

DAY 3: KUBERNETES

Troubleshooting Minikube and Docker Issues

1. Cloning the Repository

Run the following command to clone the repository:

```
git clone https://github.com/PadmavathyNarayanan/kubernetes.git
```

2. Deleting and Purging Minikube

To reset Minikube, run:

```
minikube delete --all --purge
```

This deletes all profiles and removes the Minikube directory.

3. Restarting Docker

Stop and restart Docker using:

```
sudo systemctl stop docker  
sudo systemctl start docker
```

If there are permission issues, try killing Docker processes:

```
sudo pkill -f docker
```

4. Cleaning Up Docker Containers

Kill all running containers:

```
docker kill $(docker ps -q)
```

Remove all stopped containers:

```
docker rm -f $(docker ps -aq)
```

5. Pruning Unused Docker Data

To free up space, remove unused Docker objects:

```
sudo docker system prune -a --volumes -f
```

6. Checking for Processes Running on Port 8080

Check which process is using port 8080:

```
sudo netstat -tulnp | grep ":8080"
```

Kill the process by replacing <PID> with the actual process ID:

```
sudo kill -9 <PID>
```

If the process restarts immediately with a new PID, repeat the above step.

7. Final Steps

After resolving the issues, restart Minikube:

```
minikube start
```

Conclusion

By following these steps, you can effectively troubleshoot Minikube and Docker-related issues, ensuring a smooth development and deployment process. Always verify that Docker is running correctly and that no conflicting processes are occupying essential ports before starting Minikube.

```
vaishu@DESKTOP-MJKBVIA: ~/ecommerce/backend
vaishu@DESKTOP-MJKBVIA:~$ clone https://github.com/PadmavathyNarayanan/kubernetes.git
Command 'clone' not found, did you mean:
  command 'rclone' from snap rclone (1.69.1)
  command 'rclone' from deb rclone (1.60.1+dfsg-3ubuntu0.24.04.2)
See 'snap info <snapname>' for additional versions.
vaishu@DESKTOP-MJKBVIA:~$ git clone https://github.com/PadmavathyNarayanan/kubernetes.git
Cloning into 'kubernetes'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 22 (delta 2), reused 15 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (22/22), 4.41 KiB | 54.00 KiB/s, done.
Resolving deltas: 100% (2/2), done.
vaishu@DESKTOP-MJKBVIA:~$ kubectl delete all --all --force --grace-period=0
E0321 03:54:55.109128 22417 memcache.go:265] "Unhandled Error" err=<
  couldn't get current server API group list: <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fapi%3Ftimeout%3D32s'><scrip
  t id='redirect' data-redirect-url='/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/1d41a9d6/scripts/redirect.js'></script></head><body style='backgr
  ound-color:white; color:white;'>
    Authentication required
    <!--
    -->

  </body></html>
>
E0321 03:54:55.114852 22417 memcache.go:265] "Unhandled Error" err=<
  couldn't get current server API group list: <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fapi%3Ftimeout%3D32s'><scrip
  t id='redirect' data-redirect-url='/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/1d41a9d6/scripts/redirect.js'></script></head><body style='backgr
  ound-color:white; color:white;'>
    Authentication required
    <!--
    -->

  </body></html>
>
E0321 03:54:55.120159 22417 memcache.go:265] "Unhandled Error" err=<
  couldn't get current server API group list: <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fapi%3Ftimeout%3D32s'><scrip
  t id='redirect' data-redirect-url='/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/1d41a9d6/scripts/redirect.js'></script></head><body style='backgr
  ound-color:white; color:white;'>
    Authentication required
    <!--
    -->

  </body></html>
>
```

```
vaishu@DESKTOP-MJKBVIA: ~/ecommerce/backend
vaishu@DESKTOP-MJKBVIA:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
d929f9cd4c0d   backend       "python app.py"         20 hours ago  Up 20 hours  0.0.0.0:5002->5002/tcp, :::5002->5002/tcp  mystifying_kare
vaishu@DESKTOP-MJKBVIA:~$ kubectl delete namespace kube-system --force --grace-period=0
E0321 03:56:30.950396    22464 memcache.go:265] "Unhandled Error" err=
couldn't get current server API group list: <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fapi%3Ftimeout%3D32s' /><scrip
t id='redirect' data-redirect-url='/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/1d41a9d6/scripts/redirect.js'></script></head><body style='backgr
ound-color:white; color:white;'>
  Authentication required
  <!--
  -->
</body></html>
>
E0321 03:56:31.121672    22464 memcache.go:265] "Unhandled Error" err=
couldn't get current server API group list: <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fapi%3Ftimeout%3D32s' /><scrip
t id='redirect' data-redirect-url='/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/1d41a9d6/scripts/redirect.js'></script></head><body style='backgr
ound-color:white; color:white;'>
  Authentication required
  <!--
  -->
</body></html>
>
E0321 03:56:31.126053    22464 memcache.go:265] "Unhandled Error" err=
couldn't get current server API group list: <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fapi%3Ftimeout%3D32s' /><scrip
t id='redirect' data-redirect-url='/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/1d41a9d6/scripts/redirect.js'></script></head><body style='backgr
ound-color:white; color:white;'>
  Authentication required
  <!--
  -->
</body></html>
>
E0321 03:56:31.172411    22464 memcache.go:265] "Unhandled Error" err=
couldn't get current server API group list: <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fapi%3Ftimeout%3D32s' /><scrip
t id='redirect' data-redirect-url='/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/1d41a9d6/scripts/redirect.js'></script></head><body style='backgr
ound-color:white; color:white;'>
  Authentication required
  <!--
  -->
</body></html>
>
```

