

# Devops day 3

```
backend
vboxuser@Ubuntu:~/kubernetes/backend$ m
! The image 'backend:latest' was not found; unable to add it to cache.
vboxuser@Ubuntu:~/kubernetes/backend$ docker buildx build --load -t backend:latest .
unknown flag: --load
See 'docker --help'.

Usage: docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
run          Create and run a new container from an image
exec         Execute a command in a running container
ps           List containers
build        Build an image from a Dockerfile
pull         Download an image from a registry
push         Upload an image to a registry
images       List images
login        Log in to a registry
logout       Log out from a registry
search       Search Docker Hub for images
version      Show the Docker version information
info         Display system-wide information

Management Commands:
builder      Manage builds
container    Manage containers
context      Manage contexts
image        Manage images
manifest     Manage Docker image manifests and manifest lists
network      Manage networks
plugin       Manage plugins
system       Manage Docker
trust        Manage trust on Docker images
volume       Manage volumes

Swarm Commands:
swarm        Manage Swarm
```

Screenshot captured  
You can paste the image from the clipboard.

```
vboxuser@Ubuntu: ~/kubernetes
vboxuser@Ubuntu:~/kubernetes$ cd backend
vboxuser@Ubuntu:~/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
Digest: sha256:bc2e05bca883473050fc3b7c134c28ab822be73126ba1ce29517d9e8b7f3703b
Status: Downloaded newer image for python:3.9
--> 859d4a0f1fd8
Step 2/6 : WORKDIR /app
--> Using cache
--> cf2352738c97
Step 3/6 : COPY requirements.txt .
--> Using cache
--> 8c885ea43be9
Step 4/6 : RUN pip install -r requirements.txt
--> Using cache
--> ead79da3259e
Step 5/6 : COPY . .
--> Using cache
--> 9a97d433dad9
Step 6/6 : CMD ["python", "app.py"]
--> Using cache
--> db8420a4881e
Successfully built db8420a4881e
Successfully tagged backend:latest
vboxuser@Ubuntu:~/kubernetes/backend$ docker images | grep backend
backend        latest         db8420a4881e   4 hours ago    1.19GB
vboxuser@Ubuntu:~/kubernetes/backend$ minikube image load backend:latest
! The image 'backend:latest' was not found; unable to add it to cache.
vboxuser@Ubuntu:~/kubernetes/backend$ docker image

Usage: docker image COMMAND

Manage images

Commands:
```

```
vboxuser@Ubuntu: ~/kubernetes
vboxuser@Ubuntu:~/kubernetes$ sudo systemctl start docker
sudo systemctl enable docker
[sudo] password for vboxuser:
vboxuser@Ubuntu:~/kubernetes$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1-22.04.1
vboxuser@Ubuntu:~/kubernetes$ minikube start
🐳 minikube v1.35.0 on Ubuntu 22.04 (vbox/amd64)
🔧 Using the docker driver based on existing profile

💡 The requested memory allocation of 1967MiB does not leave room for system overhead (total system memory: 1967MiB). You may face stability issues.
💡 Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1967mb'

👉 Starting "minikube" primary control-plane node in "minikube" cluster
📶 Pulling base image v0.0.46 ...
🔄 Restarting existing docker container for "minikube" ...
📦 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔍 Verifying Kubernetes components...
   ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
❗ The image 'backend:latest' was not found; unable to add it to cache.
🔧 Enabled addons: default-storageclass, storage-provisioner
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
vboxuser@Ubuntu:~/kubernetes$ minikube start
🐳 minikube v1.35.0 on Ubuntu 22.04 (vbox/amd64)
🔧 Using the docker driver based on existing profile

💡 The requested memory allocation of 1967MiB does not leave room for system overhead (total system memory: 1967MiB). You may face stability issues.
💡 Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1967mb'

👉 Starting "minikube" primary control-plane node in "minikube" cluster
📶 Pulling base image v0.0.46 ...
🔄 Updating the running docker "minikube" container ...
📦 Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
🔍 Verifying Kubernetes components...
   ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
❗ The image 'backend:latest' was not found; unable to add it to cache.
🔧 Enabled addons: storage-provisioner, default-storageclass
🎉 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
vboxuser@Ubuntu:~/kubernetes$ eval $(minikube docker-env)

---> a6010cc470a0
successfully built a6010cc470a0
successfully tagged backend:latest
vboxuser@Ubuntu:~/kubernetes/backend$ cd ..
