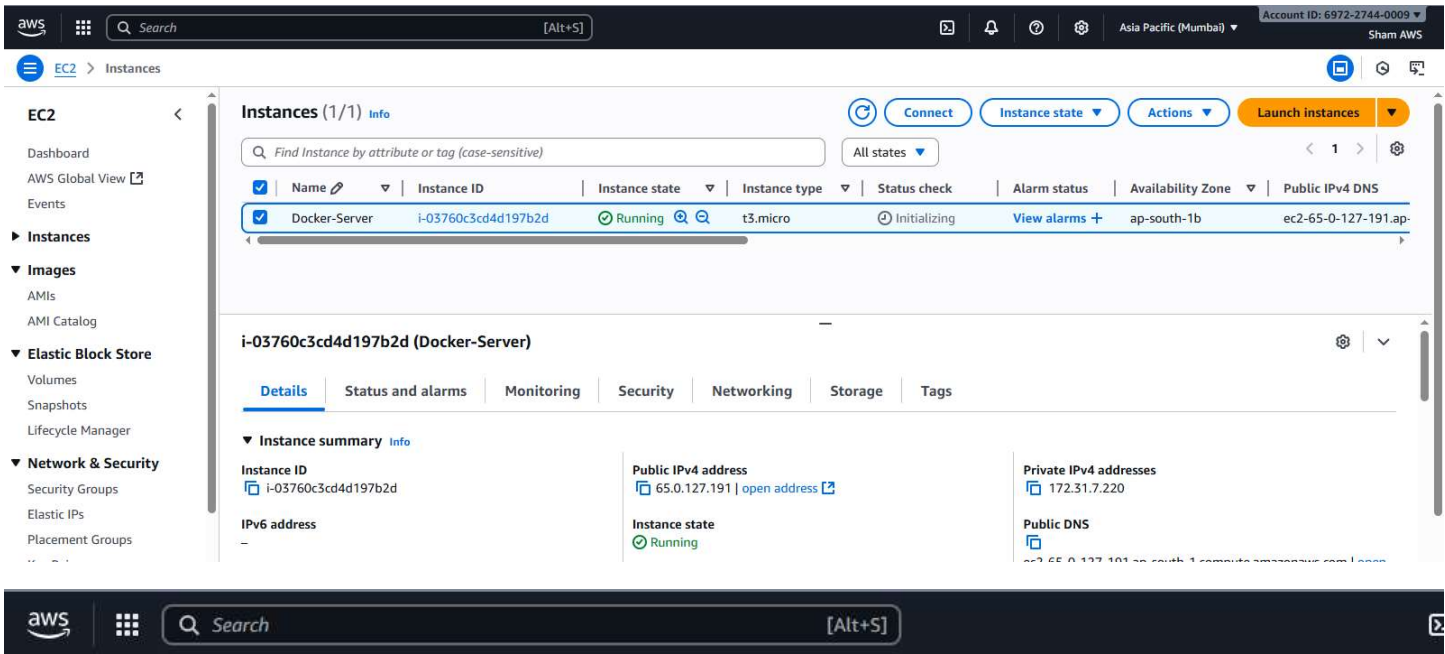


# Build and Deploy a Custom Docker Image using Docker Compose

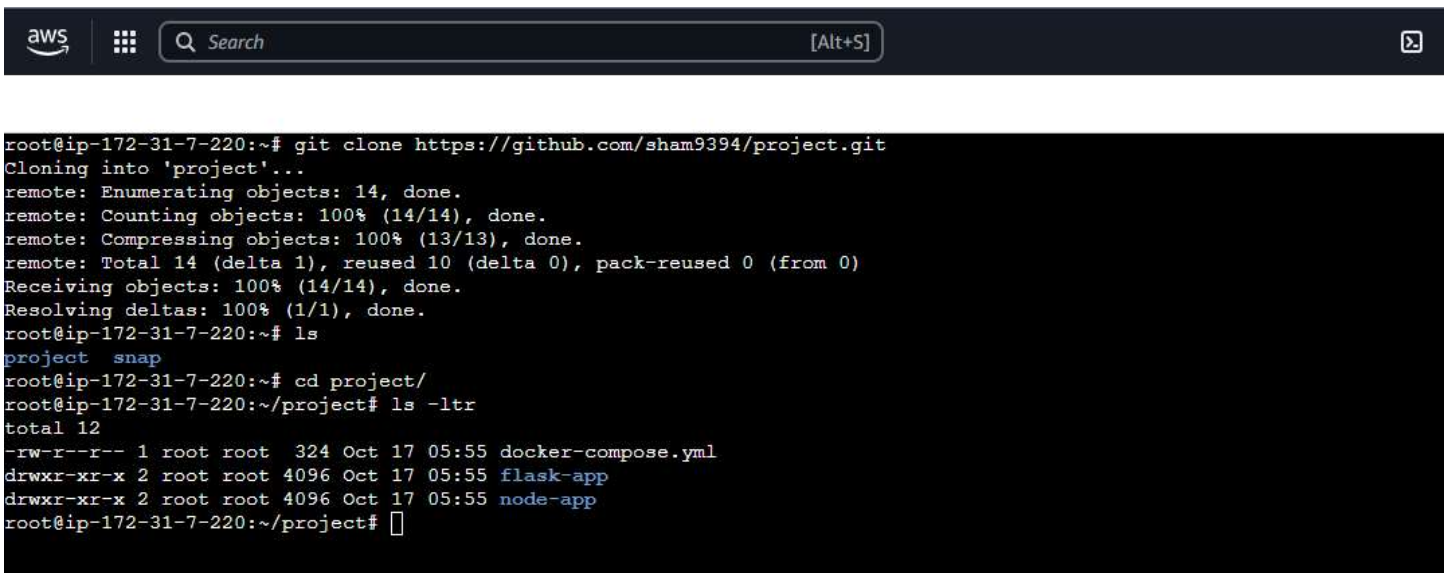
## Scenario 1: Build Image and Run Container using Docker Compose.

### 1. Setup EC2 Server with Configured Docker and Docker Compose.



```
root@ip-172-31-7-220:~# docker --version
Docker version 28.2.2, build 28.2.2-0ubuntu1~24.04.1
root@ip-172-31-7-220:~# docker compose version
Docker Compose version v2.40.0
root@ip-172-31-7-220:~#
```

### 2. Pull the Code on Server and Write a Docker-Compose.yml File.



```

root@ip-172-31-7-220:~/project# vi docker-compose.yml
root@ip-172-31-7-220:~/project# cat docker-compose.yml

services:
  flask-app:
    image: flask-app
    build: ./flask-app
    container_name: flask-container
    ports:
      - "5000:5000"
    restart: always

  node-app:
    image: node-app
    build: ./node-app
    container_name: node-container
    ports:
      - "3000:3000"
    restart: always
root@ip-172-31-7-220:~/project# 

```

### 3. Run Container using Docker-compose.yml.

```

root@ip-172-31-7-220:~/project# docker compose up -d
[*] Running 2/2
! flask-app Warning pull access denied for flask-app, repository does not exist or may require 'docker login': denied: requested access to the reso...
! node-app Warning pull access denied for node-app, repository does not exist or may require 'docker login': denied: requested access to the resour...
WARN[0002] Docker Compose is configured to build using Bake, but buildx isn't installed
[*] Building 28.5s (21/21) FINISHED
=> [flask-app internal] load build definition from Dockerfile
=> => transferring dockerfile: 368B
=> [node-app internal] load build definition from Dockerfile