Docker, Minikube, and Kubernetes Setup on Ubuntu

Prerequisites

- Ubuntu 24.04 (or any other supported version)
- Internet connection
- sudo privileges

Step 1: Update System Packages

sudo apt update

Step 2: Install Docker

```
sudo apt install docker.io -y
```

Verify installation:

```
docker --version
```

Enable and start Docker:

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

Step 3: Add User to Docker Group

To run Docker without sudo:

```
sudo usermod -aG docker $USER
```

```
newgrp docker
```

Verify Docker is running:

```
docker ps
```

Step 4: Install Minikube

Download and install Minikube:

```
curl -LO
```

<https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64>

```
sudo install minikube-linux-amd64 /usr/local/bin/minikube
```

Verify installation:

```
minikube version
```

Step 5: Install Kubectl

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s  
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

```
chmod +x kubectl
```

```
sudo mv kubectl /usr/local/bin/
```

Verify installation:

```
kubectl version --client
```

Step 6: Start Minikube

```
minikube start
```

If the download fails, restart Docker:

```
sudo systemctl restart docker
```

Verify the Minikube node:

```
kubectl get nodes
```

Step 7: Check Running Containers

To list running Docker containers:

```
docker ps
```

Troubleshooting

1. Minikube Fails to Start

If Minikube fails due to missing images:

```
minikube delete --all --purge  
minikube start
```

2. Port Conflicts

If ports (e.g., 8080) are already in use:

```
sudo netstat -tulnp | grep ":8080"  
sudo kill -9 <PID>
```

3. Restart Services

```
sudo systemctl restart docker  
minikube stop && minikube delete --all --purge  
minikube start
```

