Microsoft Windows [Version 10.0.19045.5371]

(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>docker -v

Docker version 27.4.0, build bde2b89

C:\Windows\system32>docker --version

Docker version 27.4.0, build bde2b89

C:\Windows\system32>docker info

Client:

Version: 27.4.0

Context: desktop-linux

Debug Mode: false

#2 to pull iamges from docker hub

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

#all these images are imported from docker-hub

C:\Windows\system32>docker pull hello-world

Using default tag: latest

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

hello-world latest 74cc54e27dc4 12 days ago 10.1kB

# it will pull the images from docker-hub – by –default using latest tag

C:\Windows\system32>docker search mysql

NAME DESCRIPTION STARS OFFICIAL

mysql MySQL is a widely used, open-source relation… 15617 [OK]

# give information about mysql image

C:\Windows\system32>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

# show only running contianers

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS

# show all contianers including stopped ones

#docker run to create container

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker run hello-world - create container using iamge hello-world

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker inspect f77

**#config or network setting part very iimportatn**

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

f77b14ae6cba hello-world "/hello" 4 minutes ago Exited (0) 4 minutes ago

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

C:\Windows\system32>docker run hello-world:latest - careate new continer using latest image tag

Hello from Docker! - docker run to crate container using image hello-world image

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker run --name hello-c1 hello-world

#create new container using hello-world image and name of container hello-c1 – we use docker run to create new conatianer

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

C:\Windows\system32>docker run --name python-c1 python:3.9-slim

**#create new container using python:3.9-slim image**

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker run --name python-c2 -d python:3.9-slim

0aada851820ecfb28eaf8d90596484a310b56aa1176a4a6e81cde483dcf779d3

#run coantienr in bg and print coantienr id

C:\Windows\system32>docker run --name python-c3 -d python:3.9-slim

1e26611c10f6ef56f742da197fdfa3abcd5ee548e48e4f9a8daec9ddc4f8b7a0

C:\Windows\system32>docker run --name python-c3 -d python:3.9-slim

1e26611c10f6ef56f742da197fdfa3abcd5ee548e48e4f9a8daec9ddc4f8b7a0

C:\Windows\system32>docker run --name python-c4 python:3.9-slim

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0aada851820e python:3.9-slim "python3" 50 seconds ago Exited (0) 48 seconds ago python-c2

C:\Windows\system32>docker run --name python-c3 -it -d python:3.9-slim

90d9c501b64cf39662a2e8908efb5328dd9a51c4bb15a57c578759c2a0473064

# create and run a container in interactive and detach mode , up rahega kyonki -it ka use kara

C:\Windows\system32>docker exec -it python-c3 python

Python 3.9.21 (main, Jan 14 2025, 02:22:56)

**#python ka conatienr banao aur usse use karo dabake ,provide a python termainal for running container**

C:\Windows\system32>docker inspect python-c3

**# to find cmd to run python in shell or terminal , -d background main run karta hai containe**r ko

#

C:\Windows\system32>docker run --name python-c4 -it python:3.9-slim

Python 3.9.21 (main, Jan 14 2025, 02:22:56)

**-d use backg main run karega aur exec se terminal milega apakoo , -it in interactive mode main run karega container ko**

C:\Windows\system32>docker run --name python-c5 -it --rm python:3.9-slim

**# run for specific interval till you use it and then automatically stop and remove**

C:\Windows\system32>docker stop python-c3

python-c3

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

2fdb0ee434ca python:3.9-slim "python3" 11 minutes ago Exited (0) 6 minutes ago python-c4

C:\Windows\system32>docker start python-c3

python-c3

C:\Windows\system32>docker restart python-c3

python-c3

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

**#conainer name you can use coantienr id**

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker rm python-c4

python-c4 - **# to remove contiaenr python-c4**

C:\Windows\system32>docker rmi hello-world

Error response from daemon: conflict: unable to remove repository reference "hello-world" (must force) - container f66df950e342 is using its referenced image 74cc54e27dc4

C:\Windows\system32>docker rm f7

f7

C:\Windows\system32>docker rm 4c

4c

C:\Windows\system32>docker rm f6 # remove coanteinr by id

f6

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker rmi hello-world

C:\Windows\system32>docker container prune

WARNING! This will remove all stopped containers.

Are you sure you want to continue? [y/N] y

C:\Windows\system32>docker system prune

WARNING! This will remove:

- all stopped containers

- all networks not used by at least one container

- all dangling images

- unused build cache

C:\Windows\system32>docker logs python-c3

Python 3.9.21 (main, Jan 14 2025, 02:22:56)

C:\Windows\system32>docker pull mysql

Using default tag: latest

**Docker run to run conainer to create container , -d bg main run karne ka liye**

**Docker exec to run in shell**

C:\Windows\system32>docker pull mysql

Using default tag: latest

latest: Pulling from library/mysql

2c0a233485c3: Pull complete

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

docker dind 49fc2acb217f 12 days ago 378MB

mysql latest a52cba19e8cc 12 days ago 797MB

C:\Windows\system32>docker run --name mysql-c1 -e MYSQL\_ROOT\_PASSWORD=deep -d mysql

cbdb82ca2f3e432421ff24fffecd01c67fb58e46bb3f1514b13db08e87770798

**# e for env variable , d for run in background , mysql-c1 coantiner name and mysql is image return coantiner id**

C:\Windows\system32>docker run --name mysql-c1 -e MYSQL\_ROOT\_PASSWORD=deep -d mysql

cbdb82ca2f3e432421ff24fffecd01c67fb58e46bb3f1514b13db08e87770798

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

cbdb82ca2f3e mysql "docker-entrypoint.s…" 56 seconds ago Up 55 seconds 3306/tcp, 33060/tcp mysql-c1

**# by defulat coantiner up rahega , image pa bhi depend karta hai**

C:\Windows\system32>docker exec -it mysql-c1 mysql -u root -p

Enter password: # to get mysql terminal use exec , mysql likha jata hai iamge nahi hai yha pa , p password, u for root

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 9

Server version: 9.2.0 MySQL Community Server - GPL

C:\Windows\system32>docker run --name mysql-c1 -e MYSQL\_ROOT\_PASSWORD=deep -d mysql

cbdb82ca2f3e432421ff24fffecd01c67fb58e46bb3f1514b13db08e87770798

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker exec -it mysql-c1 mysql -u root -p

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

mysql> show db

-> show databases

-> clear

-> exit

-> ^C

mysql> show databases

-> ^C

mysql> show databases;

+--------------------+

| Database |

+--------------------+

**# sara mysql ka kaam kar skate hai isme ham shell ka andar**

**# hamne docker pa install kara mysql aur sara kam pa rha hai , without OS , you can use differ versions of iamge to use it**

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker stop cb

cb

C:\Windows\system32>docker start cb

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

cbdb82ca2f3e mysql "docker-entrypoint.s…" 8 minutes ago Up 8 minutes 3306/tcp, 33060/tcp mysql-c1

C:\Windows\system32>docker stop cb

cb

C:\Windows\system32>docker rm cb

cb

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>

**# Port expose karna apne server kaa**

**Pahle port host machine ka likhete hai 8080 host machine ka port hai**

**, aur baad main nginx ka porth ai 8080:80 hai**

**# agar host machine se 8080 port hit karu toh nginx ka 80 port pa hit kar rha hu , host port or docker ka port**

C:\Windows\system32>docker run --name nginx-c1 -d -p 8080:80 nginx

e82c7183397044c1d0b8373c22c21b1a998de3b96ef6b262e30eba9d842b4ef6

# hsot machine ka 8080 port ko hit karunga toh ham 80 port ko hit kar rha hai

# create nginx container and expose it port

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

e82c71833970 nginx "/docker-entrypoint.…" 46 seconds ago Up 45 seconds 0.0.0.0:8080->80/tcp nginx-c1

**# 8080 se local machine pa hit karu toh ham 80 port hit kar rha hai docker ka**

# 80 port pa default nginx ka port serve hota hai , <http://localhost:8080/> - ispe hit kar rha hut oh 80 port hit ho rha hai nginx kaa , nginx 80 port p achal rha hai

$ if we hit on port 8080 then we simply redirect to port 80

**#nginx ka port expose kara diya coanienr se , aur ab apache ka port expose kar dunga toh dono server chala dunga coantiner ki help se**

C:\Windows\system32>docker pull nginx

Using default tag: latest

latest: Pulling from library/nginx

Digest: sha256:0a399eb16751829e1af26fea27b20c3ec28d7ab1fb72182879dcae1cca21206a

Status: Image is up to date for nginx:latest

docker.io/library/nginx:latest

What's next:

View a summary of image vulnerabilities and recommendations → docker scout quickview nginx

C:\Windows\system32>docker run --name nginx-c1 -d -p 8080:80 nginx

e82c7183397044c1d0b8373c22c21b1a998de3b96ef6b262e30eba9d842b4ef6

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

e82c71833970 nginx "/docker-entrypoint.…" 46 seconds ago Up 45 seconds 0.0.0.0:8080->80/tcp nginx-c1

C:\Windows\system32>docker pull httpd:alpine

alpine: Pulling from library/httpd

1f3e46996e29: Already exists # pulling apache iamge

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

**# nginx or apache dono cahla skate hai docker continer paa**

**#8000 port host ka expose kar rha apachae ka liye , or continaer ka port 80 hoga apache ka , ye ispe bind hoga , host ka port bind hoga usse**

**C:\Windows\system32>docker run --name apache-c1 -d -p 8000:80 httpd:alpine**

**225f06ecaca146585702238d157d8bf46ebc9318c3aa13da7b20039bde920759**

# ek port pa apache chalega or ek port pa nginx chalega , dono web servier chala sakete hai ,docker coantiner see

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

225f06ecaca1 httpd:alpine "httpd-foreground" 43 seconds ago Up 43 seconds 0.0.0.0:8000->80/tcp apache-c1

e82c71833970 nginx "/docker-entrypoint.…" 14 minutes ago Up 14 minutes 0.0.0.0:8080->80/tcp nginx-c1

#db se kaam kar rha toh volume bhi show hgoa

$ if I hit on 8000 host machine then it will automatically redirect to port 80 of apache

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

225f06ecaca1 httpd:alpine "httpd-foreground" 43 seconds ago Up 43 seconds 0.0.0.0:8000->80/tcp apache-c1

e82c71833970 nginx "/docker-entrypoint.…" 14 minutes ago Up 14 minutes 0.0.0.0:8080->80/tcp nginx-c1

C:\Windows\system32>docker stop 22

22

C:\Windows\system32>docker stop e8

e8

**# khud iamge bana sakte ho aur share kar skate ho ddocker hub pa , log usse use kar payenge**

#Dockerfiel bana instruction to make docker iamge sabse pahle

# BASE IMAGE choose karo project ka liye , jispe project depend ho

Dockerfile

#use ubuntu base image

FROM ubuntu

#setup working dirctory inside coatiner

WORKDIR /app

# copy script file to the container in app directory

COPY script.sh .

# make the script executable

RUN chmod+x script.xh

# run the script using this cmd when container starts

CMD ["./script.sh"]

echo "Hello from deepanshu image"

script.sh file hai hamari

**now build docker images and give iamge name**

PS C:\Users\my pc\Desktop\docker geekyshows\myimage> docker build -t mydockerimage1 .

# start building docker image in current directory

# iamge ka container bana skate hum , jo bhi use karega iske

. currrenet dicrecoty main build kar rha hai

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

mydockerimage1 latest 32969f459253 About a minute ago 78.1MB

Dockerfile

# Use the official Ubuntu base image

FROM ubuntu:latest

# Set up the working directory inside the container

WORKDIR /app

# Copy script file to the container in the /app directory

COPY script.sh .

# Grant execution permission to the script

RUN chmod +x script.sh

# Define the default command to run the script when the container starts

CMD ["./script.sh"]

Scrip.sh

#!/bin/bash

echo "Hello from Deepanshu's image"

PS C:\Users\my pc\Desktop\docker geekyshows\mydocker> docker build -t myimage1 .

[+] Building 6.6s (10/10) FINISHED # building our own image

C:\Windows\system32>docker run myimage1

Hello from Deepanshu's image # create coatner with our image

#DONE WITH THIS TILL NOW

C:\Windows\system32>docker run myimage1

Hello from Deepanshu's image

C:\Windows\system32>clear

'clear' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>docker run --name my-c1 myimage1

Hello from Deepanshu's image - **crating coaniner form docker iamges**

Hello from Deepanshu's image

# create container using myimage , and it exist as it not preforming any kind of work

C:\Windows\system32>docker container prune

WARNING! This will remove all stopped containers.

Are you sure you want to continue? [y/N] y

Deleted Containers:

C:\Windows\system32>docker run --name c1 myimage1

Hello from Deepanshu's image

# jaise hi script main cahgne karat oh usse d0bara build karo

#!/bin/bash

echo "Hello from Deepanshu's image"

echo "geekss"

PS C:\Users\my pc\Desktop\docker geekyshows\mydocker> docker build -t myimage .