vboxuser@Ubuntu:~/kubernetes$ minikube image load frontend:latest
vboxuser@Ubuntu:~/kubernetes$ docker images | grep backend
backend          latest      a6010cc470a0  42 minutes ago  1.19GB
vboxuser@Ubuntu:~/kubernetes$ docker images | grep frontend
frontend         latest      64eccb4115ef  41 minutes ago  47.9MB
vboxuser@Ubuntu:~/kubernetes$ kubectl apply -f k8s/backend-deployment.yaml
deployment.apps/backend unchanged
vboxuser@Ubuntu:~/kubernetes$ kubectl apply -f k8s/frontend-deployment.yaml
deployment.apps/frontend unchanged
vboxuser@Ubuntu:~/kubernetes$ kubectl apply -f k8s/service.yaml
service/backend-service unchanged
service/frontend-service unchanged
vboxuser@Ubuntu:~/kubernetes$ kubectl apply -f k8s/configmap.yaml
configmap/backend-config unchanged
vboxuser@Ubuntu:~/kubernetes$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
backend-dfd8d5579-k86t5  0/1     ErrImageNeverPull  0          35m
frontend-6cfd7c46-z7jqm  1/1     Running    2 (4m36s ago)  34m
vboxuser@Ubuntu:~/kubernetes$ kubectl get svc
NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
backend-service     ClusterIP   10.102.215.206 <none>        5000/TCP        34m
frontend-service    NodePort    10.101.198.192 <none>        3000:32759/TCP  34m
kubernetes           ClusterIP   10.96.0.1     <none>        443/TCP          59m
vboxuser@Ubuntu:~/kubernetes$ minikube service frontend-service --url
http://192.168.49.2:32759
vboxuser@Ubuntu:~/kubernetes$ kubectl get nodes -o wide
NAME     STATUS   ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-ENGINE
minikube Ready    control-plane  59m   v1.32.0   192.168.49.2 <none>        Ubuntu 22.04.5 LTS   6.8.0-52-generic   docker://4.1
vboxuser@Ubuntu:~/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
etch https://dl-cdn.alpinelinux.org/alpine/v3.21/main/x86_64/APKINDEX.tar.gz
```

```
vboxuser@Ubuntu: ~/kubernetes

-RUNTIME
minikube Ready control-plane 5h49m v1.32.0 192.168.49.2 <none> Ubuntu 22.04.5 LTS 6.8.0-52-generic docker://
27.4.1
vboxuser@Ubuntu:~/kubernetes$ kubectl run test-pod --image=alpine --restart=Never -it -- sh
Error from server (AlreadyExists): pods "test-pod" already exists
vboxuser@Ubuntu:~/kubernetes$ kubectl run test-pod --image=alpine --restart=Never -it -- sh
Error from server (AlreadyExists): pods "test-pod" already exists
^C
vboxuser@Ubuntu:~/kubernetes$ kubectl get pods
NAME READY STATUS RESTARTS AGE
backend-dfd8d5579-k86t5 1/1 Running 2 (33m ago) 5h27m
frontend-6cfd7c46-z7jqm 1/1 Running 5 (33m ago) 5h27m
test-pod 0/1 Error 0 4h52m
vboxuser@Ubuntu:~/kubernetes$ kubectl delete pod test-pod
pod "test-pod" deleted
vboxuser@Ubuntu:~/kubernetes$ kubectl get pods --watch
NAME READY STATUS RESTARTS AGE
backend-dfd8d5579-k86t5 1/1 Running 2 (33m ago) 5h28m
frontend-6cfd7c46-z7jqm 1/1 Running 5 (33m ago) 5h27m
^Cvboxuser@Ubuntu:~/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
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}]
/ # exit
^Cvboxuser@Ubuntu:~/kubernetes$

vboxuser@Ubuntu:~/kubernetes$ cd k8s
vboxuser@Ubuntu:~/kubernetes/k8s$ kubectl apply -f k8s/frontend-deployment.yaml
Error: the path "k8s/frontend-deployment.yaml" does not exist
vboxuser@Ubuntu:~/kubernetes/k8s$ ls
allow-all.yaml configmap.yaml service.yaml
backend-deployment.yaml frontend-deployment.yaml
vboxuser@Ubuntu:~/kubernetes/k8s$ cd frontend-deployment.yaml
ash: cd: frontend-deployment.yaml: No such file or directory
vboxuser@Ubuntu:~/kubernetes/k8s$ cd frontend-deployment.yaml
ash: cd: frontend-deployment.yaml: Not a directory
vboxuser@Ubuntu:~/kubernetes/k8s$ cd ^C
vboxuser@Ubuntu:~/kubernetes/k8s$ cd frontend-deployment.yaml
ash: cd: frontend-deployment.yaml: Not a directory
vboxuser@Ubuntu:~/kubernetes/k8s$ ls
allow-all.yaml configmap.yaml service.yaml
backend-deployment.yaml frontend-deployment.yaml
vboxuser@Ubuntu:~/kubernetes/k8s$ cd frontend-deployment.yaml
ash: cd: frontend-deployment.yaml: Not a directory
vboxuser@Ubuntu:~/kubernetes/k8s$ ls
allow-all.yaml configmap.yaml service.yaml
backend-deployment.yaml frontend-deployment.yaml
vboxuser@Ubuntu:~/kubernetes/k8s$ cd ..