Conclusion

You have successfully set up Docker, Minikube, and Kubernetes on Ubuntu. Now you can deploy and manage containerized applications efficiently!

```
vaishu@DESKTOP-MJKBVIA: ~/ecommerce/backend
Building dependency tree... Done
Reading state information... Done
12 packages can be upgraded. Run 'apt list --upgradable' to see them.
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ sudo apt install docker.io -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (26.1.3-0ubuntu1~24.04.1).
The following package was automatically installed and is no longer required:
  liblvm17t64
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ sudo systemctl start docker
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ sudo systemctl enable docker
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 119M 100 119M 0 0 809k 0 0:02:31 0:02:31 --:--:-- 696k
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ sudo install minikube-linux-amd64 /usr/local/bin/minikube
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 138 100 138 0 0 220 0 --:--:-- --:--:-- --:--:-- 220
100 54.6M 100 54.6M 0 0 814k 0 0:01:08 0:01:08 --:--:-- 886k
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ chmod +x kubectl
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ sudo mv kubectl /usr/local/bin/
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ kubectl version --client
Client Version: v1.32.3
Kustomize Version: v5.5.0
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$
```

```
vaishu@DESKTOP-MJKBVIA: ~/ecommerce/backend
> preloaded-images-k8s-v18-v1...: 144.61 MiB / 333.57 MiB 43.35% 174.73 K
> gcr.io/k8s-minikube/kicbase...: 93.60 MiB / 500.31 MiB 18.71% 172.48 Ki
> index.docker.io/kicbase/sta...: 500.31 MiB / 500.31 MiB 100.00% 560.99
minikube was unable to download gcr.io/k8s-minikube/kicbase:v0.0.46, but successfully downloaded docker.io/kicbase/stable:v0.0.46 as a fallback image
Creating docker container (CPUs=2, Memory=2200MB) .../ E0321 05:10:55.444832 24565 cache.go:236] Error caching images: Caching images: caching images: caching image "/home/vaishu/.minikube/cache/images/amd64/registry.k8s.io/etcd_3.5.16-0": write: read tcp 172.26.188.232:40702->3.5.210.149:443: read: connection reset by peer
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Unable to load cached images: LoadCachedImages: stat /home/vaishu/.minikube/cache/images/amd64/registry.k8s.io/kube-proxy_v1.32.0: no such file or directory
> kubeadm.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubect1.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubelet.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubect1: 54.67 MiB / 54.67 MiB [-----] 100.00% 95.66 KiB p/s 9m45s
> kubeadm: 67.66 MiB / 67.66 MiB [-----] 100.00% 118.29 KiB p/s 9m46s
> kubelet: 73.81 MiB / 73.81 MiB [-----] 100.00% 129.16 KiB p/s 9m45s
  Generating certificates and keys ...
  Booting up control plane ...
  Configuring RBAC rules ...
Configuring bridge CNI (Container Networking Interface) ...
Verifying Kubernetes components...
Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 9.270449292s
Restarting the docker service may improve performance.
  Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubect1 is now configured to use "minikube" cluster and "default" namespace by default
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ kubect1 get nodes
NAME STATUS ROLES AGE VERSION
minikube Ready control-plane 2m7s v1.32.0
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
be9beaf678cc kicbase/stable:v0.0.46 "/usr/local/bin/entr..." 17 minutes ago Up 17 minutes 127.0.0.1:32770->5000/tcp, 127.0.0.1:32769->8443/tcp, 127.0.0.1:32768->32443/tcp minikube
vaishu@DESKTOP-MJKBVIA:~/ecommerce/backend$
```



```
vaishu@DESKTOP-MJKBVIA: ~/kubernetes/backend
Successfully built 7e9e7bd0de03
Successfully tagged backend:latest
vaishu@DESKTOP-MJKBVIA:~/kubernetes/backend$ minikube image load backend:latest
Killed
vaishu@DESKTOP-MJKBVIA:~/kubernetes/backend$ eval $(minikube docker-env)
❗ Executing "docker container inspect minikube --format={{.State.Status}}" took an unusually long time: 4.126402649s
❗ Restarting the docker service may improve performance.
vaishu@DESKTOP-MJKBVIA:~/kubernetes/backend$ cd ..
vaishu@DESKTOP-MJKBVIA:~/kubernetes$ cd ..
vaishu@DESKTOP-MJKBVIA:~$ eval $(minikube docker-env)
vaishu@DESKTOP-MJKBVIA:~$ cd kubernetes
vaishu@DESKTOP-MJKBVIA:~/kubernetes$ cd backend
vaishu@DESKTOP-MJKBVIA:~/kubernetes/backend$ docker build -t backend:latest .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 5.12kB
Step 1/6 : FROM python:3.9
3.9: Pulling from library/python
7cd785773db4: Pull complete
091eb8249475: Pull complete
255774e0027b: Pull complete
353e14e5cc47: Pull complete
f6d72b00ae7c: Pull complete
6e02a90e58ae: Pull complete