[+] Building 4.6s (10/10) FINISHED

C:\Windows\system32>docker run --name c2 myimage

Hello from Deepanshu's image

geekss

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

86655baf62d3 myimage "./script.sh" 47 seconds ago Exited (0) 45 seconds ago c2

2de5229a4246 myimage1 "./script.sh" 2 minutes ago Exited (0) 2 minutes ago c1

$ If we updated anything then we need to again build the docker file . build again docker image if you updated docker file

PS C:\Users\my pc\Destop\docker geekyshows\mydocker> docker build --no-cache -t myimage .

[+] Building 6.1s (10/10) FINISHED - image build without cache , thoda slow hoga , but without cache build hoga

#!/bin/bash

echo "Hello from Deepanshu's image"

echo "geekss"

$ BUILD A IAMGE FORM DOCKER FILE

PS C:\Users\my pc\Desktop\docker geekyshows\mydocker> docker build -t mydockerimage:v1.0 .

[+] Building 6.2s (9/9) FINISHED

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

mydockerimage v1.0 1ea3d39239a1 3 minutes ago 78.1MB

#!/bin/bash

echo "Hello from Deepanshu's image"

echo "geekss"

echo "new verson2"

PS C:\Users\my pc\Desktop\docker geekyshows\mydocker> docker build -t mydockerimage:v2.0 .

[+] Building 5.8s (10/10) FINISHED - new version niakl diya image kaa hamra

#!/bin/bash

echo "Hello from Deepanshu's image"

echo "geekss"

echo "new verson2"

echo "latest vesrion"

PS C:\Users\my pc\Desktop\docker geekyshows\mydocker> docker build -t mydockerimage .

[+] Building 3.4s (9/9) FINISHED

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

**# dcoker image task specific hai hamara , Ubuntu , redis project requ pa depend karta hai**

**.dockerignore – ignore file while build the imgae**

.dockerignore - .env , node-modules

**# To push image inside docker hub**

C:\Windows\system32>docker tag 3652fda97c30 deepanshu12353/mydockerimage

**# mydockerimage ko tag kara usename ka sath , image id di jo push karni hai**

C:\Windows\system32>docker push deepanshu12353/mydockerimage

Using default tag: latest

The push refers to repository [docker.io/deepanshu12353/mydockerimage]

3cf96f23cf39: Pushed

0a56a60db3a9: Layer already exists

687d50f2f6a6: Layer already exists

latest: digest: sha256:dd4168e9e9cf4689db867371cebece21f869badbcb59ae59279d614ac71902d8 size: 1149

**# PuSHING Images**

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

mydockerimage latest 3652fda97c30 8 seconds ago 78.1MB

mydockerimage v2.0 cdf6332c25e6 2 minutes ago 78.1MB

mydockerimage v1.0 1ea3d39239a1 8 minutes ago 78.1MB

httpd alpine 9f1f788190b8 10 days ago 63.8MB

docker dind 49fc2acb217f 12 days ago 378MB

mysql latest a52cba19e8cc 12 days ago 797MB

kicbase/stable v0.0.46 e72c4cbe9b29 3 weeks ago 1.31GB

python 3.9-slim 453d3342b002 2 months ago 126MB

nginx latest 9bea9f2796e2 2 months ago 192MB

docker/desktop-kubernetes kubernetes-v1.30.5-cni-v1.4.0-critools-v1.29.0-cri-dockerd-v0.3.11-1-debian a60dd8dae2ad 3 months ago 435MB

registry.k8s.io/kube-apiserver v1.30.5 e9adc5c075a8 4 months ago 117MB

registry.k8s.io/kube-controller-manager v1.30.5 38406042cf08 4 months ago 111MB

registry.k8s.io/kube-scheduler v1.30.5 25903461e65c 4 months ago 62MB

registry.k8s.io/kube-proxy v1.30.5 71161e05b9bb 4 months ago 84.7MB

registry.k8s.io/coredns/coredns v1.11.3 c69fa2e9cbf5 6 months ago 61.8MB

registry.k8s.io/etcd 3.5.15-0 2e96e5913fc0 6 months ago 148MB

registry.k8s.io/etcd 3.5.12-0 3861cfcd7c04 12 months ago 149MB

docker/desktop-vpnkit-controller dc331cb22850be0cdd97c84a9cfecaf44a1afb6e 556098075b3d 21 months ago 36.2MB

registry.k8s.io/pause 3.9 e6f181688397 2 years ago 744kB

dockercoins/rng v0.1 9d07849dba30 3 years ago 56.5MB

docker/desktop-storage-provisioner v2.0 99f89471f470 3 years ago 41.9MB

C:\Windows\system32>docker tag 3652fda97c30 deepanshu12353

C:\Windows\system32>

C:\Windows\system32>docker tag 3652fda97c30 deepanshu12353/mydockerimage

C:\Windows\system32>docker push

"docker push" requires exactly 1 argument.

See 'docker push --help'.

Usage: docker push [OPTIONS] NAME[:TAG]

**Upload an image to a registry**

C:\Windows\system32>docker login

Authenticating with existing credentials...

Login Succeeded

C:\Windows\system32>docker push deepanshu12353/mydockerimage

Using default tag: latest

The push refers to repository [docker.io/deepanshu12353/mydockerimage]

3cf96f23cf39: Pushed

0a56a60db3a9: Layer already exists

687d50f2f6a6: Layer already exists

latest: digest: sha256:dd4168e9e9cf4689db867371cebece21f869badbcb59ae59279d614ac71902d8 size: 1149

C:\Windows\system32>docker push deepanshu12353/mydockerimage

Using default tag: latest

The push refers to repository [docker.io/deepanshu12353/mydockerimage]

$ DONE YHA SE KARO

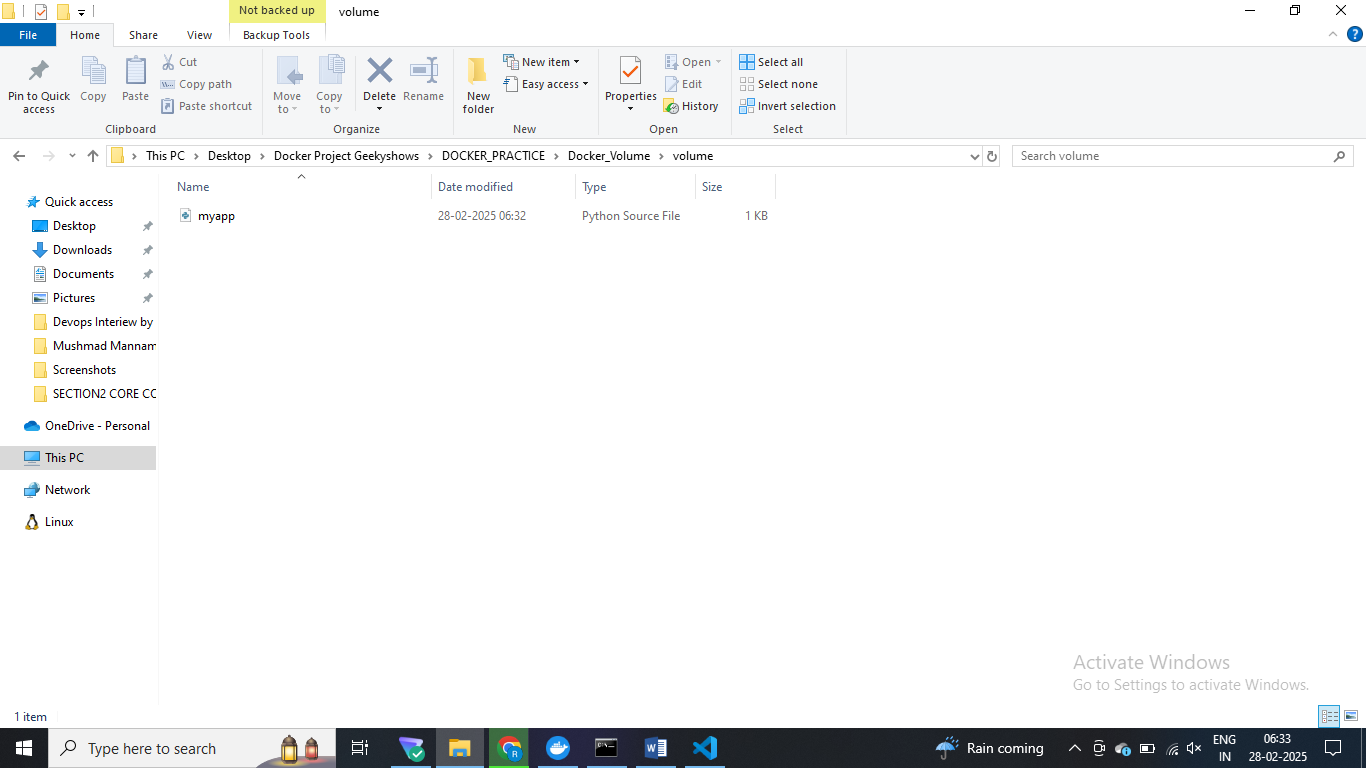
**Docker Volume**

**Ek folder ko us contaier ka path ka sath mount kar do , persisit data ka liye volume ka use karo**

**Share data betwenn host to coanitner**

docker run --name python-c1 -itd -v "C:\Users\my pc\Desktop\docker geekyshows\DockerVolume\volume:/myapp" python

ye hamara volume wala path myapp ka andar mount hoga , write anything volume show in myapp and vice versa , careate a container



C:\Windows\system32>docker run --name python-c1 -itd -v C:\Users\my pc\Desktop\docker geekyshows\DockerVolume\volume:/myapp

docker: invalid reference format: repository name (library/pc\Desktop\docker) must be lowercase.

See 'docker run --help'.

C:\Windows\system32>docker run --name python-c1 -itd -v "C:\Users\my pc\Desktop\docker geekyshows\DockerVolume\volume:/myapp" python

72220f92694b329d7cd7227632c827721cbb639786e9ba13cd966e44adf1d07e

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

72220f92694b python "python3" About a minute ago Up About a minute python-c1

Docker exec –it s

C:\Windows\system32>docker exec -it python-c1 bash

root@72220f92694b:/# print("hello")

bash: syntax error near unexpected token `"hello"'

root@72220f92694b:/# ls

bin boot dev etc home lib lib64 media mnt myapp opt proc root run sbin srv sys tmp usr var

root@72220f92694b:/# cd myapp

root@72220f92694b:/myapp# ls

app.py

root@72220f92694b:/myapp# python app.py

Hello from docker volume # python ko support karne ka liye

# python ko run ka liye bash , host pa project run kar rha docker pa container bana kee

root@72220f92694b:/myapp# python app.py

Hello from docker volume

root@72220f92694b:/myapp# python app.py

Hello from docker volume

Updates

# toh aise host to coanetiner data share kar sakte ho

C:\Windows\system32>docker stop python-c1

python-c1

**# Between container mount data volume**

C:\Windows\system32>docker volume create myvol

myvol

C:\Windows\system32>docker volume ls

DRIVER VOLUME NAME

local 1f71b4d91e267d75b476fb511579beeeb2b051abc9c7687a7bb059fdf151ecff

C:\Windows\system32>docker volume inspect myvol

[

{

"CreatedAt": "2025-02-04T14:33:22Z",

"Driver": "local",

"Labels": null,

"Mountpoint": "/var/lib/docker/volumes/myvol/\_data",

"Name": "myvol",

"Options": null,

"Scope": "local"

}

]

C:\Windows\system32>docker run --name c1 -itd -v myvol:/

**# is volume ko mount kara na hai myapp ka sath , c1 conatienr ka lye**

C:\Windows\system32>docker run --name c1 -itd -v myvol:/myapp python

d960140827fa91d04f314b3c5a448101ce82250df68fce011abf36da7cb4cb38

**# myvol mount to myapp for container c1**

C:\Windows\system32>docker run --name c2 -itd -v myvol:/myapp1 python

0daaa3d74ae8573ff1b2c2308debe8a5d8f58d1c3b7456436557bdd1804f937d

**# In terminal 2 ko ye mount karega**

# myvol volume ko 2 conainer show kar rha hai

C:\Windows\system32>docker run --name c1 -itd -v myvol:/myapp python

d960140827fa91d04f314b3c5a448101ce82250df68fce011abf36da7cb4cb38

in Termail 1

C:\Windows\system32>docker exec -it c1 bash

root@d960140827fa:/#

in termailnal2

C:\Windows\system32>docker run --name c2 -itd -v myvol:/myapp1 python

0daaa3d74ae8573ff1b2c2308debe8a5d8f58d1c3b7456436557bdd1804f937d

C:\Windows\system32>docker exec -it c2 bash

root@0daaa3d74ae8:/#

root@d960140827fa:/# ls

bin boot dev etc home lib lib64 media mnt myapp # myapp folder map hai myvol ka sath

C:\Windows\system32>docker exec -it c2 bash

root@0daaa3d74ae8:/# ls

bin boot dev etc home lib lib64 media mnt myapp1 # myapp1 se myvol map hai

C:\Windows\system32>docker exec -it c1 bash

root@d960140827fa:/# ls

bin boot dev etc home lib lib64 media mnt myapp opt proc root run sbin srv sys tmp usr var

root@d960140827fa:/# cd myapp

root@d960140827fa:/myapp# mkdir c1-data

root@d960140827fa:/myapp# ls

c1-data

root@d960140827fa:/myapp#

C:\Windows\system32>docker exec -it c2 bash

root@0daaa3d74ae8:/# ls

bin boot dev etc home lib lib64 media mnt myapp1 opt proc root run sbin srv sys tmp usr var

root@0daaa3d74ae8:/# cd myapp1

root@0daaa3d74ae8:/myapp1# ls

c1-data

root@0daaa3d74ae8:/myapp1#

**# dono terminal pa data share hoga aapka , between container data share karenge , toh volume create kara pahle phir mount kar diya dono continaer se**

# continaer delte kar do par volume delte nahi hoga ,same rahega data voleume ka persisit reahge

C:\Windows\system32>docker volume rm myvol

myvol

C:\Windows\system32>docker run --name c1 -itd -v /myvol python

2a5a7f50ce790bdb548eedc300510bd5dda9ad2b72b7dbc80919f3019f5a46cf # conainer or volume sath ma crate kara

C:\Windows\system32>docker run --name c1 -itd -v /myvol python

2a5a7f50ce790bdb548eedc300510bd5dda9ad2b72b7dbc80919f3019f5a46cf

C:\Windows\system32>docker volume ls

DRIVER VOLUME NAME

**# IS container ka volume dosera coantiner share karega**

Terminal 2

C:\Windows\system32>docker run --name -itd --volume from c1 python

Unable to find image 'c1:latest' locally

docker: Error response from daemon: pull access denied for c1, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.