vboxuser@Ubuntu:~/kubernetes$ cd k8s/frontend-deployment.yaml
ash: cd: k8s/frontend-deployment.yaml: Not a directory
vboxuser@Ubuntu:~/kubernetes$ ls
backend frontend README.md
commands-to-execute k8s
commands-to-stop-instances pre-requisites-command
vboxuser@Ubuntu:~/kubernetes$ cd k8s
vboxuser@Ubuntu:~/kubernetes/k8s$ ls
allow-all.yaml configmap.yaml service.yaml
backend-deployment.yaml frontend-deployment.yaml
vboxuser@Ubuntu:~/kubernetes/k8s$ cd frontend-deployment.yaml
ash: cd: frontend-deployment.yaml: Not a directory
vboxuser@Ubuntu:~/kubernetes/k8s$ ls
allow-all.yaml configmap.yaml service.yaml
backend-deployment.yaml frontend-deployment.yaml
vboxuser@Ubuntu:~/kubernetes/k8s$ kubectl apply -f k8s/frontend-deployment.yaml
```

```
vboxuser@Ubuntu: ~/kubernetes

Username: siva182004
Labels:
  provider=docker
  Experimental: false
  Insecure Registries:
    10.96.0.0/12
    127.0.0.0/8
  Live Restore Enabled: false

vboxuser@Ubuntu:~/kubernetes/backend$ docker images ls
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
No images found matching "ls": did you mean "docker image ls"?
vboxuser@Ubuntu:~/kubernetes/backend$ docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
backend             latest          db8420a4881e    4 hours ago    1.19GB
frontend            latest          64ecb4115ef     5 hours ago    47.9MB
registry.k8s.io/kube-apiserver    v1.32.0        c2e17b8d0f4a    3 months ago    97MB
registry.k8s.io/kube-scheduler    v1.32.0        a389e107f4ff    3 months ago    69.6MB
registry.k8s.io/kube-controller-manager v1.32.0        8cab3d2a8bd0    3 months ago    89.7MB
registry.k8s.io/kube-proxy        v1.32.0        040f9f8aac8c    3 months ago    94MB
python               3.9            859d4a0f1fd8    3 months ago    999MB
registry.k8s.io/etcd             3.5.16-0       a9e7e6b294ba    6 months ago    150MB
registry.k8s.io/coredns/coredns  v1.11.3        c69fa2e9cbf5    7 months ago    61.8MB
registry.k8s.io/pause            3.10           873ed7510279    10 months ago   736kB
gcr.io/k8s-minikube/storage-provisioner v5             6e38f40d628d    3 years ago     31.5MB
vboxuser@Ubuntu:~/kubernetes/backend$ cd
vboxuser@Ubuntu:~$ cd kubernetes
vboxuser@Ubuntu:~/kubernetes$ ls
backend      frontend      README.md
commands-to-execute      k8s
commands-to-stop-instances pre-requisites-command
vboxuser@Ubuntu:~/kubernetes$ cd frontend
vboxuser@Ubuntu:~/kubernetes/frontend$ minikube image load frontend:latest
! The image 'frontend:latest' was not found; unable to add it to cache.
vboxuser@Ubuntu:~/kubernetes/frontend$ docker build -t frontend: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/

vboxuser@Ubuntu:~/kubernetes$
https://docs.docker.com/go/buildx/

ending build context to Docker daemon 3.584kB
step 1/2 : FROM nginx:alpine
alpine: Pulling from library/nginx
18232174bc9: Pull complete
cc35e35d420: Pull complete
3f2ec460bdf: Pull complete
84583bcf083: Pull complete
d27c072a58f: Pull complete
b3286a73463: Pull complete
d79cc6084d4: Pull complete
c7e4c092ab7: Pull complete
Digest: sha256:4ff102c5d78d254a6f0da062b3cf39eaf07f01eec0927fd21e219d0af8bc0591
Status: Downloaded newer image for nginx:alpine
--> 1ff4bb4faebc
step 2/2 : COPY index.html /usr/share/nginx/html/index.html
--> 64ecb4115ef
Successfully built 64ecb4115ef
Successfully tagged frontend:latest
vboxuser@Ubuntu:~/kubernetes/frontend$ docker images | grep frontend
frontend            latest          64ecb4115ef     11 seconds ago    47.9MB
vboxuser@Ubuntu:~/kubernetes/frontend$ minikube image load frontend:latest
vboxuser@Ubuntu:~/kubernetes/frontend$ kubectl apply -f k8s/backend-deployment.yaml
Error: the path "k8s/backend-deployment.yaml" does not exist
vboxuser@Ubuntu:~/kubernetes/frontend$ kubectl apply -f k8s/frontend-deployment.yaml
Error: the path "k8s/frontend-deployment.yaml" does not exist
vboxuser@Ubuntu:~/kubernetes/frontend$ ls
Dockerfile index.html
vboxuser@Ubuntu:~/kubernetes/frontend$ cd..
d.: command not found
vboxuser@Ubuntu:~/kubernetes/frontend$ cd ..
vboxuser@Ubuntu:~/kubernetes$ ls
backend      frontend      README.md
commands-to-execute      k8s
commands-to-stop-instances pre-requisites-command
vboxuser@Ubuntu:~/kubernetes$ cd k8s
vboxuser@Ubuntu:~/kubernetes/k8s$ kubectl apply -f k8s/frontend-deployment.yaml
```





```

/body></html>
vboxuser@Ubuntu:~/kubernetes$ minikube start
minikube v1.35.0 on Ubuntu 22.04 (vbox/amd64)
0321 14:26:43.704175    5539 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 requested memory allocation of 1967MiB does not leave room for system overhead (total system memory: 1967MiB). You may face stability issues.