=> => transferring dockerfile: 154B
=> [node-app internal] load metadata for docker.io/library/node:18-alpine
=> [flask-app internal] load metadata for docker.io/library/python:3.11-slim
=> [flask-app internal] load .dockerignore
=> => transferring context: 2B
=> [flask-app 1/4] FROM docker.io/library/python:3.11-slim@sha256:ff8533f48e12b705fc20d339fde2ec61d0b234dd9366bab3bc84d7b70a45c8c0
=> resolve docker.io/library/python:3.11-slim@sha256:ff8533f48e12b705fc20d339fde2ec61d0b234dd9366bab3bc84d7b70a45c8c0
=> sha256:7bbe597de5c76e70498898003b3e0402a6f4ef23b0ba30d33acd8d1ef863f128 5.38kB / 5.38kB
=> sha256:8c7716127147648c1751940b9709b6325f2256290d3201662eca2701cadb2cdf 29.78MB / 29.78MB
=> sha256:c72c567266265eaf3c81cecf291e32dc35cb03f44a34cc37c4bb2c3f1ca6741c 4.25MB / 4.25MB
=> sha256:76d93c681ade9d7ff7e4e590094f416d05d02ce51cb023dbb97acd48c3073470 14.36MB / 14.36MB
=> sha256:ff8533f48e12b705fc20d339fde2ec61d0b234dd9366bab3bc84d7b70a45c8c0 10.37kB / 10.37kB
=> sha256:6818dccc897e2708b24af119860e4d678d9d40a725264beb20d4988deef1ccfe 1.75kB / 1.75kB
=> sha256:80061c640d6316e0810fd1007261f33680529077a173449bba4b55579c66db45 250B / 250B
=> extracting sha256:8c7716127147648c1751940b9709b6325f2256290d3201662eca2701cadb2cdf
=> extracting sha256:c72c567266265eaf3c81cecf291e32dc35cb03f44a34cc37c4bb2c3f1ca6741c
=> extracting sha256:76d93c681ade9d7ff7e4e590094f416d05d02ce51cb023dbb97acd48c3073470
=> extracting sha256:80061c640d6316e0810fd1007261f33680529077a173449bba4b55579c66db45
=> [flask-app internal] load build context
=> => transferring context: 1.57kB
=> [node-app internal] load .dockerignore
=> => transferring context: 2B
=> [node-app 1/5] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e