See 'docker run --help'.

C:\Windows\system32>docker run --name c2 -itd --volumes-from c1 python

4f5dfda15836d784cf460241006a4d029f625a4a027424a498fad88880d8d036

C:\Windows\system32>

C:\Windows\system32>docker exec -it c1 bash

root@2a5a7f50ce79:/# bash milega

C:\Windows\system32>docker exec -it c2 bash

root@4f5dfda15836:/#

T1

C:\Windows\system32>docker exec -it c1 bash

root@2a5a7f50ce79:/# ls

bin boot dev etc home lib lib64 media mnt myvol opt proc root run sbin srv sys tmp usr var

root@2a5a7f50ce79:/# cd myvol

root@2a5a7f50ce79:/myvol# ls

root@2a5a7f50ce79:/myvol# touch p1.py

root@2a5a7f50ce79:/myvol# ls

p1.py

root@2a5a7f50ce79:/myvol#

C:\Windows\system32>docker exec -it c2 bash

root@4f5dfda15836:/# ls

bin boot dev etc home lib lib64 media mnt myvol opt proc root run sbin srv sys tmp usr var

root@4f5dfda15836:/# cd mvol

bash: cd: mvol: No such file or directory

root@4f5dfda15836:/# cd myvol

root@4f5dfda15836:/myvol# ls

p1.py

root@4f5dfda15836:/myvol#

# ALAG alag conaienr main data share ho rha hai

From Ubuntu

VOLUME[“/data”]

**# conainer ka andar mount point create karo iss wala ka andar , data directory main jo likhoge iske data outside volume pa kahi hoga data iske , volume nahi mount point create kar rha coantiner ka liye**

# volume carete nahi karta , is container ko use karke mount point banagega container

# new 1

Dockerfile

FROM ubuntu

VOLUME [ "/data" ]

PS C:\Users\my pc\Desktop\docker geekyshows\DockerVolume\volume\new> docker build -t mynewimage .

[+] Building 6.3s (4/5)

S C:\Users\my pc\Desktop\docker geekyshows\DockerVolume\volume\new> docker run --name c1 -itd -v myvol:/data mynewimage /bin/bash

f8a571252808d9a9286583b5951e3bbf674f26977e9a3a302b674bd16c895f9f

PS C:\Users\my pc\Desktop\docker geekyshows\DockerVolume\volume\new>

PS C:\Users\my pc\Desktop\docker geekyshows\DockerVolume\volume\new> docker exec -it c1 bash

xecute a command in a running container

PS C:\Users\my pc\Desktop\docker geekyshows\DockerVolume\volume\new> docker exec -it c1 bash

root@f8a571252808:/# ls

bin data etc lib media opt root sbin sys usr

boot dev home lib64 mnt proc run srv tmp var

**# khud ka ocnainter banao aur data share karo different conainer ka**

PS C:\Users\my pc\Desktop\docker geekyshows\DockerVolume\volume\new> docker inspect c1

[

{

"Id": "f8a571252808d9a9286583b5951e3bbf674f26977e9a3a302b674bd16c895f9f",

"Created": "2025-02-04T15:04:57.27578657Z",

"Path": "/bin/bash",

"Args": [],

"State": {

DOCKER Networking

C:\Windows\system32>docker network ls

NETWORK ID NAME DRIVER SCOPE

e9799a33f543 bridge bridge local

8a6c0d7761dc host host local

b08384fa2bb6 none null local

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

C:\Windows\system32>docker pull ngninx

Using default tag: latest

Error response from daemon: pull access denied for ngninx, repository does not exist or may require 'docker login'

C:\Windows\system32>docker pull nginx

Using default tag: latest

latest: Pulling from library/nginx

976e8f6b25dd: Download complete

C:\Windows\system32>docker pull alpine

Using default tag: latest

docker.io/library/alpine:latest

$ by defuault it pull images from docker hub

C:\Windows\system32>docker network inpsect bridge

Usage: docker network COMMAND

C:\Windows\system32>docker network inspect bridge

[

{

"Name": "bridge",

"Id": "e9799a33f543b2b514a79aabfe6e59c07910fd229d937846546c34f4f0fc703e",

"Created": "2025-02-06T14:42:13.515180196Z",

"Scope": "local",

"Network": ""

},

"ConfigOnly": false,

]

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest 91734281c0eb 18 hours ago 279MB

alpine latest 56fa17d2a7e7 4 weeks ago 12.1MB

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest 91734281c0eb 18 hours ago 279MB

alpine latest 56fa17d2a7e7 4 weeks ago 12.1MB

C:\Windows\system32>docker run --name c1 -itd alpine ash

4a843a3ce801ecd6109fef84a82ad9797d6c59628907d55bbc73b531d97235a9

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker inspect bridge

[

C:\Windows\system32>docker run --name c2 -itd alpine ash

60efbae0f180c5ad8c4a5f8cc04785fa57a7fe92b835a4554bf3656e4e00bee8

# any container you create is present insdie the bridge, by default it uses bridge netwroks

C:\Windows\system32>docker inspect bridge

[

{

"Name": "bridge",

"Id": "e9799a33f543b2b514a79aabfe6e59c07910fd229d937846546c34f4f0fc703e",

"Created": "2025-02-06T14:42:13.515180196Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": null,

"Config": [

{

"Subnet": "172.17.0.0/16",

"Gateway": "172.17.0.1"

}

]

},

"Internal": false,

"Attachable": false,

C:\Windows\system32>docker attach c1 = # login into c1 and

/ # ip addr show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

5: eth0@if6: <BROADCAST,MULTICAST,UP,LOWER\_UP,M-DOWN> mtu 1500 qdisc noqueue state UP

link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff

inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0

valid\_lft forever preferred\_lft forever

C:\Windows\system32>docker attach c1

/ # ip addr show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

5: eth0@if6: <BROADCAST,MULTICAST,UP,LOWER\_UP,M-DOWN> mtu 1500 qdisc noqueue state UP

link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff

inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0

valid\_lft forever preferred\_lft forever

/ # ping -c 5 google.com

PING google.com (142.250.199.142): 56 data bytes

64 bytes from 142.250.199.142: seq=0 ttl=63 time=38.559 ms

64 bytes from 142.250.199.142: seq=1 ttl=63 time=40.283 ms

64 bytes from 142.250.199.142: seq=2 ttl=63 time=34.946 ms

64 bytes from 142.250.199.142: seq=3 ttl=63 time=36.768 ms

64 bytes from 142.250.199.142: seq=4 ttl=63 time=34.483 ms

--- google.com ping statistics ---

5 packets transmitted, 5 packets received, 0% packet loss

round-trip min/avg/max = 34.483/37.007/40.283 ms

/ # ping -c 5 172.17.0.3/16

ping: bad address '172.17.0.3/16'

/ # ping -c 5 172.17.0.3 $ ping c2 from c1 sucdcefully

PING 172.17.0.3 (172.17.0.3): 56 data bytes

64 bytes from 172.17.0.3: seq=0 ttl=64 time=1.208 ms

64 bytes from 172.17.0.3: seq=1 ttl=64 time=0.238 ms

64 bytes from 172.17.0.3: seq=2 ttl=64 time=0.113 ms

64 bytes from 172.17.0.3: seq=3 ttl=64 time=0.191 ms

64 bytes from 172.17.0.3: seq=4 ttl=64 time=0.305 ms

--- 172.17.0.3 ping statistics ---

5 packets transmitted, 5 packets received, 0% packet loss

round-trip min/avg/max = 0.113/0.411/1.208 ms

ctrl p and crtrl q

C:\Windows\system32>docker network create --driver bridge net1

0bb59ab32b49f119eda808272d517e74c95811443b9d1e6a18412c60d5add9dc

C:\Windows\system32>docker network ls

NETWORK ID NAME DRIVER SCOPE

e9799a33f543 bridge bridge local

8a6c0d7761dc host host local

0bb59ab32b49 net1 bridge local

b08384fa2bb6 none null local

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

60efbae0f180 alpine "ash" 6 minutes ago Up 6 minutes c2

4a843a3ce801 alpine "ash" 12 minutes ago Up 12 minutes c1

C:\Windows\system32>docker network create --driver bridge net1

0bb59ab32b49f119eda808272d517e74c95811443b9d1e6a18412c60d5add9dc

C:\Windows\system32>docker network ls

NETWORK ID NAME DRIVER SCOPE

e9799a33f543 bridge bridge local

8a6c0d7761dc host host local

0bb59ab32b49 net1 bridge local

b08384fa2bb6 none null local

C:\Windows\system32>docker inpect bridge

docker: 'inpect' is not a docker command.

See 'docker --help'

C:\Windows\system32>docker inspect bridge

[

{

"Name": "bridge",

"Id": "e9799a33f543b2b514a79aabfe6e59c07910fd229d937846546c34f4f0fc703e",

"Created": "2025-02-06T14:42:13.515180196Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": null,

"Config": [

{

"Subnet": "172.17.0.0/16",

"Gateway": "172.17.0.1"

}

]

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {

"4a843a3ce801ecd6109fef84a82ad9797d6c59628907d55bbc73b531d97235a9": {

"Name": "c1",

"EndpointID": "dd41606202e86fcfeca1105aab203187cdfd284df262a1eeb61014013272e202",

"MacAddress": "02:42:ac:11:00:02",

"IPv4Address": "172.17.0.2/16",

"IPv6Address": ""

},

"60efbae0f180c5ad8c4a5f8cc04785fa57a7fe92b835a4554bf3656e4e00bee8": {

"Name": "c2",

"EndpointID": "546d8dbbcf0671fa1b5b10be4f7cad655019d7c9c07b75645d3b6d1297f53e8e",

"MacAddress": "02:42:ac:11:00:03",

"IPv4Address": "172.17.0.3/16",

"IPv6Address": ""

}

},

"Options": {

"com.docker.network.bridge.default\_bridge": "true",

"com.docker.network.bridge.enable\_icc": "true",

"com.docker.network.bridge.enable\_ip\_masquerade": "true",

"com.docker.network.bridge.host\_binding\_ipv4": "0.0.0.0",

"com.docker.network.bridge.name": "docker0",

"com.docker.network.driver.mtu": "1500"

},

"Labels": {}

}

]

C:\Windows\system32>docker network inspect net1 $ createing customer network attach container to c3 and c4

[

{

"Name": "net1",

"Id": "0bb59ab32b49f119eda808272d517e74c95811443b9d1e6a18412c60d5add9dc",

"Created": "2025-02-06T15:49:04.615142913Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": {},

"Config": [

{

"Subnet": "172.18.0.0/16",

"Gateway": "172.18.0.1"

}

]

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {},

"Options": {},

"Labels": {}

}

]

C:\Windows\system32>docker run --name c3 -itd --network net1 alpine ash

#create c3 coantienr c3 and attach to network net1

0fb03a4f4eb3c84da128a2fbb7f37650e5fb1d0b178b3203abffe90366a019b0

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0fb03a4f4eb3 alpine "ash" 5 seconds ago Up 4 seconds c3

60efbae0f180 alpine "ash" 11 minutes ago Up 11 minutes c2

4a843a3ce801 alpine "ash" 17 minutes ago Up 17 minutes c1

C:\Windows\system32>docker container ls

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0fb03a4f4eb3 alpine "ash" 12 seconds ago Up 11 seconds c3