💡 Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1967mb'

👉 Starting "minikube" primary control-plane node in "minikube" cluster
📡 Pulling base image v0.0.46 ...
🔗 Creating docker container (CPUs=2, Memory=1967MB) ...
🔧 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! kubectrl is now configured to use "minikube" cluster and "default" namespace by default
vboxuser@Ubuntu:~/kubernetes$ kubectrl get nodes
NAME        STATUS   ROLES    AGE   VERSION
minikube    Ready    control-plane   27s   v1.32.0
vboxuser@Ubuntu:~/kubernetes$ sudo apt update
gn:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
it:2 https://pkg.jenkins.io/debian-stable binary/ Release
it:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
it:5 http://in.archive.ubuntu.com/ubuntu jammy InRelease
it:6 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease
it:7 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 packages can be upgraded. Run 'apt list --upgradable' to see them.
vboxuser@Ubuntu:~/kubernetes$ sudo apt install docker.io -y
vboxuser@Ubuntu: ~
❌ Exiting due to RSRC_OVER_ALLOC_MEM: Requested memory allocation 2048MB is more than your system limit 1967MB.
💡 Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1967mb'

vboxuser@Ubuntu:~$ minikube start --cpus=1 --memory=1800mb --force
minikube v1.35.0 on Ubuntu 22.04 (vbox/amd64)
minikube skips various validations when --force is supplied; this may lead to unexpected behavior
👉 Automatically selected the docker driver

❌ Requested cpu count 1 is less than the minimum allowed of 2

❌ Requested memory allocation (1800MB) is less than the recommended minimum 1900MB. Deployments may fail.

! The requested memory allocation of 1800MiB does not leave room for system overhead (total system memory: 1967MiB). You may face stability issues.
💡 Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1967mb'

👉 Using Docker driver with root privileges
👉 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% 2.49 Mi
   > gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 2.72 Mi
🔗 Creating docker container (CPUs=1, Memory=1800MB) ...
🔗 Failing to connect to https://registry.k8s.io/ from inside the minikube container
🔗 To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
🔧 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: default-storageclass, storage-provisioner
👉 Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default
vboxuser@Ubuntu:~$ kubectrl get nodes
NAME        STATUS   ROLES    AGE   VERSION
minikube    Ready    control-plane   60s   v1.32.0
vboxuser@Ubuntu:~$

```

```

vboxuser@Ubuntu:~$ ls
Desktop      Documents  fetchData  Pictures    snap        Videos
docker-python-app  Downloads  Music      Public      Templates
vboxuser@Ubuntu:~$ su -
Password:
root@Ubuntu:~# ls
kubernetes  minikube-linux-amd64  snap  vboxpostinstall.sh
root@Ubuntu:~# cd kubernetes
root@Ubuntu:~/kubernetes# ls
backend  commands-to-stop-instances  frontend  k8s  README.md
root@Ubuntu:~/kubernetes# cd backend
root@Ubuntu:~/kubernetes/backend# ls
app.py  dockerfile  products.csv  requirements.txt
root@Ubuntu:~/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/

"docker build" requires exactly 1 argument.
See 'docker build --help'.

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
root@Ubuntu:~/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: Extracting [=====] 173.8MB/211.4MB
f6d72b00ae7c: Download complete
6e02a90e58ae: Download complete
f299e0671245: Download complete

```

```

vboxuser@Ubuntu:~/kubernetes$ git clone https://github.com/PadmavathyNarayanan/kubernetes.git
Cloning into 'kubernetes'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 34 (delta 7), reused 15 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (34/34), 8.67 KiB | 1.73 MiB/s, done.
Resolving deltas: 100% (7/7), done.
vboxuser@Ubuntu:~$ ls
Desktop      Documents  fetchData  Music      Public  Templates
docker-python-app  Downloads  kubernetes  Pictures    snap    Videos
vboxuser@Ubuntu:~$ cd kubernetes
vboxuser@Ubuntu:~/kubernetes$ ls
backend  frontend  README.md
commands-to-execute  k8s
commands-to-stop-instances  pre-requisites-command
vboxuser@Ubuntu:~/kubernetes$ kubectl delete all --all --force --grace-period=0
Warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
vboxuser@Ubuntu:~/kubernetes$ kubectl delete namespace kube-system --force --grace-period=0
Warning: Immediate deletion does not wait for confirmation that the running resource has been terminated. The resource may continue to run on the cluster indefinitely.
