59c1696ab5509e1b21
    Image: nginx:latest
    Image ID: docker-pullable://nginx@sha256:84ec966e61a8c7846f509da7eb081c5c1d56817448728924a87ab32f12a72fb
    Port: 80/TCP
    Host Port: 0/TCP
    State: Running
    Started: Mon, 11 Aug 2025 19:40:34 +0530
    Ready: True
    Restart Count: 0
    Limits:
      cpu: 500m
      memory: 128Mi
    Requests:
      cpu: 250m
      memory: 64Mi
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-859fs (ro)
Conditions:
  Type              Status
  PodReadyToStartContainers  True
  Initialized         True
  Ready               True
  ContainersReady      True
```



Logs:



10. Rollback Deployment in Kubernetes

- Check Deployment Rollout History

Run this command to see the versions (revisions) of your deployment

kubectl rollout history deployment/nginx-deployment

- Make a change (optional) to create multiple revisions

For example, update the nginx image version

kubectl set image deployment/nginx-deployment nginx=nginx:1.19

- Then check rollout history again

kubectl rollout history deployment/nginx-deployment

- Rollback to a previous revision
- To rollback to the last working revision (usually previous one), run

kubectl rollout undo deployment/nginx-deployment

- Check rollout status
- After rollback, check if the deployment rolled out successfully

kubectl rollout status deployment/nginx-deployment

```
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl rollout history deployment/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1          <none>

PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl set image deployment/nginx-deployment nginx=nginx:1.19
deployment.apps/nginx-deployment image updated

PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl rollout history deployment/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1          <none>
2          nginx=nginx:1.19

PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl rollout undo deployment/nginx-deployment
deployment.apps/nginx-deployment rolled back

PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> kubectl rollout history deployment/nginx-deployment
deployment.apps/nginx-deployment
REVISION  CHANGE-CAUSE
1          <none>
2          nginx=nginx:1.19
3          nginx=nginx:latest
```

how can we come to original revision 1

Redeploy the original Deployment YAML (nginx:latest)

- Update your deployment back to the original image

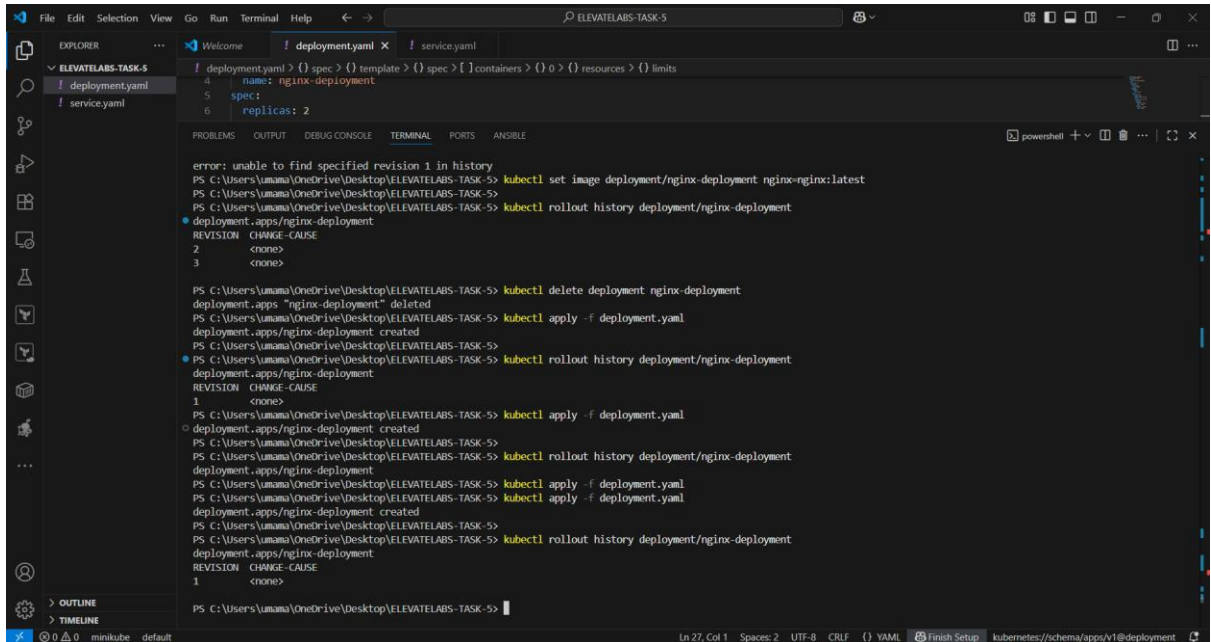
kubectl set image deployment/nginx-deployment nginx=nginx:latest

- This creates a new revision (e.g., revision 4) which is effectively your original revision 1 state.
- Verify rollout history: kubectl rollout history deployment/nginx-deployment

Delete and re-create deployment

If you want a completely clean slate

- `kubectl delete deployment nginx-deployment`
- `kubectl apply -f deployment.yaml`
- This deletes all revisions and creates a fresh deployment (which becomes revision 1 again).
- Use this if you want to reset everything.

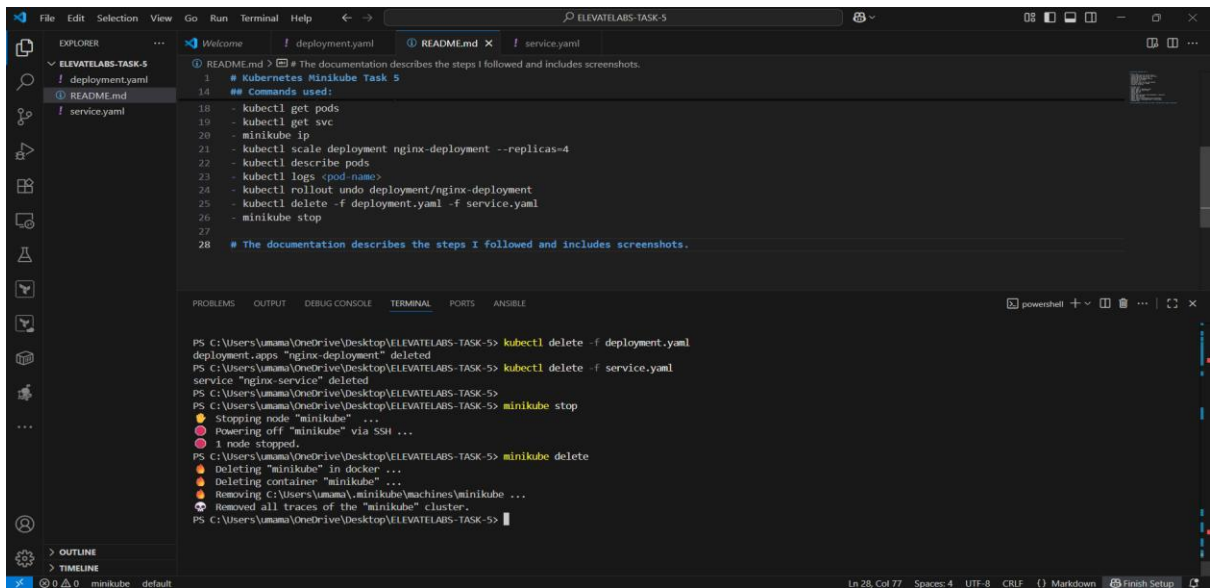


11.Cleanup

Delete deployment and service:

- `kubectl delete -f deployment.yaml`
- `kubectl delete -f service.yaml`

Stop minikube: minikube stop



12. Key Concepts

- **Pod:** Smallest deployable unit in Kubernetes.
- **Deployment:** Manages pods and ensures the desired state.
- **Service:** Exposes applications to the network.
- **Scaling:** Increasing/decreasing number of pods.
- **Rolling Update:** Gradually replacing old pods with new ones.
- **Rollback:** Restoring a previous version.

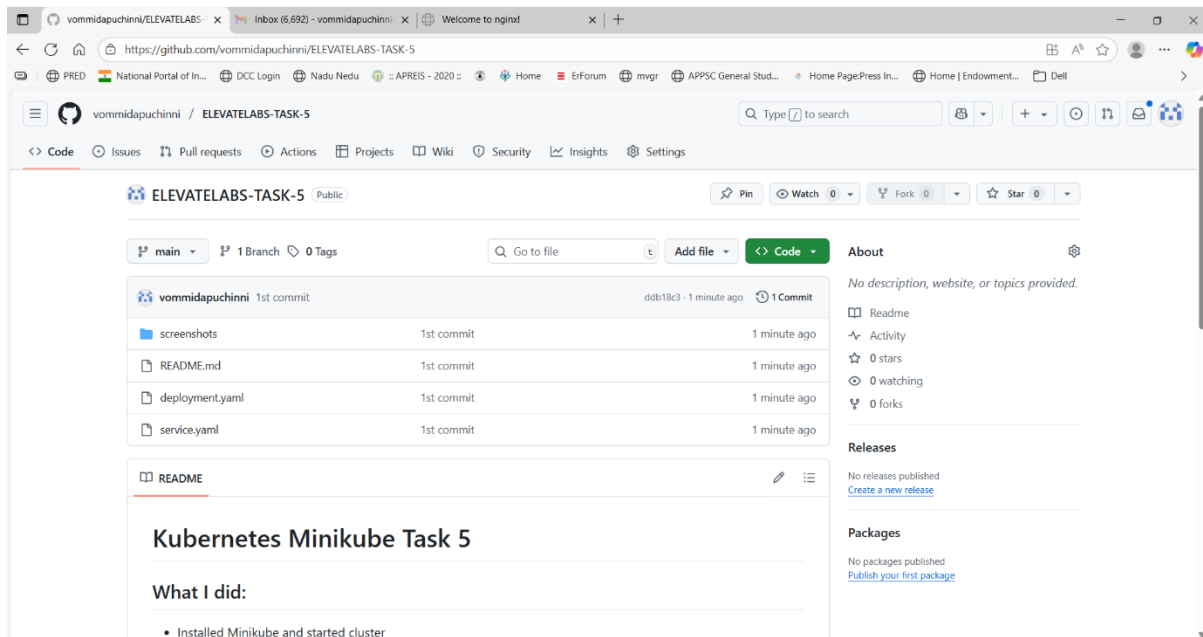
13. GitHub Repository Setup

Create GitHub Repository: Logged into GitHub → Created a new repository.