60efbae0f180 alpine "ash" 12 minutes ago Up 12 minutes c2

4a843a3ce801 alpine "ash" 17 minutes ago Up 17 minutes c1

C:\Windows\system32>docker inspect network net1

[

{

"Name": "net1",

"Id": "0bb59ab32b49f119eda808272d517e74c95811443b9d1e6a18412c60d5add9dc",

"Created": "2025-02-06T15:49:04.615142913Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": {},

"Config": [

{

"Subnet": "172.18.0.0/16",

"Gateway": "172.18.0.1"

}

]

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {

"0fb03a4f4eb3c84da128a2fbb7f37650e5fb1d0b178b3203abffe90366a019b0": {

"Name": "c3",

"EndpointID": "7c335e97a5d9964cff432e39eef1552d4ef656ba3e9b78b7ed7e3cbcf539790b",

"MacAddress": "02:42:ac:12:00:02",

"IPv4Address": "172.18.0.2/16",

"IPv6Address": ""

}

},

"Options": {},

"Labels": {}

}

]

Error: No such object: network

C:\Windows\system32>docker run --name c4 -itd --network net1 alpine ash

cb222a4987c126745e5e812efb28a05cef42af9716374af3f160a0904c85901b

C:\Windows\system32>docker inspect network net1

[

{

"Name": "net1",

"Id": "0bb59ab32b49f119eda808272d517e74c95811443b9d1e6a18412c60d5add9dc",

"Created": "2025-02-06T15:49:04.615142913Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": {},

"Config": [

{

"Subnet": "172.18.0.0/16",

"Gateway": "172.18.0.1"

}

]

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {

"0fb03a4f4eb3c84da128a2fbb7f37650e5fb1d0b178b3203abffe90366a019b0": {

"Name": "c3",

"EndpointID": "7c335e97a5d9964cff432e39eef1552d4ef656ba3e9b78b7ed7e3cbcf539790b",

"MacAddress": "02:42:ac:12:00:02",

"IPv4Address": "172.18.0.2/16",

"IPv6Address": ""

},

"cb222a4987c126745e5e812efb28a05cef42af9716374af3f160a0904c85901b": {

"Name": "c4",

"EndpointID": "076a2f08fd85c8921bee7f65f901fa2a599e1a46e11d25d928c4382ec38680a6",

"MacAddress": "02:42:ac:12:00:03",

"IPv4Address": "172.18.0.3/16",

"IPv6Address": ""

}

},

"Options": {},

"Labels": {}

}

]

Error: No such object: network

**$ two container are present inside the network net1 such as c3 and c4**

C:\Windows\system32>docker inspect bridge

[

{

"Name": "bridge",

"Id": "e9799a33f543b2b514a79aabfe6e59c07910fd229d937846546c34f4f0fc703e",

"Created": "2025-02-06T14:42:13.515180196Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": null,

"Config": [

{

"Subnet": "172.17.0.0/16",

"Gateway": "172.17.0.1"

}

]

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {

"4a843a3ce801ecd6109fef84a82ad9797d6c59628907d55bbc73b531d97235a9": {

"Name": "c1",

"EndpointID": "dd41606202e86fcfeca1105aab203187cdfd284df262a1eeb61014013272e202",

"MacAddress": "02:42:ac:11:00:02",

"IPv4Address": "172.17.0.2/16",

"IPv6Address": ""

},

"60efbae0f180c5ad8c4a5f8cc04785fa57a7fe92b835a4554bf3656e4e00bee8": {

"Name": "c2",

"EndpointID": "546d8dbbcf0671fa1b5b10be4f7cad655019d7c9c07b75645d3b6d1297f53e8e",

"MacAddress": "02:42:ac:11:00:03",

"IPv4Address": "172.17.0.3/16",

"IPv6Address": ""

}

},

"Options": {

"com.docker.network.bridge.default\_bridge": "true",

"com.docker.network.bridge.enable\_icc": "true",

"com.docker.network.bridge.enable\_ip\_masquerade": "true",

"com.docker.network.bridge.host\_binding\_ipv4": "0.0.0.0",

"com.docker.network.bridge.name": "docker0",

"com.docker.network.driver.mtu": "1500"

},

"Labels": {}

}

]

C:\Windows\system32>C:\Windows\system32>docker attach c1

/ # ip addr show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

5: eth0@if6: <BROADCAST,MULTICAST,UP,LOWER\_UP,M-DOWN> mtu 1500 qdisc noqueue state UP

link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff

inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0

valid\_lft forever preferred\_lft forever

/ # pint -c 5 172.17.0.3

ash: pint: not found

/ # ping -c 5 172.17.0.3

PING 172.17.0.3 (172.17.0.3): 56 data bytes

64 bytes from 172.17.0.3: seq=0 ttl=64 time=0.145 ms

64 bytes from 172.17.0.3: seq=1 ttl=64 time=0.109 ms

64 bytes from 172.17.0.3: seq=2 ttl=64 time=0.088 ms

64 bytes from 172.17.0.3: seq=3 ttl=64 time=0.088 ms

64 bytes from 172.17.0.3: seq=4 ttl=64 time=0.106 ms

--- 172.17.0.3 ping statistics ---

5 packets transmitted, 5 packets received, 0% packet loss

round-trip min/avg/max = 0.088/0.107/0.145 ms

/ # ping -c 5 172.18.0.2 # c1 se c3 ka ip ping kara false comminucate nahi ho payge a

PING 172.18.0.2 (172.18.0.2): 56 data bytes

**$ from container c1 sending reques to c2 and successfully getting response**

C:\Windows\system32>docker inspect bridge

[

{

"Name": "bridge",

"Id": "e9799a33f543b2b514a79aabfe6e59c07910fd229d937846546c34f4f0fc703e",

"Created": "2025-02-06T14:42:13.515180196Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": null,

"Config": [

{

"Subnet": "172.17.0.0/16",

"Gateway": "172.17.0.1"

}

]

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {

"4a843a3ce801ecd6109fef84a82ad9797d6c59628907d55bbc73b531d97235a9": {

"Name": "c1",

"EndpointID": "dd41606202e86fcfeca1105aab203187cdfd284df262a1eeb61014013272e202",

"MacAddress": "02:42:ac:11:00:02",

"IPv4Address": "172.17.0.2/16",

"IPv6Address": ""

},

"60efbae0f180c5ad8c4a5f8cc04785fa57a7fe92b835a4554bf3656e4e00bee8": {

"Name": "c2",

"EndpointID": "546d8dbbcf0671fa1b5b10be4f7cad655019d7c9c07b75645d3b6d1297f53e8e",

"MacAddress": "02:42:ac:11:00:03",

"IPv4Address": "172.17.0.3/16",

"IPv6Address": ""

}

},

"Options": {

"com.docker.network.bridge.default\_bridge": "true",

"com.docker.network.bridge.enable\_icc": "true",

"com.docker.network.bridge.enable\_ip\_masquerade": "true",

"com.docker.network.bridge.host\_binding\_ipv4": "0.0.0.0",

"com.docker.network.bridge.name": "docker0",

"com.docker.network.driver.mtu": "1500"

},

"Labels": {}

}

]

C:\Windows\system32>docker insect network net1

docker: 'insect' is not a docker command.

See 'docker --help'

C:\Windows\system32>docker network inspect net1

[

{

"Name": "net1",

"Id": "0bb59ab32b49f119eda808272d517e74c95811443b9d1e6a18412c60d5add9dc",

"Created": "2025-02-06T15:49:04.615142913Z",

"Scope": "local",

"Driver": "bridge",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": {},

"Config": [

{

"Subnet": "172.18.0.0/16",

"Gateway": "172.18.0.1"

}

]

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {

"0fb03a4f4eb3c84da128a2fbb7f37650e5fb1d0b178b3203abffe90366a019b0": {

"Name": "c3",

"EndpointID": "7c335e97a5d9964cff432e39eef1552d4ef656ba3e9b78b7ed7e3cbcf539790b",

"MacAddress": "02:42:ac:12:00:02",

"IPv4Address": "172.18.0.2/16",

"IPv6Address": ""

},

"cb222a4987c126745e5e812efb28a05cef42af9716374af3f160a0904c85901b": {

"Name": "c4",

"EndpointID": "076a2f08fd85c8921bee7f65f901fa2a599e1a46e11d25d928c4382ec38680a6",

"MacAddress": "02:42:ac:12:00:03",

"IPv4Address": "172.18.0.3/16",

"IPv6Address": ""

}

},

"Options": {},

"Labels": {}

}

]

C:\Windows\system32>docket attach c3

'docket' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>docker attach c3

/ # ping -c 5 ^C

/ # ping -c 5 172.18.0.3

PING 172.18.0.3 (172.18.0.3): 56 data bytes

64 bytes from 172.18.0.3: seq=0 ttl=64 time=2.820 ms

64 bytes from 172.18.0.3: seq=1 ttl=64 time=0.186 ms

64 bytes from 172.18.0.3: seq=2 ttl=64 time=0.177 ms

64 bytes from 172.18.0.3: seq=3 ttl=64 time=0.091 ms

64 bytes from 172.18.0.3: seq=4 ttl=64 time=0.157 ms

--- 172.18.0.3 ping statistics ---

5 packets transmitted, 5 packets received, 0% packet loss

round-trip min/avg/max = 0.091/0.686/2.820 ms

/ # ip addr show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

10: eth0@if11: <BROADCAST,MULTICAST,UP,LOWER\_UP,M-DOWN> mtu 1500 qdisc noqueue state UP

link/ether 02:42:ac:12:00:02 brd ff:ff:ff:ff:ff:ff

inet 172.18.0.2/16 brd 172.18.255.255 scope global eth0

valid\_lft forever preferred\_lft forever

/ # ping -c 5 172.17.0.2

PING 172.17.0.2 (172.17.0.2): 56 data bytes

--- 172.17.0.2 ping statistics ---

5 packets transmitted, 0 packets received, 100% packet loss

/ # exit

**# container can only communicate when they are in same network like in a bridge or in net1 custome network – when they both are at same network**

C:\Windows\system32>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

cb222a4987c1 alpine "ash" 9 minutes ago Up 9 minutes c4

60efbae0f180 alpine "ash" 23 minutes ago Up 23 minutes c2

4a843a3ce801 alpine "ash" 29 minutes ago Up 29 minutes c1

C:\Windows\system32>docker container rm c1 c2 c4

Error response from daemon: cannot remove container "/c1": container is running: stop the container before removing or force remove

Error response from daemon: cannot remove container "/c2": container is running: stop the container before removing or force remove

Error response from daemon: cannot remove container "/c4": container is running: stop the container before removing or force remove

C:\Windows\system32>docker container stop c1 c2 c4

C:\Windows\system32>docker rm c1 c2 c4

C:\Windows\system32>docker container ls

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0fb03a4f4eb3 alpine "ash" 13 minutes ago Exited (1) About a minute ago c3

C:\Windows\system32>docker rm 0f

0f

C:\Windows\system32>

C:\Windows\system32>docker network inspect host

[

{

"Name": "host",

"Id": "8a6c0d7761dce119ee0ea60dc9e19c6853a93a909bac39aa10fdc68ecc9b29b9",

"Created": "2025-02-06T14:42:13.452981794Z",

"Scope": "local",

"Driver": "host",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": null,

"Config": null

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {},

"Options": {},

"Labels": {}

}

]

**# is container se hsot se sattach kar diya hamnee**

C:\Windows\system32>docker run -itd --network host --name nginx1 nginx

e8154c374dba46beafd4aaacf1e29325cd156320db4b8e1e231d98af4026e1e4

C:\Windows\system32>docker inspect host

[

{

"Name": "host",

"Id": "8a6c0d7761dce119ee0ea60dc9e19c6853a93a909bac39aa10fdc68ecc9b29b9",

"Created": "2025-02-06T14:42:13.452981794Z",

"Scope": "local",

"Driver": "host",

"EnableIPv6": false,

"IPAM": {

"Driver": "default",

"Options": null,

"Config": null

},

"Internal": false,

"Attachable": false,

"Ingress": false,

"ConfigFrom": {

"Network": ""

},

"ConfigOnly": false,

"Containers": {

"e8154c374dba46beafd4aaacf1e29325cd156320db4b8e1e231d98af4026e1e4": {

"Name": "nginx1",

"EndpointID": "276b7c165f9f22a246677a5c524986c8654fbec89d7aedb90d9a8a49065afec8",

"MacAddress": "",

"IPv4Address": "",

"IPv6Address": ""