Unable to connect to the server: dial tcp 192.168.49.2:8443: connect: no route to host
vboxuser@Ubuntu:~/kubernetes$ minikube stop
Stopping node "minikube" ...
1 node stopped.
vboxuser@Ubuntu:~/kubernetes$ minikube delete --all --purge
Deleting "minikube" in docker ...

```

```

boxuser@Ubuntu:~/kubernetes$ docker kill $(docker ps -q)
docker kill requires at least 1 argument.
See 'docker kill --help'.

sage: docker kill [OPTIONS] CONTAINER [CONTAINER...]

Kill one or more running containers
boxuser@Ubuntu:~/kubernetes$ docker rm -f $(docker ps -aq)
docker rm requires at least 1 argument.
See 'docker rm --help'.

sage: docker rm [OPTIONS] CONTAINER [CONTAINER...]

Remove one or more containers
boxuser@Ubuntu:~/kubernetes$ systemctl stop docker
Warning: Stopping docker.service, but it can still be activated by:
  docker.socket
boxuser@Ubuntu:~/kubernetes$ systemctl start docker
boxuser@Ubuntu:~/kubernetes$ kill -f docker
kill: killing pid 4455 failed: Operation not permitted
boxuser@Ubuntu:~/kubernetes$ kill -f minikube
boxuser@Ubuntu:~/kubernetes$ kill -f kubectl
boxuser@Ubuntu:~/kubernetes$ kill -f containerd
kill: killing pid 703 failed: Operation not permitted
kill: killing pid 4455 failed: Operation not permitted
boxuser@Ubuntu:~/kubernetes$ docker system prune -a --volumes -f
Deleted Images:
Untagged: gcr.io/k8s-minikube/kicbase:v0.0.46
Untagged: gcr.io/k8s-minikube/kicbase@sha256:fd2d445ddcc33ebc5c6b68a17e6219ea207ce63c005095ea1525296da2d1a279
Deleted: sha256:e72c4cbe9b296d8a58fbcae1a7b969fa1cee662cd7b86f2d4efc5e146519cf0a
Deleted: sha256:5afe0607e8a82226399f6352a39419bb8765f01cfbaed3e95c7426770a727d4
Untagged: backend:latest
Deleted: sha256:1e23462d2b295ddb3ce1ba18b74067ff87caa316e537a1f5f54ae53b351fec4d
Deleted: sha256:adfbf85ee54aa87e13de0fb0c49ba486ddf352afbfbc77de9f20bb0456ad17
Deleted: sha256:0a3b5fb1d925f7036b87e90262624c29994a504400e60a2fba6d443480a73958
Deleted: sha256:c6ca88995a7d7402d9b23f07617377dcfb32283a64d692ae6ad346fd5f0bb76d
Deleted: sha256:018845dc1a0dc7924e3b9453574b0d00654cd2ac3a63534102eed46b31dfda46
Deleted: sha256:17a6e85d70d14ce7080b9644d8cc21490738527e554001ecc52a5c50fcb65da0

Client Version: v1.32.3
Customize Version: v5.5.0
boxuser@Ubuntu:~/kubernetes$ minikube start
🌟 minikube v1.35.0 on Ubuntu 22.04 (vbox/amd64)
🔧 Automatically selected the docker driver. Other choices: none, ssh

⚠️ The requested memory allocation of 1967MiB does not leave room for system overhead (total system memory: 1967MiB). You may face stability issues.
💡 Suggestion: Start minikube with less memory allocated: 'minikube start --memory=1967mb'

🔧 Using Docker driver with root privileges
🏠 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% 3.29 Mi
> gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 2.99 Mi

C
boxuser@Ubuntu:~/kubernetes$ kubectrl get nodes
0321 14:26:31.716466 5526 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%3Fti
t%3D32s' /><script id='redirect' data-redirect-url='/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/4922217f/scripts/redirect.js
script></head><body style='background-color:white; color:white;'>
  Authentication required
  <!--
  -->

  </body></html>
>
0321 14:26:31.847349 5526 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%3Fti
t%3D32s' /><script id='redirect' data-redirect-url='/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/4922217f/scripts/redirect.js
script></head><body style='background-color:white; color:white;'>
  Authentication required
  <!--
  -->

  </body></html>

```



```
vboxuser@ubuntu: ~/k8s
deleted: sha256:27e66bba1c7251b5d8b5d80725b7e5c9ae37897c75cb8c9097df42fdb6976ac6
deleted: sha256:24d3228e29e3d279f23859f348347039001d5131c018fd334355cf97b0ebd99f
deleted: sha256:53babe930602419c46727120034c3748ae023bef28f10edb018fbcfbcedbc0e8

total reclaimed space: 2.497GB
vboxuser@Ubuntu:~/kubernetes$ sudo netstat -tulnp | grep ":8080"
sudo] password for vboxuser:
tcp6      0      0  :::*          :::*          LISTEN    684/java
vboxuser@Ubuntu:~/kubernetes$ sudo kill -9 684
vboxuser@Ubuntu:~/kubernetes$ sudo apt update
gn:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
it:2 https://pkg.jenkins.io/debian-stable binary/ Release
it:4 http://in.archive.ubuntu.com/ubuntu jammy InRelease
et:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
et:6 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
it:7 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
etched 257 kB in 3s (80.5 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 packages can be upgraded. Run 'apt list --upgradable' to see them.