The screenshot shows the GitHub 'Create a new repository' page. The 'General' tab is active, showing the repository name 'ELEVATELABS-TASK-5' and a public visibility setting. The 'Configuration' tab is also visible, showing options for adding a README and a .gitignore file.

Push files to github

```
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> git init
Initialized empty Git repository in c:/Users/umama/OneDrive/Desktop/ELEVATELABS-TASK-5/.git/
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> git remote add origin https://github.com/vommidapuchinni/ELEVATELABS-TASK-5.git
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> git add .
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> git commit -m "1st commit"
[master (root-commit) db18c3] 1st commit
9 files changed, 66 insertions(+)
create mode 100644 README.md
create mode 100644 deployment.yaml
create mode 100644 screenshots/Application Access via browser.png
create mode 100644 screenshots/Application Access.png
create mode 100644 screenshots/Cleanup.png
create mode 100644 screenshots/Environment Setup.png
create mode 100644 screenshots/pods.png
create mode 100644 screenshots/service.png
create mode 100644 service.yaml
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5> git push origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 708.67 KiB | 5.41 MiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/vommidapuchinni/ELEVATELABS-TASK-5.git
 * [new branch] main -> main
PS C:\Users\umama\OneDrive\Desktop\ELEVATELABS-TASK-5>
```



14. Conclusion:

This task demonstrated how to:

Set up a Kubernetes cluster locally with Minikube

Deploy, scale, update, and rollback applications

Use GitHub for code and screenshot version control