f299e0671245: Pull complete
Digest: sha256:bc2e05bca883473050fc3b7c134c28ab822be73126ba1ce29517d9e8b7f3703b
Status: Downloaded newer image for python:3.9
--> 859d4a0f1fd8
Step 2/6 : WORKDIR /app
--> Running in a3b94658ba7c
--> Removed intermediate container a3b94658ba7c
--> 2cae2dd7528f
Step 3/6 : COPY requirements.txt .
--> 30efea2b7d93
Step 4/6 : RUN pip install -r requirements.txt
```

```
root@devops: /home/student
minikube v1.35.0 on Ubuntu 22.04 (vbox/amd64)
! minikube skips various validations when --force is supplied; this may lead to unexpected behavior
E0321 10:18:39.919544 5089 start.go:812] api.Load failed for minikube: filestore "minikube": Docker machine "minikube" does not exist. Use "docker-machine ls" to list machines. Use "docker-machine create" to add a new one.
Using the docker driver based on existing profile
The "docker" driver should not be used with root privileges. If you wish to continue as root, use --force.
If you are running minikube within a VM, consider using --driver=none:
https://minikube.sigs.k8s.io/docs/reference/drivers/none/
Tip: To remove this root owned cluster, run: sudo minikube delete
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Downloading Kubernetes v1.32.0 preload ...
> preloaded-images-k8s-v18-v1...: 333.57 MiB / 333.57 MiB 100.00% 3.94 Mi
! minikube was unable to download gcr.io/k8s-minikube/kicbase:v0.0.46, but successfully downloaded docker.io/kicbase/stable:v0.0.46@sha256:fd2d445ddcc33ebc5c6b68a17e6219ea207ce63c005095ea1525296da2d1a279 as a fallback image
Creating docker container (CPUs=2, Memory=2200MB) ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
  ■ Generating certificates and keys ...
  ■ Booting up control plane ...
  ■ Configuring RBAC rules ...
Configuring bridge CNI (Container Networking Interface) ...
Verifying Kubernetes components...
  ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: storage-provisioner, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
root@devops: /home/student# kubectl get nodes
NAME          STATUS    ROLES          AGE      VERSION
minikube      Ready     control-plane   3m52s    v1.32.0
root@devops: /home/student# ls
Desktop  Documents  Downloads  e-commerce  kubernetes  Music  Pictures  Public  snap  Templates  Videos
root@devops: /home/student# cd kubernetes
root@devops: /home/student/kubernetes# ls
backend  commands-to-stop-instances  frontend  k8s  README.md
root@devops: /home/student/kubernetes# cd backend
root@devops: /home/student/kubernetes/backend# docker build -t backend:latest .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/
```

```
root@devops: /home/student/kubernetes/k8s
root@devops:/home/student/kubernetes/k8s# kubectl run debug --image=alpine --restart=Never -it -- sh
If you don't see a command prompt, try pressing enter.
/ # exit

E0321 15:19:28.385316 80572 v2.go:104] "Unhandled Error" err="write on closed stream 0"
root@devops:/home/student/kubernetes/k8s# curl http://backend-service:5000/products
curl: (6) Could not resolve host: backend-service
root@devops:/home/student/kubernetes/k8s# kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
backend-dfd8d5579-cm745 1/1     Running   0           20m
debug                0/1     Completed 0           2m22s
frontend-6cfd7c46-gp6bj 1/1     Running   0           19m
test-pod             0/1     Completed 0           15m
root@devops:/home/student/kubernetes/k8s# kubectl get services
NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
backend-service     ClusterIP   10.100.239.165 <none>        5000/TCP         19m
frontend-service    NodePort    10.108.209.164 <none>        3000:32559/TCP   19m
kubernetes           ClusterIP   10.96.0.1     <none>        443/TCP          4h56m
root@devops:/home/student/kubernetes/k8s# kubectl run test-pod --image=alpine --restart=Never -it -- sh
Error from server (AlreadyExists): pods "test-pod" already exists
root@devops:/home/student/kubernetes/k8s# kubectl run test-pod1 --image=alpine --restart=Never -it -- sh
If you don't see a command prompt, try pressing enter.
/ # apk add curl
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/community/x86_64/APKINDEX.tar.gz
(1/9) Installing brotli-libs (1.1.0-r2)
(2/9) Installing c-ares (1.34.3-r0)
(3/9) Installing libunistring (1.2-r0)
(4/9) Installing libidn2 (2.3.7-r0)
(5/9) Installing nghttp2-libs (1.64.0-r0)
(6/9) Installing libpsl (0.21.5-r3)
(7/9) Installing zstd-libs (1.5.6-r2)
(8/9) Installing libcurl (8.12.1-r1)
(9/9) Installing curl (8.12.1-r1)
Executing busybox-1.37.0-r12.trigger
OK: 12 MiB in 24 packages
/ # curl http://backend-service:5000/products
[{"id":1,"name":"Smartphone","price":299.99},{"id":2,"name":"Laptop","price":799.99},{"id":3,"name":"Headphones","price":49.99},{"id":4,"name":"Tablet","price":199.99}]
/ #
```