```

```

=> => naming to docker.io/library/node-app
=> [node-app] resolving provenance for metadata file
=> [flask-app] exporting to image
=> => exporting layers
=> => writing image sha256:4bec8bcfab037b1af77b421d71314904f30a6e6d344aee787eb2ef6b887d914b
=> => naming to docker.io/library/flask-app
=> [flask-app] resolving provenance for metadata file
[*] Running 5/5
✓ flask-app Built
✓ node-app Built
✓ Network project default Created
✓ Container node-container Started
✓ Container flask-container Started
root@ip-172-31-7-220:~/project# 

```

## 4. Verify Server Output using IP.

```
aws | Search [Alt+S] minimize Asia Pacific (Mu)

root@ip-172-31-7-220:~/project# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
flask-app latest 4bec8bcfab03 4 minutes ago 335MB
node-app latest db3ab7ddb00 4 minutes ago 139MB
root@ip-172-31-7-220:~/project# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
f459e0fef114 flask-app "python app.py" 4 minutes ago Up 4 minutes 0.0.0.0:5000->5000/tcp, [::]:5000->5000/tcp flask-container
5195b2c35c0f node-app "docker-entrypoint.s..." 4 minutes ago Up 4 minutes 0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp node-container
root@ip-172-31-7-220:~/project#
```



# Welcome to Flask App on EC2!

This app is running directly on EC2 without Docker.

- [/fetch - External API Call](#)
- [/health - Health Check](#)



# Welcome to My Node.js App!

Click below to get a random programming quote:

[Get Quote](#)

```
aws | Search [Alt+S]

root@ip-172-31-7-220:~/project# docker compose down
[+] Running 3/3
✓ Container flask-container Removed
✓ Container node-container Removed
✓ Network project default Removed
root@ip-172-31-7-220:~/project#
```

## Scenario 2: Build Image and Run Container using Docker Compose (Dependency Case)

### 5. Reconfigure Docker Compose file for Depend on case. (flask-app will run after node-app will start successfully)

```
aws | Search [Alt+S]

root@ip-172-31-7-220:~/project# vi docker-compose.yml
root@ip-172-31-7-220:~/project# cat docker-compose.yml

services:

  node-app:
    image: node-app
    build: ./node-app
    container_name: node-container
    ports:
      - "3000:3000"
    restart: always

  flask-app:
    image: flask-app
    build: ./flask-app
    container_name: flask-container
    ports:
      - "5000:5000"
    depends_on:
      - node-app
    restart: always
root@ip-172-31-7-220:~/project#
```