vboxuser@Ubuntu:~/kubernetes$ 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 (26.1.3-0ubuntu1~22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 60 not upgraded.
vboxuser@Ubuntu:~/kubernetes$ sudo systemctl start docker
vboxuser@Ubuntu:~/kubernetes$ sudo systemctl enable docker
vboxuser@Ubuntu:~/kubernetes$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~22.04.1
vboxuser@Ubuntu:~/kubernetes$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 54.6M  100 54.6M    0     0  2007k      0  0:00:27  0:00:27 --:--:-- 3357k
vboxuser@Ubuntu:~/kubernetes$ chmod +x kubectl

root@Ubuntu: ~
vboxuser@Ubuntu: ~/FetchData/k8s
deleted: sha256:266ac9954e89edd0de0f000d12a2cdad5b1594fc63af069128cf89aca4d2d5d9
deleted: sha256:070fe2d6cef576e4679c76f8259462b62c0a62a4008a7bddc0a3eeb1e6411b5
deleted: sha256:501f0a6916e7f44f4df19851efb8f177a5e45f8a08e091388dcadd9eae5e266c
deleted: sha256:dfd3e0a33d22121d4cf6a2dea796eceeabaeedf923b43c87c1f391b09d37691f
deleted: sha256:1e157aded8660c64a15efbcb118c4ac516c2d553c7de170208180b5ecf32e2af
deleted: sha256:9e340bb42a7f926345df897e9015d75e0b29c1a4635087d567a779ef37542947
deleted: sha256:7f65035481c1db44c4abb6d5a7f69c845f86afce070cbd29c358d8fd3d6d11e2b
deleted: sha256:4c6bb0b35d82c77cc1d38bafac255f304bc7547283daca4bfd2f34a34fd9c1bb
deleted: sha256:3d1562857b714a742ebe5771b0c35197bdcf11bd3bba52f45c1e559fdded2c63f
deleted: sha256:0dede4d981c104cd3c217c6e4f3c8b77f6e6f4f8e1ea98281514287d3767c3d8
deleted: sha256:53d940824e66bf479a7a5b21b13eded40e8f107f77b02acee94aa7a896bca905
deleted: sha256:629d2b6c177a92a6bd15f08e0b88bc5845ffa52a8b4639660b115c5448792c7f
deleted: sha256:72ce103e86552c9c2416734243c96a0af785bb39d2c8c59ed81c97b770cf3a3c
deleted: sha256:18cf02265fd94d4d17a0a465d6053f18dd19fcd89a93041eeb67c832e07d915a

Deleted build cache objects:
jgn4v66qdc1oiudg5tmc9ael1
c4409t398d0utulfuhkyzq4hx
tshw1so3ddwo1w2d49b0cfdw5
w5thjxlkf0ch8h628vorpnvyo
rc328pwwoswp6xlr97kqg6wuiP
lumkcuebuk3lz9zmh21f5ge8
upgzr3jj56atltldrqvcw67rv
mnxhrg6u5nlocgj121q29lbp30
p36kx45cign259092vvq7bs4d
ds6gqz4li3rggd090kyj353y4
rts3m0tu9atw00x22n0vuykid
dji8gqfvzbvxyx1y7fngwgkmd
3lr1hxt44r9adxa4cg9e01hw9
kx3lvz6a5fgnleaq1fvq71af

Total reclaimed space: 449.8MB
root@Ubuntu:~# sudo netstat -tulnp | grep ":8080"
tcp6      0      0  :::*          :::*          LISTEN    575/java
root@Ubuntu:~# sudo kill -9 <PID>
-bash: syntax error near unexpected token `newline'
root@Ubuntu:~#
```

```
vboxuser@Ubuntu: ~/fetchData/backend × vboxuser
GNU nano 6.2 docker-compose.yml
version: '3.8' # Use a specific Docker Compose version

services:
  web:
    build: .
    ports:
      - "7000:7000"
    volumes:
      - ./app # Mount current directory to /app
    working_dir: /app # Ensure the app runs from the /app directory
    restart: always
```

```
vboxuser@Ubuntu: ~/fetchData/backend × vboxuser
GNU nano 6.2 Dockerfile
# Use an official Python runtime as a parent image
FROM python:3.11
# Set the working directory in the container
WORKDIR /app
# Copy the requirements file and install dependencies
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
# Copy the application source code
COPY . .