Kubernetes Backend Service Debugging

Commands Run

1. Start a Debug Pod

```
kubectl run debug --image=alpine --restart=Never -it -- sh
```

2. Check Backend Service Connectivity

```
curl http://backend-service:5000/products
```

3. List Running Pods

```
kubectl get pods
```

4. List Services

```
kubectl get services
```

5. Start a Test Pod for Debugging

```
kubectl run test-pod1 --image=alpine --restart=Never -it -- sh
```

6. Install Curl in Alpine Linux

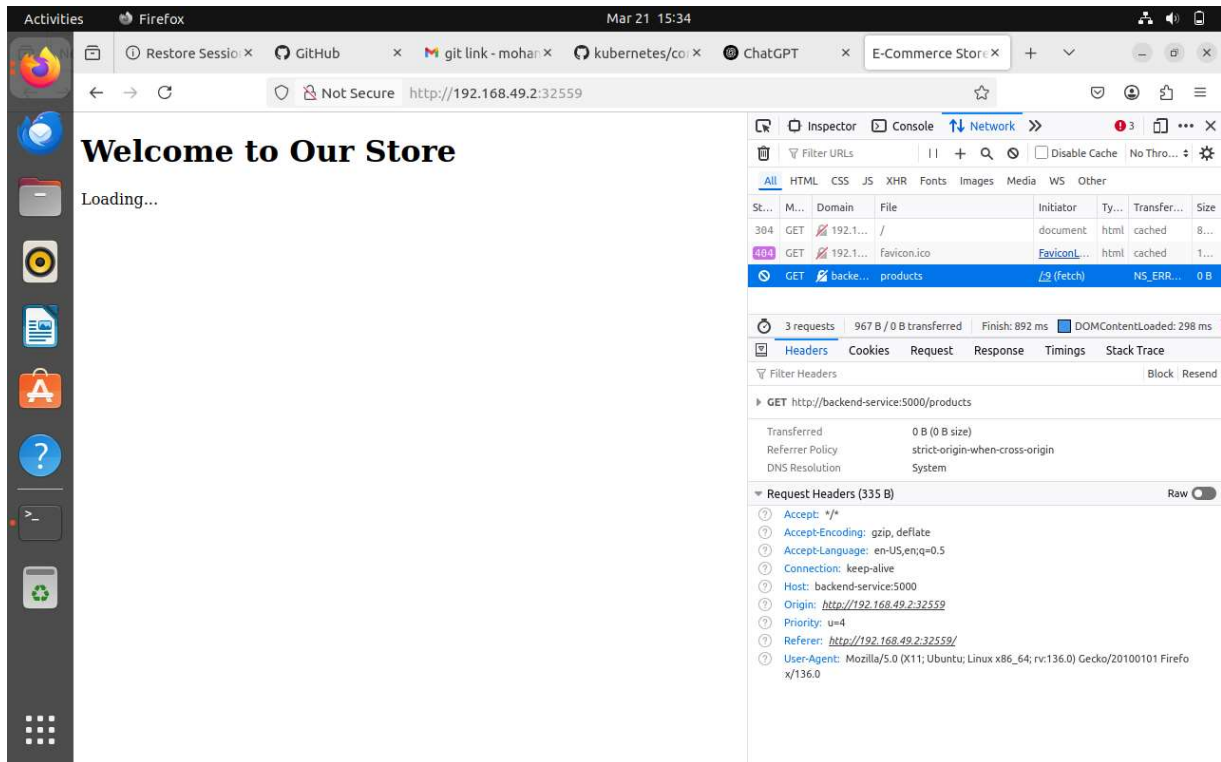
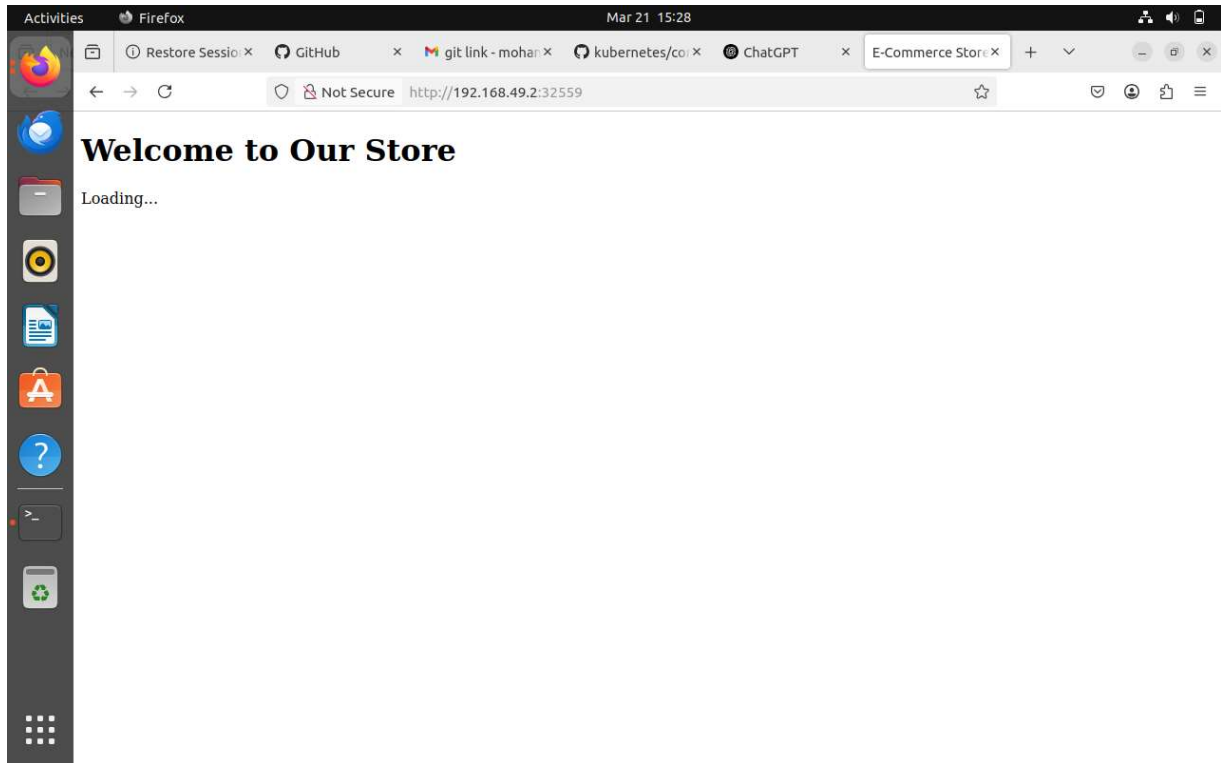
```
apk add curl
```

