

Proof of Work – Dockerized Node.js + AWS ECR + Docker Compose

Screenshots and short captions demonstrating each stage of the project.

1) SSH into the Ubuntu EC2 instance — verifying remote access.

```
adnanahmed@Adnans-MacBook-Air.local ~/Users/adnanahmed
% ssh -i ~/.ssh/docker-key.pem ec2-user@13.40.139.211
ec2-user@13.40.139.211: Permission denied (publickey).
adnanahmed@Adnans-MacBook-Air.local ~/Users/adnanahmed
% ssh -i ~/.ssh/docker-key.pem ubuntu@13.40.139.211
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.8.0-1035-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed Sep  3 18:23:33 UTC 2025

System load:  0.0               Processes:    119
Usage of /:   52.4% of 7.57GB   Users logged in:  0
Memory usage: 47%              IPv4 address for ens5: 172.31.13.127
Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

11 updates can be applied immediately.
11 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

9 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '24.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Sep  3 16:07:17 2025 from 81.157.70.199
ubuntu@ip-172-31-13-127:~$
```

2) Docker & Docker Compose installed — versions confirmed.

```
ubuntu@ip-172-31-13-127:~$ docker --version
docker compose version
Docker version 28.3.3, build 980b856
Docker Compose version v2.39.1
```

3) Project directory on server — compose + app files in place.

```
ubuntu@ip-172-31-13-127:~$ cd ~/app
ls -l
total 12
drwxrwxr-x 7 ubuntu docker 4096 Sep  3 17:02 devops-journey
-rw-rw-r-- 1 ubuntu docker  763 Sep  3 17:29 docker-compose.yml
-rw-rw-r-- 1 ubuntu docker  770 Sep  3 16:57 docker-compose.yml.bak
```

4) *docker compose up / logs — stack brought up successfully.*

```
awscli@ip-172-31-43-127:~/app$ docker compose ps
```

NAME	IMAGE	COMMAND	SERVICE	CREATED	STATUS	PORTS
exp-app-1	3514933873335.dkr.ecr.eu-west-2.amazonaws.com/my-node-app:1.0-and64	"docker-entrypoint.s."	app	56 minutes ago	Up 56 minutes	0.0.0.0:3000->3000/tcp, [::]:3000->3000/tcp
app-mongo-express-1	mongo-express:1.0.0-alpha.4	"tm -- /docker-ent."	mongo-express	56 minutes ago	Up 55 minutes	0.0.0.0:8081->8081/tcp, [::]:8081->8081/tcp
mongo	mongo:6.0.6	"docker-entrypoint.s."	mongo	56 minutes ago	Up 56 minutes	27017/tcp

5) *docker compose ps* — *services running (app, mongo, mongo-express).*

```
mongo: mongo.0.0.0
ubuntu@ip-172-31-13-127:~/app$ docker logs $(docker compose ps -q app)

> my-app@1.0.0 start
> node index.js

Server listening on port 3000
```

6) *Browser result — Hello from Dockerized Node.js* ■

Hello from Dockerized Node.js 🚀