}

},

"Options": {},

"Labels": {}

}

**Overlay network – alag host pa diff container hai who aapas main communicate kar paye toh hsot akr use karenge**

**Closeted setup ka use kar sakte hai , kubertens main aache se semajh payega**

**DOCKER Compose**

#multi container appn ka liye , jusme alag component chaiye application ko run ka liye ,

jaise different iamge , contaienrs chaiye hamko toh docker compose use karo

**Redis mysql python sab chij ek sath use hori toh ham docker compose ka use karenge**

We write yml file and define all services inside container

# appn ko run kanrne ka liye multiple cheej chahiye toh docker compose use karo , ek cmd main sare kam kar skate hai

C:\Windows\system32>cd C:\Users\my pc\Desktop\docker geekyshows

C:\Users\my pc\Desktop\docker geekyshows>cd compose

C:\Users\my pc\Desktop\docker geekyshows\compose>docker pull redis

Using default tag: latest

latest: Pulling from library/redis

c29f5b76f736: Download complete

f5bc47c37726: Download complete

d128ccd842a6: Download complete

20eea55b3ebb: Download complete

5de2dd3ff2ef: Download complete

6c334acf232e: Download complete

4f4fb700ef54: Download complete

3090e1a50a6c: Download complete

Digest: sha256:93a8d83b707d0d6a1b9186edecca2e37f83722ae0e398aee4eea0ff17c2fad0e

Status: Downloaded newer image for redis:latest

docker.io/library/redis:latest

C:\Users\my pc\Desktop\docker geekyshows\compose>docker pull alpine

Using default tag: latest

latest: Pulling from library/alpine

1f3e46996e29: Download complete

Digest: sha256:56fa17d2a7e7f168a043a2712e63aed1f8543aeafdcee47c58dcffe38ed51099

Status: Downloaded newer image for alpine:latest

docker.io/library/alpine:latest

C:\Users\my pc\Desktop\docker geekyshows\compose>docker pull mysql

Using default tag: latest

latest: Pulling from library/mysql

4473ac30a868: Download complete

b48319207fe2: Download complete

274b57fed122: Download complete

bdf87f630bc8: Download complete

b49812044423: Download complete

886200dd5e38: Download complete

7dd0a00427a8: Download complete

e4312bbe95c6: Download complete

2f4e973a399a: Download complete

1e59b1e9b7fc: Download complete

Digest: sha256:d56d039139a7f3b71f6d1c9f07ca4ee9f977b0fca13acdd27a1b13bfd4a4e3be

Status: Downloaded newer image for mysql:latest

docker.io/library/mysql:latest

C:\Users\my pc\Desktop\docker geekyshows\compose>docker pull nginx

Using default tag: latest

latest: Pulling from library/nginx

e19db8451adb: Download complete

c558df217949: Download complete

976e8f6b25dd: Download complete

84cade77a831: Download complete

24ff42a0d907: Download complete

6c78b0ba1a32: Download complete

Digest: sha256:91734281c0ebfc6f1aea979cffeed5079cfe786228a71cc6f1f46a228cde6e34

Status: Downloaded newer image for nginx:latest

docker.io/library/nginx:latest

C:\Users\my pc\Desktop\docker geekyshows\compose>

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose version

Docker Compose version v2.31.0-desktop.2

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose version

Docker Compose version v2.31.0-desktop.2

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose --help

Usage: docker compose [OPTIONS] COMMAND

Docker-compose.yml

services:

  web:

    image: "nginx"

  rcache:

    image: "redis:6.0-alpine3.18"

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose config

name: compose

services:

rcache:

image: redis:6.0-alpine3.18

networks:

default: null

web:

image: nginx

networks:

default: null

networks:

default:

name: compose\_default

C:\Users\my pc\Desktop\docker geekyshows\compose>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest 91734281c0eb 21 hours ago 279MB

mysql latest d56d039139a7 2 weeks ago 1.09GB

alpine latest 56fa17d2a7e7 4 weeks ago 12.1MB

redis latest 93a8d83b707d 4 weeks ago 173MB

C:\Users\my pc\Desktop\docker geekyshows\compose>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Users\my pc\Desktop\docker geekyshows\compose>docker container ls

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Users\my pc\Desktop\docker geekyshows\compose>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx latest 91734281c0eb 21 hours ago 279MB

mysql latest d56d039139a7 2 weeks ago 1.09GB

alpine latest 56fa17d2a7e7 4 weeks ago 12.1MB

redis latest 93a8d83b707d 4 weeks ago 173MB

C:\Users\my pc\Desktop\docker geekyshows\compose>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Users\my pc\Desktop\docker geekyshows\compose>docker container ls

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

C:\Users\my pc\Desktop\docker geekyshows\compose>docker network ls

NETWORK ID NAME DRIVER SCOPE

e9799a33f543 bridge bridge local

8a6c0d7761dc host host local

0bb59ab32b49 net1 bridge local

b08384fa2bb6 none null local

C:\Users\my pc\Desktop\docker geekyshows\compose>docker volume ls

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose up –d #create and start containers

[+] Running 8/8

✔ rcache Pulled 11.0s

✔ 39333280c8c6 Download complete 2.6s

✔ f12f505e3a3c Download complete 2.0s

✔ d37d50d32a98 Download complete 1.5s

[+] Running 3/3

✔ Network compose\_default Created 0.5s

✔ Container compose-rcache-1 Started 5.8s

✔ Container compose-web-1 Started

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose up -d

[+] Running 8/8

✔ rcache Pulled 11.0s

✔ 39333280c8c6 Download complete 2.6s

✔ c105db640f3f Download complete 1.4s

✔ c926b61bad3b Download complete 1.6s

✔ cdf33658b38c Download complete 1.6s

✔ 4f4fb700ef54 Already exists 0.0s

✔ f12f505e3a3c Download complete 2.0s

✔ d37d50d32a98 Download complete 1.5s

[+] Running 3/3

✔ Network compose\_default Created 0.5s

✔ Container compose-rcache-1 Started 5.8s

✔ Container compose-web-1 Started 5.8s

C:\Users\my pc\Desktop\docker geekyshows\compose>docker container ls

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0d1f79919ffc nginx "/docker-entrypoint.…" About a minute ago Up About a minute 80/tcp compose-web-1

29fce167b225 redis:6.0-alpine3.18 "docker-entrypoint.s…" About a minute ago Up About a minute 6379/tcp compose-rcache-1

C:\Users\my pc\Desktop\docker geekyshows\compose>docker network ls

NETWORK ID NAME DRIVER SCOPE

e9799a33f543 bridge bridge local

fc607f58147e compose\_default bridge local

8a6c0d7761dc host host local

0bb59ab32b49 net1 bridge local

b08384fa2bb6 none null local

C:\Users\my pc\Desktop\docker geekyshows\compose>docker volume ls

DRIVER VOLUME NAME

local 02dd6307ec55e7d9ba2e60021fbe13c60e71c4184d4b5881bd66f3e5c0ef0cde

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose ps

NAME IMAGE COMMAND SERVICE CREATED STATUS PORTS

compose-rcache-1 redis:6.0-alpine3.18 "docker-entrypoint.s…" rcache 3 minutes ago Up 3 minutes 6379/tcp

compose-web-1 nginx "/docker-entrypoint.…" web 3 minutes ago Up 3 minutes 80/tcp

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose exec web bash

root@0d1f79919ffc:/# ls

bin boot dev docker-entrypoint.d docker-entrypoint.sh etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

root@0d1f79919ffc:/# exi

bash: exi: command not found

root@0d1f79919ffc:/# exit

exit

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose stop

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose down --volume

$ Remove all the containers that were started from docker compose

[+] Running 3/3

✔ Container compose-web-1 Removed 1.1s

✔ Container compose-rcache-1 Removed 1.0s

✔ Network compose\_default Removed

Using of tag in

.env

TAG = 6.0-alpine3.18

Docker-compose.yml

Docker-compose.yml

services:

  web:

    image: "nginx"

  rcache:

    image: "redis:${TAG}"

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose up -d

[+] Running 3/3

✔ Network compose\_default Created 0.1s

✔ Container compose-web-1 Started 2.1s

✔ Container compose-rcache-1 Started

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose up -d

[+] Running 3/3

✔ Network compose\_default Created 0.1s

✔ Container compose-web-1 Started 2.1s

✔ Container compose-rcache-1 Started 2.0s

C:\Users\my pc\Desktop\docker geekyshows\compose>docker container ls

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

bfe46ee140ea nginx "/docker-entrypoint.…" 47 seconds ago Up 45 seconds 80/tcp compose-web-1

5f8b8010153b redis:6.0-alpine3.18 "docker-entrypoint.s…" 47 seconds ago Up 45 seconds 6379/tcp compose-rcache-1

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose config

name: compose

services:

rcache:

image: redis:6.0-alpine3.18

networks:

default: null

web:

image: nginx

networks:

default: null

networks:

default:

name: compose\_default

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose down --volume

# sab kuch delete ho jayega

[+] Running 3/3

✔ Container compose-rcache-1 Removed 0.8s

✔ Container compose-web-1 Removed 1.0s

✔ Network compose\_default Removed

Docker-compose.yml

services:

  web:

    image: "nginx"

  rcache:

    image: "redis:${TAG}"

  db:

    image: mysql

    environment:

      - MYSQL\_ROOT\_PASSWORD=deep

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose up -d

[+] Running 3/3

[+] Running 4/4

✔ Network compose\_default Created 0.2s

✔ Container compose-web-1 Started 3.5s

✔ Container compose-rcache-1 Started 3.9s

✔ Container compose-db-1 Started 3.8s

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose ls

NAME STATUS CONFIG FILES

compose running(3) C:\Users\my pc\Desktop\docker geekyshows\compose\docker-compose.yml

C:\Users\my pc\Desktop\docker geekyshows\compose>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

aef185753c8a nginx "/docker-entrypoint.…" 43 seconds ago Up 41 seconds 80/tcp compose-web-1

a5e67d82c1b4 redis:6.0-alpine3.18 "docker-entrypoint.s…" 43 seconds ago Up 41 seconds 6379/tcp compose-rcache-1

68ddb829a042 mysql "docker-entrypoint.s…" 44 seconds ago Up 41 seconds 3306/tcp, 33060/tcp compose-db-1

C:\Users\my pc\Desktop\docker geekyshows\compose>docker network ls

NETWORK ID NAME DRIVER SCOPE

e9799a33f543 bridge bridge local

6404fa7f904a compose\_default bridge local

8a6c0d7761dc host host local

0bb59ab32b49 net1 bridge local

b08384fa2bb6 none null local

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose down --volumes

[+] Running 4/4

✔ Container compose-rcache-1 Removed 0.7s

✔ Container compose-web-1 Removed 0.9s

✔ Container compose-db-1 Removed 4.4s

✔ Network compose\_default Removed

services:

  web:

    image: "nginx"

  rcache:

    image: "redis:${TAG}"

  db:

    image: mysql

    # environment:

    #   - MYSQL\_ROOT\_PASSWORD=deep

    env\_file:

      - mysqlconfig.env

mysqlconfig.env

MYSQL\_ROOT\_PASSWORD=deep

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose config

name: compose

services:

db:

environment:

MYSQL\_ROOT\_PASSWORD: deep

image: mysql

networks:

default: null

rcache:

image: redis:6.0-alpine3.18

networks:

default: null

web:

image: nginx

networks:

default: null

networks:

default:

name: compose\_default

services:

  web:

    image: "nginx"

  rcache:

    image: "redis:${TAG}"

    profiles:

      - rediscache

  db:

    image: mysql

    # environment:

    #   - MYSQL\_ROOT\_PASSWORD=deep

    env\_file:

      - mysqlconfig.env

$ toh ab redis execute or show nahi hoga , hame batna padega tab lega redis kpo , profile manualay diya hai redis ka toh batana padega

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose config

name: compose

services:

db:

environment:

MYSQL\_ROOT\_PASSWORD: deep

image: mysql

networks:

default: null

web:

image: nginx

networks:

default: null

networks:

default:

name: compose\_default

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose --profile rediscache config

# we need to manully need to tell then it will show inside config files

name: compose

services:

db:

environment:

MYSQL\_ROOT\_PASSWORD: deep

image: mysql

networks:

default: null

rcache:

profiles:

- rediscache

image: redis:6.0-alpine3.18

networks:

default: null

web:

image: nginx

networks:

default: null

networks:

default:

name: compose\_default

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose up -d

[+] Running 2/2

✔ web Pulled

[+] Running 3/3Desktop\docker geekyshows\compose>docker compose down --volumes

✔ Container compose-db-1 Removed 3.0s

✔ Container compose-web-1 Removed 0.7s

✔ Network compose\_default Removed 0.4s

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose --profile rediscache up -d

# it will start all containers including rcahce first , we need to specify profile inside cmd to run rcache container

[+] Running 4/4

✔ Network compose\_default Created 0.1s

✔ Container compose-db-1 Started 2.2s

✔ Container compose-web-1 Started 2.2s

✔ Container compose-rcache-1 Started

$ab redis ka bhi container badega db or web service ka sath main hi

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose down --volume

time="2025-02-07T00:56:35+05:30" level=warning msg="--volume is deprecated, please use --volumes"

time="2025-02-07T00:56:35+05:30" level=warning msg="--volume is deprecated, please use --volumes"

time="2025-02-07T00:56:35+05:30" level=warning msg="--volume is deprecated, please use --volumes"

time="2025-02-07T00:56:35+05:30" level=warning msg="--volume is deprecated, please use --volumes"

[+] Running 3/2

✔ Container compose-web-1 Removed 0.6s

✔ Container compose-db-1 Removed 1.8s

! Network compose\_default Resource is still in use

$ redis abhi bhi removed nahi hua aapka

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose --profile rediscache down --volumes

[+] Running 2/2

✔ Container compose-rcache-1 Removed 0.6s

✔ Network compose\_default Removed 0.3s

services:

  web:

    image: "nginx"

    ports:

      - "8000:80"

  rcache:

    image: "redis:${TAG}"

    profiles:

      - rediscache

  db:

    image: mysql

    # environment:

    #   - MYSQL\_ROOT\_PASSWORD=deep

    env\_file:

      - mysqlconfig.env

$ PORT EXPOSE 8000 host machine ka port hamari machine ka port , or conainer ka port hai 80hamara , 8000 ko bind kara hamne 80 ka port se , 80 port container ka port hai

services:

  web:

    image: "nginx"

    ports:

      - "8000:80"

  rcache:

    image: "redis:${TAG}"

    # profiles:

    #   - rediscache

  db:

    image: mysql

    # environment:

    #   - MYSQL\_ROOT\_PASSWORD=deep

    env\_file:

      - mysqlconfig.env

[+] Running 4/4Desktop\docker geekyshows\compose>docker compose up -d

✔ Network compose\_default Created 0.1s

✔ Container compose-db-1 Started 2.5s

✔ Container compose-web-1 Started 2.8s

✔ Container compose-rcache-1 Started 2.8s

C:\Users\my pc\Desktop\docker geekyshows\compose>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

93312535e96c nginx "/docker-entrypoint.…" 31 seconds ago Up 29 seconds 0.0.0.0:8000->80/tcp compose-web-1

bfa7126b84ba redis:6.0-alpine3.18 "docker-entrypoint.s…" 31 seconds ago Up 29 seconds 6379/tcp compose-rcache-1

fae6b6ff0ec7 mysql "docker-entrypoint.s…" 31 seconds ago Up 29 seconds 3306/tcp, 33060/tcp compose-db-1

**8000 ko bind kara 80 sath nginx ka liye**

**Redis apache sabke port expose kar sakte aise hi**

C:\Users\my pc\Desktop\docker geekyshows\compose>docker network ls

NETWORK ID NAME DRIVER SCOPE

e9799a33f543 bridge bridge local

9e2012bf7cf9 compose\_default bridge local

8a6c0d7761dc host host local

0bb59ab32b49 net1 bridge local

b08384fa2bb6 none null local

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose down --volume

[+] Running 4/4

✔ Container compose-web-1 Removed 0.8s

✔ Container compose-rcache-1 Removed 1.1s

✔ Container compose-db-1 Removed 1.9s

✔ Network compose\_default Removed 0.4s

C:\Users\my pc\Desktop\docker geekyshows\compose>docker network ls

NETWORK ID NAME DRIVER SCOPE

e9799a33f543 bridge bridge local

8a6c0d7761dc host host local

0bb59ab32b49 net1 bridge local

b08384fa2bb6 none null local

services:

  web:

    image: "nginx"

    ports:

      - "8000:80"

  rcache:

    image: "redis:${TAG}"

    # profiles:

    #   - rediscache

  db:

    image: mysql

    # environment:

    #   - MYSQL\_ROOT\_PASSWORD=deep

    env\_file:

      - mysqlconfig.env

networks:

  my\_network:

    driver: bridge

**# ye network in service main checiye specify akro**

services:

  web:

    image: "nginx"

    ports:

      - "8000:80"

    networks:

      - my\_network

  rcache:

    image: "redis:${TAG}"

    # profiles:

    #   - rediscache

    networks:

      - my\_network

  db:

    image: mysql

    # environment:

    #   - MYSQL\_ROOT\_PASSWORD=deep

    networks:

      - my\_network

    env\_file:

      - mysqlconfig.env

networks:

  my\_network:

    driver: bridge

**#service main network specify kara**

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose up -d

[+] Running 4/4

✔ Network compose\_my\_network Created 0.1s

✔ Container compose-db-1 Started 2.5s

✔ Container compose-web-1 Started 2.4s

✔ Container compose-rcache-1 Started 2.3s

C:\Users\my pc\Desktop\docker geekyshows\compose>docker network ls

NETWORK ID NAME DRIVER SCOPE

e9799a33f543 bridge bridge local

ecb3b979c7a4 compose\_my\_network bridge local

8a6c0d7761dc host host local

0bb59ab32b49 net1 bridge local

b08384fa2bb6 none null local

# toh custome network hamen create akra hai , custom bridge network hamene create kara

**$ depends upon service pahle ye execute ho baad main ye ,service depend karta hia aapas main**

services:

  web:

    image: "nginx"

    ports:

      - "8000:80"

    networks:

      - my\_network

    depends\_on:

      - db

      - rcache

  rcache:

    image: "redis:${TAG}"

    # profiles:

    #   - rediscache

    networks:

      - my\_network

  db:

    image: mysql

    # environment:

    #   - MYSQL\_ROOT\_PASSWORD=deep

    networks:

      - my\_network

    env\_file:

      - mysqlconfig.env

networks:

  my\_network:

    driver: bridge

$ toh web service db or rcahce pa depend hogi hamari , pahle db or rcahce banegi phir hamari web service create hote hai

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose up -d

[+] Running 4/4

✔ Network compose\_my\_network Created 0.1s

✔ Container compose-db-1 Started 3.3s

✔ Container compose-rcache-1 Started 3.2s

✔ Container compose-web-1 Started 49.5s

#pahel db phir rcahce uske baad web service create hua hamra

C:\Users\my pc\Desktop\docker geekyshows\compose>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

0afe758e3791 nginx "/docker-entrypoint.…" About a minute ago Up About a minute 0.0.0.0:8000->80/tcp compose-web-1

637ab0f829dd redis:6.0-alpine3.18 "docker-entrypoint.s…" About a minute ago Up About a minute 6379/tcp compose-rcache-1

093888bdb0f1 mysql "docker-entrypoint.s…" About a minute ago Up About a minute 3306/tcp, 33060/tcp compose-db-1

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose down --volumes

[+] Running 4/4

✔ Container compose-web-1 Removed 0.9s

✔ Container compose-db-1 Removed 43.8s

✔ Container compose-rcache-1 Removed 43.7s

✔ Network compose\_my\_network Removed

Docker-compose-dev.yml – sabhi env ka liye alag file hogi , staging ka liye alag hogi , or prodn ka liye alag file hogi for both

services:

  web:

    image: "nginx"

    ports:

      - "8000:80"

    networks:

      - my\_network

    depends\_on:

      - db

      - rcache

  rcache:

    image: "redis:${TAG}"

    # profiles:

    #   - rediscache

    networks:

      - my\_network

  db:

    image: mysql

    # environment:

    #   - MYSQL\_ROOT\_PASSWORD=deep

    networks:

      - my\_network

    env\_file:

      - mysqlconfig.env

networks:

  my\_network:

    driver: bridge

docker-compose-dev.yml

services:

  web:

    image: "nginx"

  db:

    image: mysql

    # environment:

    #   - MYSQL\_ROOT\_PASSWORD=deep

Dev wali yml file hai hamari – docker-compose.dev.yml – isse run kasie karoge

services:

  web:

    image: "nginx"

  db:

    image: mysql

    env\_file:

      - mysqlconfig.env

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose -f docker-compose.dev.yml up –d

$ -f flag ka use karo , -f –f multiple file run kar sakte ho ,dev wali run kar paoge ,web or nignx cconater create hoge

C:\Users\my pc\Desktop\docker geekyshows\compose>docker compose -f docker-compose.dev.yml down

[+] Running 3/3

✔ Container compose-web-1 Removed 0.6s

✔ Container compose-db-1 Removed 2.4s

✔ Network compose\_default Removed

Compose 2

FROM alpine

WORKDIR /app

Dockerfile

services:

  web:

    image: "nginx"

  db:

    image: mysql

    environment:

      - MYSQL\_ROOT\_PASSWORD=deep

**Docker compose main docker file use karenge toh build ka use karenge , Build likhenge**

services:

  app:

    build:

      dockerfile: Dockerfile

  web:

    image: "nginx"

  db:

    image: mysql

    environment:

      - MYSQL\_ROOT\_PASSWORD=deep

C:\Users\my pc\Desktop\docker geekyshows\compose>cd compose2

$docker file se docker compose ka container create kar rha hai

C:\Users\my pc\Desktop\docker geekyshows\compose\compose2>docker compose up -d

[+] Running 2/2

✔ web Pulled 2.6s

✔ db Pulled 2.6s

[+] Running 0/15s (4/6) docker:desktop-linux

- Service app Building 2.7s

[+] Building 4.2s (8/8) FINISHED docker:desktop-linux

=> [app internal] load build definition from Dockerfile 0.3s

=> => transferring dockerfile: 70B 0.1s

=> [app internal] load metadata for docker.io/library/alpine:latest 0.1s

=> [app internal] load .dockerignore 0.1s

=> => transferring context: 2B 0.0s

=> [app 1/2] FROM docker.io/library/alpine:latest@sha256:56fa17d2a7e7f168a043a2712e63aed1f8543aeafdcee47c58dcffe38ed51099 2.3s

=> => resolve docker.io/library/alpine:latest@sha256:56fa17d2a7e7f168a043a2712e63aed1f8543aeafdcee47c58dcffe38ed51099 2.2s

=> [app auth] library/alpine:pull token for registry-1.docker.io 0.0s

=> [app 2/2] WORKDIR /app 0.1s

=> [app] exporting to image 0.8s

=> => exporting layers 0.4s

=> => exporting manifest sha256:ea1253d796a4394dc464a081d061913072f9fabf1f50c18b084ddbe6247b62fb 0.0s

=> => exporting config sha256:481fb288d0311928f567ca91c03f25f837ae414b0b3aebd0940ca8224364e7aa 0.0s

=> => exporting attestation manifest sha256:d348be83253daff8dc375ec72b5ed263c9c63e243e945eaa895398f007595e0e 0.1s

=> => exporting manifest list sha256:2c4a75d41efec51890c64a8b50b4e3db1923e2f836135980a7298c69ebb39e71 0.0s

=> => naming to docker.io/library/compose2-app:latest 0.0s

[+] Running 5/5g to docker.io/library/compose2-app:latest 0.1s

✔ Service app Built 4.8s

✔ Network compose2\_default Created 0.1s

✔ Container compose2-db-1 Started 61.7s

✔ Container compose2-web-1 Started 61.7s

✔ Container compose2-app-1 Started

C:\Users\my pc\Desktop\docker geekyshows\compose\compose2>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

2a426d69c33d nginx "/docker-entrypoint.…" 2 minutes ago Up 2 minutes 80/tcp compose2-web-1

f41e27e50630 mysql "docker-entrypoint.s…" 2 minutes ago Up 2 minutes 3306/tcp, 33060/tcp compose2-db-1

8e291dff9e9e compose2-app "/bin/sh" 2 minutes ago Exited (0) 2 minutes ago compose2-app-1

C:\Users\my pc\Desktop\docker geekyshows\compose\compose2>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

compose2-app latest 2c4a75d41efe 2 minutes ago 12.2MB

nginx latest 91734281c0eb 23 hours ago 279MB

mysql latest d56d039139a7 2 weeks ago 1.09GB

alpine latest 56fa17d2a7e7 4 weeks ago 12.1MB

redis latest 93a8d83b707d 4 weeks ago 173MB

redis 6.0-alpine3.18 cf878970baae 15 months ago 40.4MB

C:\Users\my pc\Desktop\docker geekyshows\compose\compose2>docker compose exec db mysql db mysql -u root -p

C:\Users\my pc\Desktop\docker geekyshows\compose\compose2>docker compose exec db mysql -u root -p

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 9

Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> $ get shell of mysql

**DONE YHA TAKK**

REACT PROJECT DOCKERIZE

Microsoft Windows [Version 10.0.19045.5487]

(c) Microsoft Corporation. All rights reserved.

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project>npm createvite@latest

Unknown command: "createvite@latest"

To see a list of supported npm commands, run:

npm help

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project>npm create vite@latest

Need to install the following packages:

create-vite@6.2.0

Ok to proceed? (y) y

> npx

> create-vite

√ Project name: ... react-project

√ Select a framework: » React

√ Select a variant: » JavaScript

Scaffolding project in C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project...

Done. Now run:

cd react-project

npm install

npm run dev

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project>cd react-project

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>npm i

added 259 packages, and audited 260 packages in 3m

108 packages are looking for funding

run `npm fund` for details

3 moderate severity vulnerabilities

To address all issues (including breaking changes), run:

npm audit fix --force

Run `npm audit` for details.

VITE v6.1.0 ready in 917 ms

➜ Local: http://localhost:5173/

➜ Network: use --host to expose

➜ press h + enter to show help

8:38:30 pm [vite] (client) hmr update /src/App.jsx

8:38:30 pm [vite] (client) hmr update /src/App.jsx (x2)

$ Docker main cotnaienr banayenge isko dockerize karenge

$ Creating of image - to rrun react project ew need node iamge

# sabse pahle docker file – use docker image create karenge – phir use container banayenge

# Use the official Node. js image as the base image

FROM node : 18.17-slim AS development

# Set the working directory inside the container

WORKDIR /react-project

# Copy package. json to the container

COPY package. json .

# Install dependencies

RUN npm install

# Copying all the files in our project

COPY . .