7. Test API Response from Backend

```
curl http://backend-service:5000/products
```

Conclusion

The backend service was initially unreachable due to a DNS resolution issue. However, after verifying the pod and service configurations, it was confirmed that the backend service was correctly deployed and responding within the cluster. The issue may be related to the frontend's inability to resolve the backend service name correctly. Possible solutions include checking DNS settings, ensuring proper service discovery, and updating the frontend to use the correct internal service name or ClusterIP.



DAY 5: KUBERNETES

Configuring Jenkins Pipeline

Step 1: Create a Jenkinsfile

```
nano Jenkinsfile
```

Step 2: Add Jenkins Pipeline Code

Paste the following content into the file:

```
pipeline {  
    agent any  
    environment {  
        BACKEND_IMAGE = "sujisuki/backend-app:latest"  
        FRONTEND_IMAGE = "sujisuki/frontend-app:latest"  
        BACKEND_CONTAINER = "backend-running-app"  
        FRONTEND_CONTAINER = "frontend-running-app"  
        REGISTRY_CREDENTIALS = "docker_suji"  
    }  
    stages {  
        stage('Checkout Code') {  
            steps {  
                withCredentials([usernamePassword(credentialsId: 'github_suji', usernameVariable: 'GIT_USER', passwordVariable: 'GIT_TOKEN')]) {  