# Expose the port the app runs on
EXPOSE 7000
# Define the command to run the application
CMD ["python", "app.py"]
```

Store Data in String × Online Python Compiler ( × (no subject) - sivak.24mce × 172.17.0.2:7000/product × +

← → ↺ 172.17.0.2:7000/product ☆

```
[{"id":101,"productName":"rice","price":200,"quantity":"1kg"}, {"id":102,"productName":"vegetables","price":100,"quantity":"2kg"}, {"id":103,"productName":"coffee","price":200,"quantity":"100g"}]
```

```
vboxuser@Ubuntu: ~/fetchData
vboxuser@Ubuntu: ~/fetchData
get:16 http://in.archive.ubuntu.com/ubuntu jammy-backports/main amd64 DEP-11 Met
data [7,040 B]
get:17 http://in.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 DEP-
1 Metadata [212 B]
get:18 http://in.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 DEP-11
Metadata [17.7 kB]
get:19 http://in.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 DEP-
1 Metadata [212 B]
etched 1,042 kB in 6s (163 kB/s)
reading package lists... Done
building dependency tree... Done
reading state information... Done
0 packages can be upgraded. Run 'apt list --upgradable' to see them.
vboxuser@Ubuntu:~/fetchData$ 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 (26.1.3-0ubuntu1~22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 60 not upgraded.
vboxuser@Ubuntu:~/fetchData$ curl -LO https://storage.googleapis.com/minikube/re
leases/latest/minikube-linux-amd64
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 119M  100 102M   0     0  1276k      0  0:01:35  0:01:22  0:00:13 1066k
vboxuser@Ubuntu:~/fetchData
vboxuser@Ubuntu:~/fetchData
vboxuser@Ubuntu:~/fetchData/k8s$ nano k8s-deploy.yml
vboxuser@Ubuntu:~/fetchData/k8s$ nano frontend-deployment.yml
vboxuser@Ubuntu:~/fetchData/k8s$ nano service.yml
vboxuser@Ubuntu:~/fetchData/k8s$ nano configmap.yml
vboxuser@Ubuntu:~/fetchData/k8s$ nano configmap.yml
vboxuser@Ubuntu:~/fetchData/k8s$ cd ..
vboxuser@Ubuntu:~/fetchData$ sudo apt update
[sudo] password for vboxuser:
gn:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
gn:2 https://pkg.jenkins.io/debian-stable binary/ Release
gn:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
get:5 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
get:6 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadat
a [43.1 kB]
get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 M
etadata [208 B]
get:10 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Me
tadata [126 kB]
get:11 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 DEP-11
metadata [208 B]
get:12 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 DEP-11 Metad
ata [103 kB]
```

```
root@Ubuntu: ~  
vboxuser@Ubuntu: ~/fetchData/k8s  
vboxuser@Ubuntu:~/fetchData/k8s$ cd  
vboxuser@Ubuntu:~$ su -  
Password:  
su: Authentication failure  
vboxuser@Ubuntu:~$ su -  
Password:  
root@Ubuntu:~# 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 | 1.10 MiB/s, done.  
Resolving deltas: 100% (2/2), done.  
root@Ubuntu:~# ls  
kubernetes  snap  vboxpostinstall.sh  
root@Ubuntu:~# cd kubernetes  
root@Ubuntu:~/kubernetes# ls  
backend  commands-to-stop-instances  frontend  k8s  README.md  
root@Ubuntu:~/kubernetes# ls  
backend  commands-to-stop-instances  frontend  k8s  README.md  
root@Ubuntu:~/kubernetes# kubectl delete all --all --force --grace-period=0  
Command 'kubectl' not found, but can be installed with:  
snap install kubectl  
root@Ubuntu:~/kubernetes# kubectl delete namespace kube-system --force --grace-period=0  
Command 'kubectl' not found, but can be installed with:  
snap install kubectl  
root@Ubuntu:~/kubernetes# minikube stop  
🚧 Profile "minikube" not found. Run "minikube profile list" to view all profiles.  
🚧 To start a cluster, run: "minikube start"  
root@Ubuntu:~/kubernetes# minikube delete --all --purge  
🔥 Successfully deleted all profiles  
🔥 Successfully purged minikube directory located at - [/root/.minikube]  
root@Ubuntu:~/kubernetes# docker kill $(docker ps -q)  
"docker kill" requires at least 1 argument.  
See 'docker kill --help'.  
Usage: docker kill [OPTIONS] CONTAINER [CONTAINER...]
```

```
vboxuser@Ubuntu: ~/fetchData/backend  
GNU nano 6.2 product.csv *  
id,productName,price,quantity  
101,rice,200,1kg  
102,vegetables,100,2kg  
103,coffee,200,100g
```

```
vboxuser@Ubuntu: ~/fetchData/backend × vboxuser
GNU nano 6.2 requirements.txt
flask
pandas

vboxuser@Ubuntu:~/fetchData/backend$ nano requirlements.txt
vboxuser@Ubuntu:~/fetchData/backend$ nano requirements.txt
vboxuser@Ubuntu:~/fetchData/backend$ sudo 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 6.656kB
Step 1/7 : FROM python:3.11
--> 18c0f2265fd9
Step 2/7 : WORKDIR /app
--> Using cache
--> 629d2b6c177a
Step 3/7 : COPY requirements.txt .