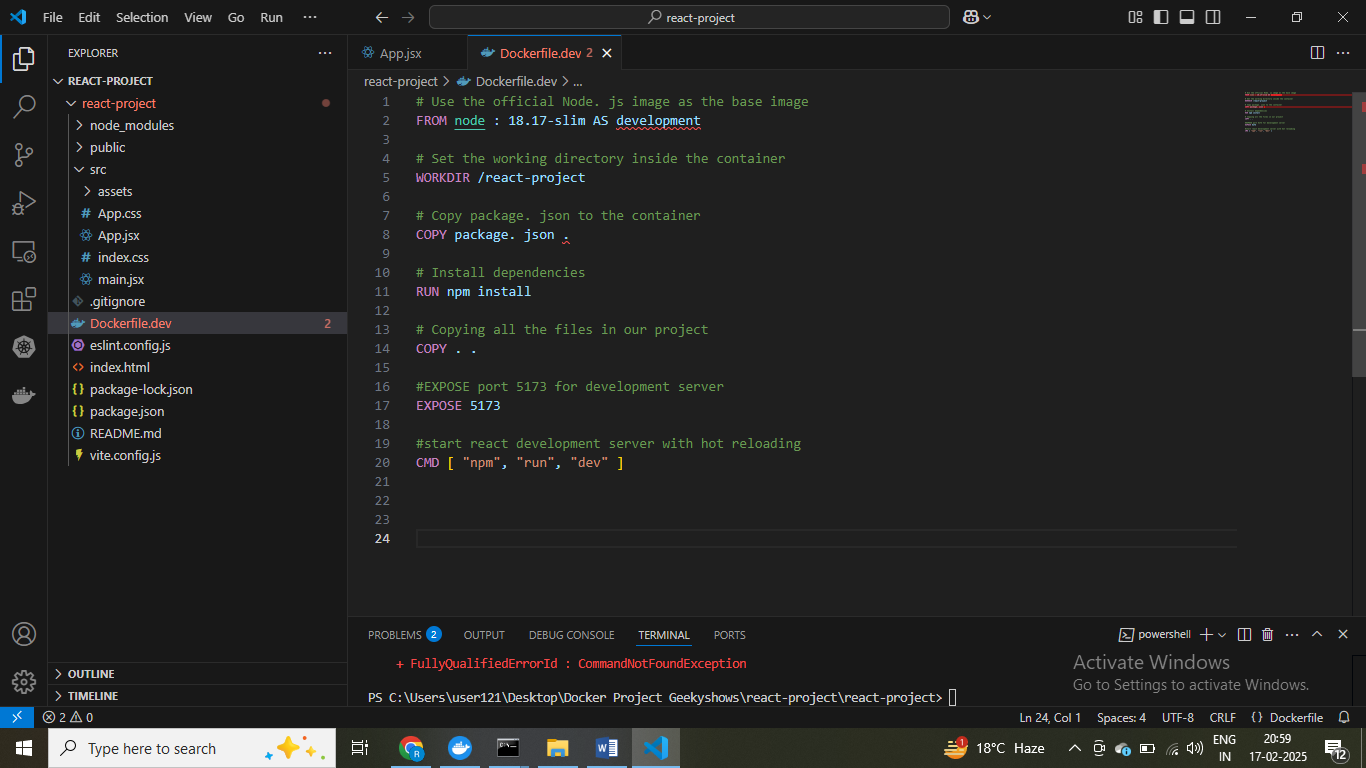
#EXPOSE port 5173 for development server

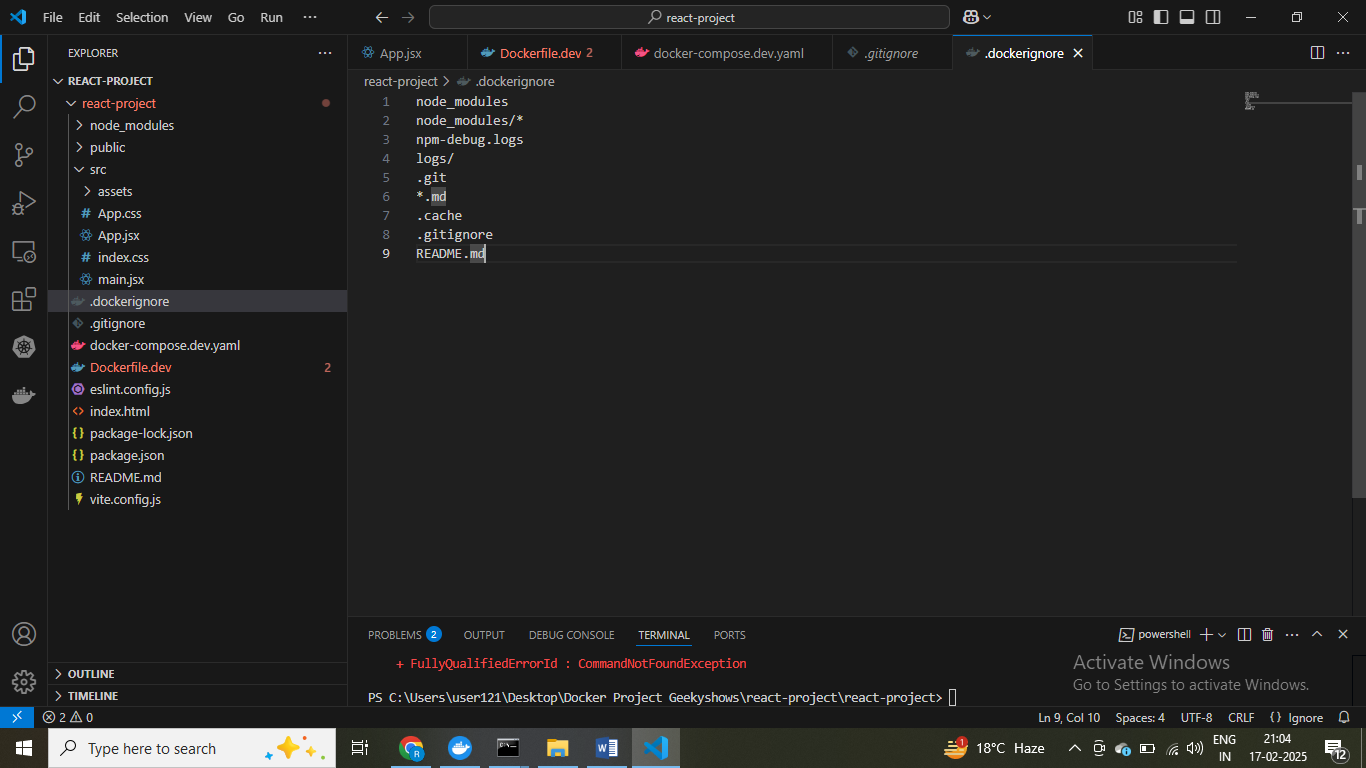
EXPOSE 5173

#start react development server with hot reloading

CMD [ "npm", "run", "dev" ]

Dockerfile





version: '3.8'

services:

  react-project-dev:

    container\_name: react-project-container

    image: react-project-dev-image

    build:

      context: .

      dockerfile: Dockerfile.dev

    ports:

      - "5173:5173"

    volumes:

      - .:/react-project

      - ///react-project/node\_modules  # Prevents overwriting node\_modules inside the container

    command: ["npm", "run", "dev"]

docker-compose-dev.yaml

# Use the official Node.js image as the base image

FROM node:18.17-slim AS development

# Set the working directory inside the container

WORKDIR /react-project

# Copy package.json and package-lock.json for dependency installation

COPY package.json package-lock.json ./

# Install dependencies

RUN npm install

# Copy all project files

COPY . .

# Expose port 5173 for the development server

EXPOSE 5173

# Start React development server with hot reloading

CMD ["npm", "run", "dev"]

Dockerfile.dev

[+] Running 1/11\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml up –d

! react-project-dev Warning pull access denied for react-project-dev-image, repository does not exist or may require 'docker login' 4.1s

[+] Building 149.7s (12/12) FINISHED

Error: Cannot find module @rollup/rollup-linux-x64-gnu. npm has a bug related to optional dependencies (https://github.com/npm/cli/issues/4828). Please try `npm i` again after removing both package-lock.json and node\_modules directory.

2025-02-17 21:20:36 react-project-container | at requireWithFriendlyError (/react-project/

$ geeting error container created but not running

services:

  react-project-dev:

    container\_name: react-project-container

    image: react-project-dev-image

    build:

      context: .

      dockerfile: Dockerfile.dev

    ports:

      - "5173:5173"

    volumes:

      - .:/react-project

      - /react-project/node\_modules  # Prevents overwriting node\_modules inside the container

    command: ["npm", "run", "dev"]

$ Isolate node modules folder to prevernt overwrite , window ka node module folder utka jarahai

$ Delete container and then restart it again

$ Image network container create ho jayega ek sath sab

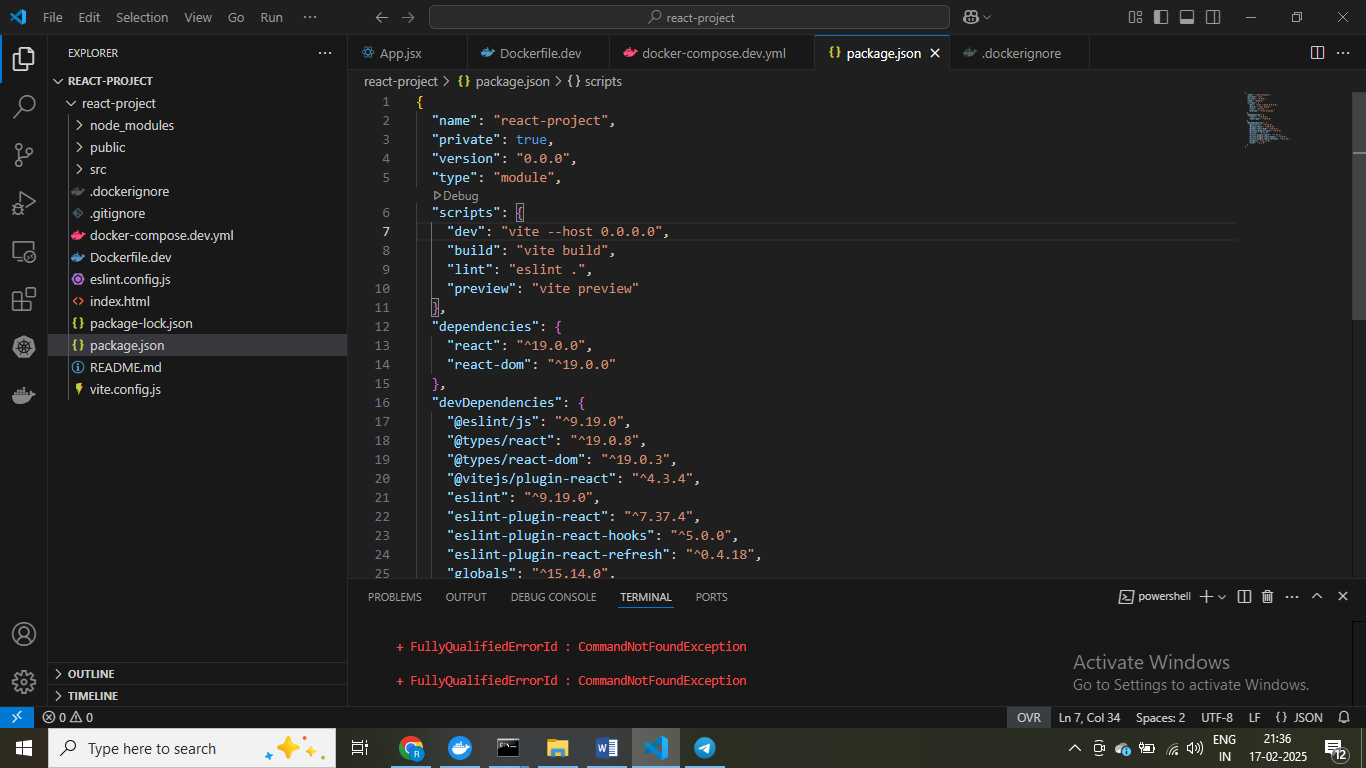
$ fist is host prot then contienr poert use