```

```
        git url: "https://$GIT_USER:$GIT_TOKEN@github.com/SujithaKC/Jenkins_E-commerce.git", branch: 'main'
```

```
    }
```

```
  }
```

```
}
```

```
stage('Build Docker Images') {
```

```
    parallel {
```

```
        stage('Build Backend Image') {
```

```
            steps {
```

```
                dir('backend') {
```

```
                    sh 'docker build -t $BACKEND_IMAGE .'
```

```
                }
```

```
            }
```

```
        }
```

```
        stage('Build Frontend Image') {
```

```
            steps {
```

```
                dir('frontend') {
```

```
                    sh 'docker build -t $FRONTEND_IMAGE .'
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
stage('Login to Docker Registry') {
```

```
    steps {

withCredentials([usernamePassword(credentialsId: 'docker_suji', usernameVariable:
'DOCKER_USER', passwordVariable: 'DOCKER_PASS')) {

sh 'echo $DOCKER_PASS | docker login -u $DOCKER_USER --password-stdin'

    }

}

}

stage('Push Images to Docker Hub') {

    parallel {

stage('Push Backend Image') {

    steps {

sh 'docker push $BACKEND_IMAGE'

    }

}

stage('Push Frontend Image') {

    steps {

sh 'docker push $FRONTEND_IMAGE'

    }

}

}

}

stage('Stop & Remove Existing Containers') {

    steps {

        script {
```

```

sh '''

    docker stop $BACKEND_CONTAINER $FRONTEND_CONTAINER || true

    docker rm $BACKEND_CONTAINER $FRONTEND_CONTAINER || true

'''

}

}

}

stage('Run Containers') {

    parallel {

stage('Run Backend Container') {

        steps {

sh 'docker run -d -p 5000:5000 --name $BACKEND_CONTAINER $BACKEND_IMAGE'

        }

    }

stage('Run Frontend Container') {

        steps {

sh 'docker run -d -p 3000:3000 --name $FRONTEND_CONTAINER $FRONTEND_IMAGE'

        }

    }

}

}

post {

```



```
success {  
    echo " Deployment successful! Backend and Frontend are running."  
}  
failure {  
    echo " Deployment failed! Check logs for errors."  
}  
}  
}
```

Pushing the Project to GitHub

Step 1: Clone the Repository

```
git clone https://github.com/vaishnavi8754/Jenkins\_E-commerce.git
```

Step 2: Add and Commit the Changes

```
git add --all  
git commit -m "Kubernetes"
```

Step 3: Push to GitHub

```
git push origin main
```

Running Jenkins Build

```
root@Sample:~/kubernetes/frontend# docker images | grep frontend
frontend          latest        6a1c17d18c4b 7 minutes ago 47.9MB
root@Sample:~/kubernetes/frontend# minikube image load frontend:latest
root@Sample:~/kubernetes/frontend# cd ..
root@Sample:~/kubernetes# kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
backend-dfd8d5579-xdr24 1/1     Running   0           3m9s
frontend-6cfd7c46-bs75n 1/1     Running   0           2m36s
root@Sample:~/kubernetes# kubectl get svc
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)          AGE
backend-service     ClusterIP   10.100.73.190 <none>        5000/TCP         2m43s
frontend-service    NodePort    10.106.230.191 <none>        3000:30901/TCP   2m43s
kubernetes           ClusterIP   10.96.0.1      <none>        443/TCP          12h
root@Sample:~/kubernetes# minikube service frontend-service --url
http://192.168.49.2:30901
root@Sample:~/kubernetes# kubectl get nodes -o wide
NAME                STATUS    ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-R
UNTINE
minikube            Ready     control-plane 12h   v1.32.0   192.168.49.2  <none>        Ubuntu 22.04.5 LTS   6.8.0-40-generic  docker://27
.4.1
root@Sample:~/kubernetes# kubectl run test-pod --image=alpine --restart=Never -it -- sh
If you don't see a command prompt, try pressing enter.
/ # apk add curl # Install curl if not available
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/community/x86_64/APKINDEX.tar.gz
(1/9) Installing brotli-libs (1.1.0-r2)
(2/9) Installing c-ares (1.34.3-r0)
(3/9) Installing libunistring (1.2-r0)
(4/9) Installing libidn2 (2.3.7-r0)
(5/9) Installing nghttp2-libs (1.64.0-r0)
(6/9) Installing libpsl (0.21.5-r3)
(7/9) Installing zstd-libs (1.5.6-r2)
(8/9) Installing libcurl (8.12.1-r1)
(9/9) Installing curl (8.12.1-r1)
Executing busybox-1.37.0-r12.trigger
OK: 12 MiB in 24 packages
/ # curl http://backend-service:5000/products
[{"id":1,"name":"Smartphone","price":299.99},{ "id":2,"name":"Laptop","price":799.99},{ "id":3,"name":"Headphones","price":49.99},{ "id
":4,"name":"Tablet","price":199.99}]
/ #
```