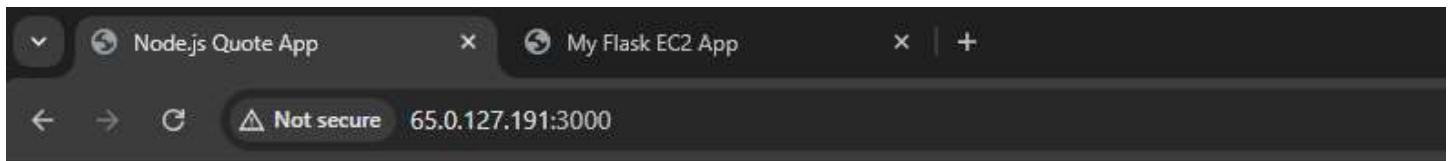
```
aws | Search [Alt+S] | Asia Pacific (Mumbai)

root@ip-172-31-7-220:~/project# docker compose up -d
[+] Running 2/2
 ! flask-app Warning pull access denied for flask-app, repository does not exist or may require 'docker login': denied: requested access to the reso...
 ! node-app Warning pull access denied for node-app, repository does not exist or may require 'docker login': denied: requested access to the resour...
WARN[0002] Docker Compose is configured to build using Bake, but buildx isn't installed
[+] Building 1.5s (21/21) FINISHED
=> [node-app internal] load build definition from Dockerfile
=> => transferring dockerfile: 154B
=> [node-app internal] load metadata for docker.io/library/node:18-alpine
=> [node-app internal] load .dockerignore
=> => transferring context: 2B
=> [node-app 1/5] FROM docker.io/library/node:18-alpine@sha256:8d6421d663b4c28fd3ebc498332f249011d118945588d0a35cb9bc4b8ca09d9e
=> [node-app internal] load build context
=> => transferring context: 89B
=> CACHED [node-app 2/5] WORKDIR /app
=> CACHED [node-app 3/5] COPY package.json ./
=> CACHED [node-app 4/5] RUN npm install
=> CACHED [node-app 5/5] COPY . .
=> [node-app] exporting to image
=> => exporting layers
=> => writing image sha256:db3ab7ddbba0ea81c4c4a9a4051c93354a122b6b51c80abab474a51e49c4ad9
=> => naming to docker.io/library/node-app
=> [node-app] resolving provenance for metadata file
=> [flask-app internal] load build definition from Dockerfile
=> => transferring dockerfile: 368B
=> [flask-app internal] load metadata for docker.io/library/python:3.11-slim
=> [flask-app internal] load .dockerignore
=> => transferring context: 2B
=> [flask-app 1/4] FROM docker.io/library/python:3.11-slim@sha256:ff8533f48e12b705fc20d339fde2ec61d0b234dd9366bab3bc84d7b70a45c8c0
=> [flask-app internal] load build context
=> => transferring context: 93B
=> CACHED [flask-app 2/4] WORKDIR /app

=> => exporting layers
=> => writing image sha256:4bec8bcfab037b1af77b421d71314904f30a6e6d344aee787eb2ef6b887d914b
=> => naming to docker.io/library/flask-app
=> [flask-app] resolving provenance for metadata file
[+] Running 5/5
✓ flask-app Built
✓ node-app Built
✓ Network project_default Created
✓ Container node-container Started 1
✓ Container flask-container Started 2
root@ip-172-31-7-220:~/project#
```



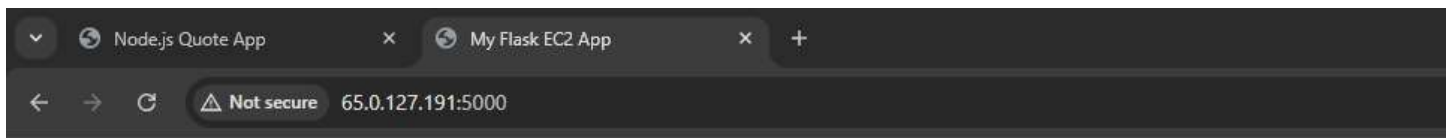
## 6. Verify Server Output using IP.



# Welcome to My Node.js App!

Click below to get a random programming quote:

[Get Quote](#)



# Welcome to Flask App on EC2!

This app is running directly on EC2 without Docker.

- [/fetch - External API Call](#)
- [/health - Health Check](#)



```
root@ip-172-31-7-220:~/project# docker compose down
[+] Running 3/3
✓ Container flask-container   Removed
✓ Container node-container    Removed
✓ Network project_default     Removed
root@ip-172-31-7-220:~/project#
```