[+] Running 1/11\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yaml up -d

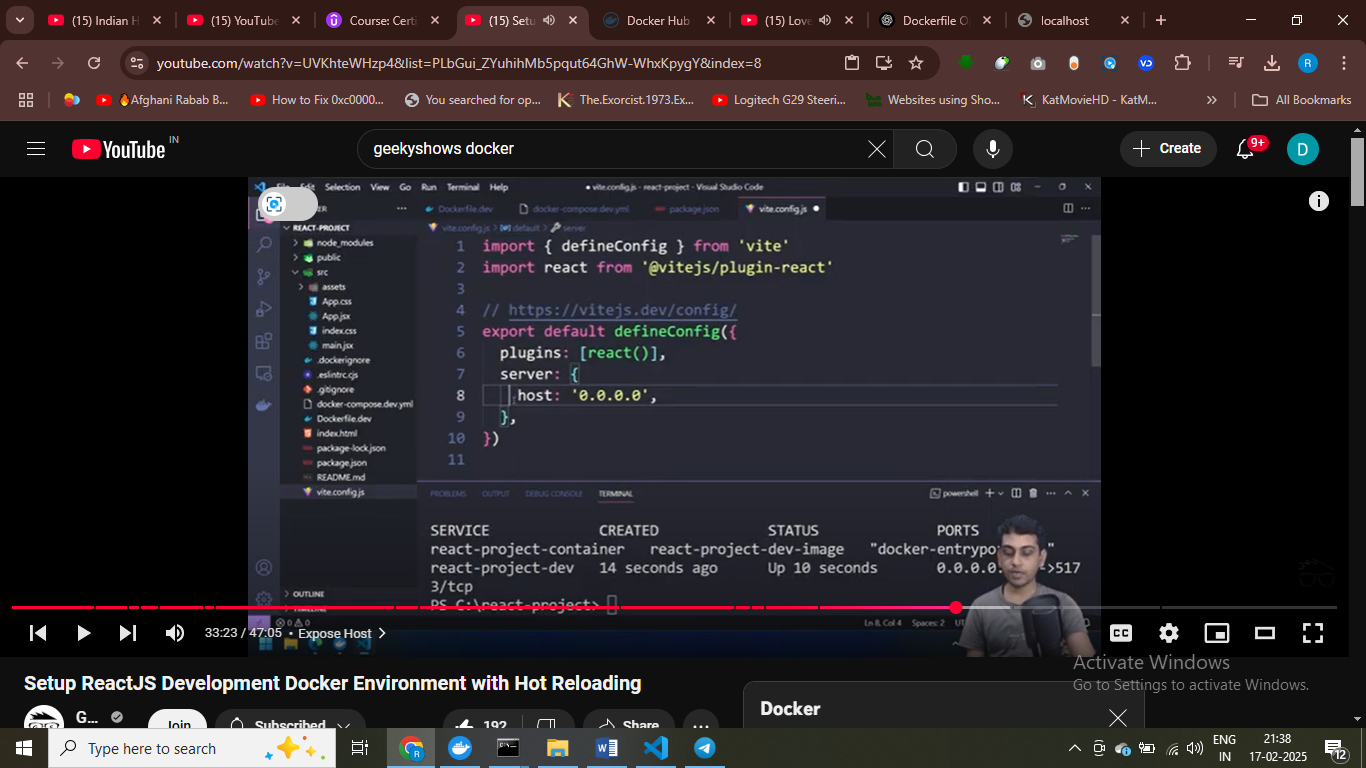
! react-project-dev Warning pull access denied for react-project-dev-image, repository does not exist or may require 'docker login'

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml ps

NAME IMAGE COMMAND SERVICE CREATED STATUS



$exposeing host port , paremmetn change hoge port expose hoge



[+] Stopping 1/1\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml ps

✔ Container react-project-container Stopped

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml up -d

[+] Running 1/1

✔ Container react-project-container Started 0.7s

$ container pa run kar pa rahe proect hai apne

$ Stop and then restart the container

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml stop

[+] Stopping 1/1

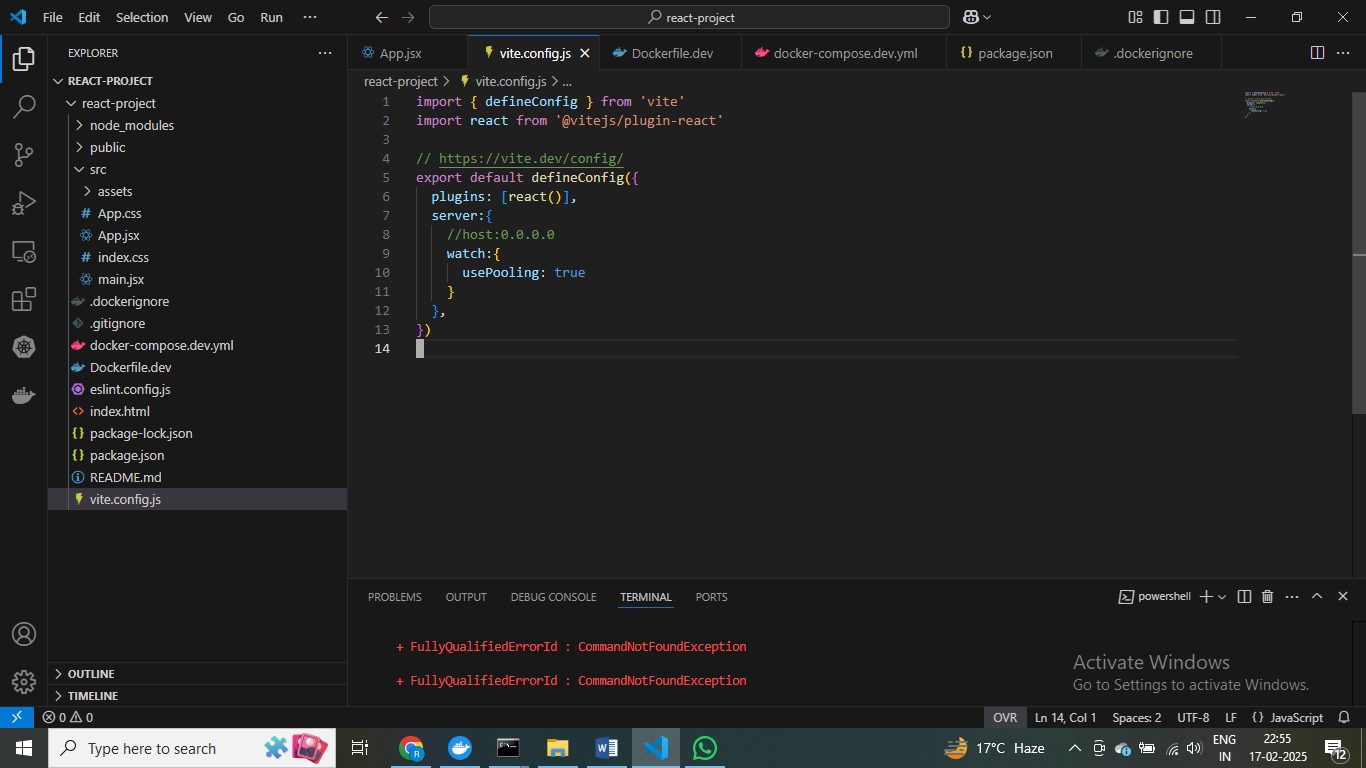
✔ Container react-project-container Stopped 10.2s

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml start

[+] Running 1/1

✔ Container react-project-container Started 0.6s

$ Hot REALOADING NOT WORKING



$ HOT realoading enabled

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml stop

[+] Stopping 1/1

✔ Container react-project-container Stopped 11.5s

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml start

[+] Running 1/1

✔ Container react-project-container Started

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml stop

[+] Stopping 1/1

✔ Container react-project-container Stopped 10.3s

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml up -d

[+] Running 1/1

✔ Container react-project-container Started

**% image ko dcoekr hub pa share kar do aur koi bhi is project ko run kar payge**

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml ps

NAME IMAGE COMMAND SERVICE CREATED STATUS PORTS

react-project-container react-project-dev-image "docker-entrypoint.s…" react-project-dev 2 hours ago Up About a minute 0.0.0.0:5173->5173/tcp

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml exec react-project-dev bash

root@349c6c7cfdaa:/react-project# ls

Dockerfile.dev README.md docker-compose.dev.yml eslint.config.js index.html node\_modules package-lock.json package.json public src vite.config.js

root@349c6c7cfdaa:/react-project# cd node\_modules/

root@349c6c7cfdaa:/react-project/node\_modules# cd @esbuild

root@349c6c7cfdaa:/react-project/node\_modules/@esbuild# ls

linux-x64

root@349c6c7cfdaa:/react-project/node\_modules/@esbuild# cd ..

root@349c6c7cfdaa:/react-project/node\_modules# cd ..

root@349c6c7cfdaa:/react-project# ls -a

. .dockerignore Dockerfile.dev docker-compose.dev.yml index.html package-lock.json public vite.config.js

.. .gitignore README.md eslint.config.js node\_modules package.json src

root@349c6c7cfdaa:/react-project#

$ compose file main sabkuch bind kar diya isliye node modules folder yaha pa aajega

root@349c6c7cfdaa:/react-project# exit

exit

$ dev main nginx ya apache server use kareta hai , isko use nahi karte ham

$ dev pupose main aise use hoge , dev ka km host machine pa aur run karege host machine paa

$ container pa run ho rha project

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml stop

[+] Stopping 1/1

✔ Container react-project-container Stopped 10.5s

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml start

[+] Running 1/1

✔ Container react-project-container Started 0.5s

C:\Users\user121\Desktop\Docker Project Geekyshows\react-project\react-project>docker compose -f docker-compose.dev.yml stop

# DONE

DJANGO project using docker containers

<https://github.com/geekyshow1/dockerdjangosetup.git>

$ left side main host ka port aur right side main container ka port hoge hamraa = remberthis

PS C:\Users\user121\Desktop\Docker Project Geekyshows\Docker\dockerdjangosetup-main\dockerdjangosetup-main> docker compose -f docker-compose.dev.yml up -d

env file C:\Users\user121\Desktop\Docker Project Geekyshows\Docker\dockerdjangosetup-main\dockerdjangosetup-main\.env not found: CreateFile C:\Users\user121\Desktop\D

PS C:\Users\user121\Desktop\Docker Project Geekyshows\Docker\dockerdjangosetup-main\dockerdjangosetup-main> docker compose -f docker-compose.dev.yml up -d

[+] Running 33/33

✔ phpmyadmin Pulled

PS C:\Users\user121\Desktop\Docker Project Geekyshows\Docker\dockerdjangosetup-main\dockerdjangosetup-main> docker compose -f docker-compose.dev.yml stop

[+] Stopping 3/3

✔ Container phpmyadmin\_con Stopped 1.7s

✔ Container django\_con Stopped 0.0s

✔ Container mysql\_con Stopped 4.0s

PS C:\Users\user121\Desktop\Docker Project Geekyshows\Docker\dockerdjangosetup-main\dockerdjangosetup-main> docker compose -f docker-compose.dev.yml start

[+] Running 3/3

✔ Container mysql\_con Healthy

PS C:\Users\user121\Desktop\Docker Project Geekyshows\Docker\dockerdjangosetup-main\dockerdjangosetup-main> docker compose -f docker-compose.dev.yml ps

NAME IMAGE COMMAND SERVICE CREATED STATUS PORTS

<http://127.0.0.1:8000/> - proje is running successfully on this port

PS C:\Users\user121\Desktop\Docker Project Geekyshows\Docker\dockerdjangosetup-main\dockerdjangosetup-main> docker compose -f docker-compose.dev.yml exec django\_proje -f docker-compose.dev.yml exec django\_project bash

root@c6a7faf75f02:/dockerdjangosetup# ls

Dockerfile.dev blog docker-compose.dev.yml dockerdjangosetup manage.py mysql\_data requirements.txt

root@c6a7faf75f02:/dockerdjangosetup# ls

Dockerfile.dev blog demo1 docker-compose.dev.yml dockerdjangosetup manage.py mysql\_data requirements.txt

PS C:\Users\user121\Desktop\Docker Project Geekyshows\Docker\dockerdjangosetup-main\dockerdjangosetup-main> docker compose -f docker-compose.dev.yml exec django\_proje -f docker-compose.dev.yml exec django\_project bash

root@c6a7faf75f02:/dockerdjangosetup# ls

Dockerfile.dev blog docker-compose.dev.yml dockerdjangosetup manage.py mysql\_data requirements.txt

root@c6a7faf75f02:/dockerdjangosetup# ls

Dockerfile.dev blog demo1 docker-compose.dev.yml dockerdjangosetup manage.py mysql\_data requirements.txt

root@c6a7faf75f02:/dockerdjangosetup# python manage.py createsuperuser

Username (leave blank to use 'root'): admin

Email address: dk1078451@gmail.com

Password:

Password (again):

This password is too short. It must contain at least 8 characters.

Bypass password validation and create user anyway? [y/N]: n

Password:

Password (again):

Superuser created successfully.

root@c6a7faf75f02:/dockerdjangosetup#

PS C:\Users\user121\Desktop\Docker Project Geekyshows\Docker\dockerdjangosetup-main\dockerdjangosetup-main> docker compose -f docker-compose.dev.yml stop

[+] Stopping 3/3

✔ Container phpmyadmin\_con Stopped 3.2s

✔ Container django\_con Stopped 1.6s

✔ Container mysql\_con Stopped 15.0s

PS C:\Users\user121\Desktop\Docker Project Geekyshows\Docker\dockerdjangosetup-main\dockerdjangosetup-main> docker compose -f docker-compose.dev.yml start

[+] Running 1/2

- Container mysql\_con Waiting 44.6s

✔ Container phpmyadmin\_con Started

$ project run horha hai conteinr main , docker container main runho rha ha