--> Using cache
--> 070fe2d6cef5
Step 4/7 : RUN pip install --no-cache-dir -r requirements.txt
--> Using cache
--> 2be742dfb9ea
Step 5/7 : COPY . .
--> Using cache
--> aa2d82b29b97
Step 6/7 : EXPOSE 7000
--> Using cache
--> 797595a0c852
Step 7/7 : CMD ["python", "app.py"]
see "docker run --help"
vboxuser@Ubuntu:~/fetchData/backend$ docker run backend
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:7000
* Running on http://172.17.0.2:7000
Press CTRL+C to quit
```



```
vboxuser@Ubuntu: ~/fetchData/backend x vboxx

GNU nano 6.2 app.py
from flask import Flask
import pandas as pd

app = Flask(__name__)

# Correct spelling and syntax
@app.route("/product", methods=['GET'])
def read_data():
    try:
        # Read CSV and convert to JSON
        df = pd.read_csv("product.csv")
        json_data = df.to_json(orient='records')
        return json_data, 200 # Return JSON with status 200
    except Exception as e:
        return {"error": str(e)}, 500 # Return error message with status 500

# Correct Python syntax for Flask execution
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=7000)
```

```
vboxuser@Ubuntu: ~/fetchData/backend x

vboxuser@Ubuntu:~/fetchData/backend$ ls
app.py  docker-compose.yml  Dockerfile  product.csv  requirements.txt

vboxuser@Ubuntu:~/fetchData/backend$ sudo netstat -lp
[sudo] password for vboxuser:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 localhost:domain        0.0.0.0:*               LISTEN      330/systemd-resolve
tcp        0      0 localhost:ipp            0.0.0.0:*               LISTEN      532/cupsd
tcp        0      0 0.0.0.0:http             0.0.0.0:*               LISTEN      628/nginx: master p
tcp        0      0 localhost:43275          0.0.0.0:*               LISTEN      569/containerd
tcp6       0      0 ip6-localhost:ipp       [::]:*                 LISTEN      532/cupsd
tcp6       0      0 [::]:http-alt           [::]:*                 LISTEN      534/java
tcp6       0      0 [::]:http                [::]:*                 LISTEN      628/nginx: master p
udp        0      0 0.0.0.0:44808            0.0.0.0:*               2429/firefox
udp        0      0 0.0.0.0:domain          0.0.0.0:*               330/systemd-resolve
udp        0      0 0.0.0.0:54491            0.0.0.0:*               428/avahi-daemon: r
udp        0      0 0.0.0.0:mdns             0.0.0.0:*               428/avahi-daemon: r
udp6       0      0 [::]:53912              [::]:*                 428/avahi-daemon: r
udp6       0      0 [::]:mdns                [::]:*                 428/avahi-daemon: r
raw6       0      0 [::]:ip6-icmp            [::]:*                 7          432/NetworkManager

Active UNIX domain sockets (only servers)
Proto RefCnt Flags   Type       State       I-Node  PID/Program name  Path
unix    2      [ ACC ] STREAM   LISTENING   8123    569/containerd    /run/containerd/containerd.sock.ttrpc
unix    2      [ ACC ] STREAM   LISTENING   8125    569/containerd    /run/containerd/containerd.sock
unix    2      [ ACC ] STREAM   LISTENING   10539   1723/systemd      /run/user/1000/systemd/private
unix    2      [ ACC ] STREAM   LISTENING   10557   1723/systemd      /run/user/1000/bus
unix    2      [ ACC ] STREAM   LISTENING   10563   1723/systemd      /run/user/1000/gnupg/S.dirmngr
unix    2      [ ACC ] STREAM   LISTENING   10565   1723/systemd      /run/user/1000/gnupg/S.gpg-agent.browser
unix    2      [ ACC ] STREAM   LISTENING   10567   1723/systemd      /run/user/1000/gnupg/S.gpg-agent.extra
unix    2      [ ACC ] STREAM   LISTENING   10569   1723/systemd      /run/user/1000/gnupg/S.gpg-agent.ssh
unix    2      [ ACC ] STREAM   LISTENING   10571   1723/systemd      /run/user/1000/gnupg/S.gpg-agent
```

```
vboxuser@Ubuntu:~/fetchData/backend$ ls
app.py  product.csv
vboxuser@Ubuntu:~/fetchData/backend$ nano app.py
vboxuser@Ubuntu:~/fetchData/backend$ python3 app.py
  id productName  price quantity
0  101      rice    200        1kg
1  102 vegetables    100        2kg
2  103    coffee    200       100g
vboxuser@Ubuntu:~/fetchData/backend$ nano app.py
```