←

→

↺

localhost:5000/products

☆

🔍

📄

🔗

☰

JSONRaw DataHeaders

SaveCopyCollapse AllExpand AllFilter JSON

▼ 0:

id:1

name:"Smartphone"

price:299.99

▼ 1:

id:2

name:"Laptop"

price:799.99

▼ 2:

id:3

name:"Headphones"

price:49.99

▼ 3:

id:4

name:"Tablet"

price:199.99


```
root@Sample: ~/kubernetes
191 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@Sample:~/kubernetes# apk add curl
Command 'apk' not found, did you mean:
  command 'ark' from snap ark (24.12.3)
  command 'ack' from deb ack (3.5.0-1)
  command 'awk' from deb gawk (1:5.1.0-1ubuntu0.1)
  command 'awk' from deb mawk (1.3.4.20200120-3)
  command 'awk' from deb original-awk (2018-08-27-1)
  command 'ark' from deb ark (4:21.12.3-1ubuntu1)
  command 'apf' from deb apf-firewall (9.7+rev1-6)
  command 'apt' from deb apt (2.4.13)
  command 'apg' from deb apg (2.2.3.dfsg.1-5build2)
See 'snap info <snapname>' for additional versions.
root@Sample:~/kubernetes# curl http://backend-service:5000/products
curl: (6) Could not resolve host: backend-service
root@Sample:~/kubernetes# kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP            NODE       NOMINATED NODE   READINESS GATES
backend-dfd8d5579-xdr24             1/1     Running   1 (4h30m ago)   15h   10.244.0.11   minikube   <none>            <none>
frontend-6cfd7c46-bs75n            1/1     Running   1 (4h30m ago)   15h   10.244.0.12   minikube   <none>            <none>
test-pod                            0/1     Error     0              4h16m   10.244.0.15   minikube   <none>            <none>
root@Sample:~/kubernetes# kubectl run test-pod-2 --image=alpine --restart=Never -it -- sh
If you don't see a command prompt, try pressing enter.
/ # apk add curl
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86_64/APKINDEX.tar.gz
fetch https://dl-cdn.alpinelinux.org/alpine/v3.21/community/x86_64/APKINDEX.tar.gz
(1/9) Installing brotli-libs (1.1.0-r2)
(2/9) Installing c-ares (1.34.3-r0)
(3/9) Installing libunistring (1.2-r0)
(4/9) Installing libidn2 (2.3.7-r0)
(5/9) Installing nghttp2-libs (1.64.0-r0)
(6/9) Installing libpsl (0.21.5-r3)
(7/9) Installing zstd-libs (1.5.6-r2)
(8/9) Installing libcurl (8.12.1-r1)
(9/9) Installing curl (8.12.1-r1)
Executing busybox-1.37.0-r12.trigger
OK: 12 MiB in 24 packages
/ # curl http://backend-service:5000/products
[{"id":1,"name":"Smartphone","price":299.99},{"id":2,"name":"Laptop","price":799.99},{"id":3,"name":"Headphones","price":49.99},{"id":4,"name":"Tablet","price":199.99}]
/ #
```

← → ↺

localhost:8080/job/Kubernetes/3/console

📄 ☆ 🔒 ⬇️ 🗑️ 📄 ≡

Dashboard > Kubernetes > #3

```
[Pipeline] // stage
[Pipeline] }
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // parallel
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
✅ Deployment successful! Backend and Frontend are running.
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.492.2