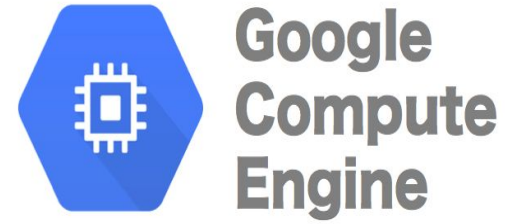
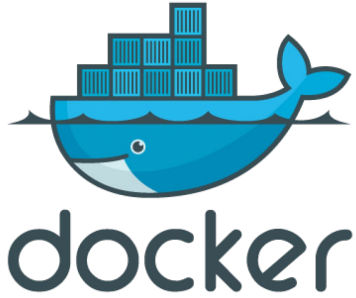


MEAN Stack Application Deployment

TECHNOLOGIES USED



STEP 1: BUILDING THE DOCKER IMAGES

```
1 FROM node:14 as node
2
3 WORKDIR /app
4
5 COPY package*.json ./
6
7 RUN npm ci
8
9 COPY . ./
10
11 RUN npm run build
12
13 FROM nginx:1.20.1-alpine
14
15 COPY --from=node public/ /usr/share/nginx/html/
```



Dockerfile for Angular App

```
1 FROM node:14 as builder
2
3 ENV NODE_ENV=production
4
5 WORKDIR /app
6
7 COPY package*.json ./
8
9 RUN npm ci --only=production
10
11 COPY . .
12
13 FROM node:14-alpine
14
15 WORKDIR /app
16
17 COPY --from=builder /app .
18
19 EXPOSE 8080
20
21 CMD [ "node", "server.js" ]
```



Dockerfile for Node.js

STEP 2: CREATING EC2 INSTANCE FOR JENKINS

Instance ID	Instance state ▾	Instance type ▾	Status check
i-055ae6d10e243f0da	✓ Running 🔍	t2.micro	✓ 2/2 checks passed



Welcome to Jenkins!

Sign in

☐ Keep me signed in

Started an EC2 instance configured to run multiple docker containers for jenkins and it's slaves.

STEP 3: SLACK INTEGRATION WITH JENKINS



nader.tarek.elsherbin 4:03 PM

joined #deployment.



nader.tarek.elsherbin 4:09 PM

added an integration to this channel: [jenkins](#)

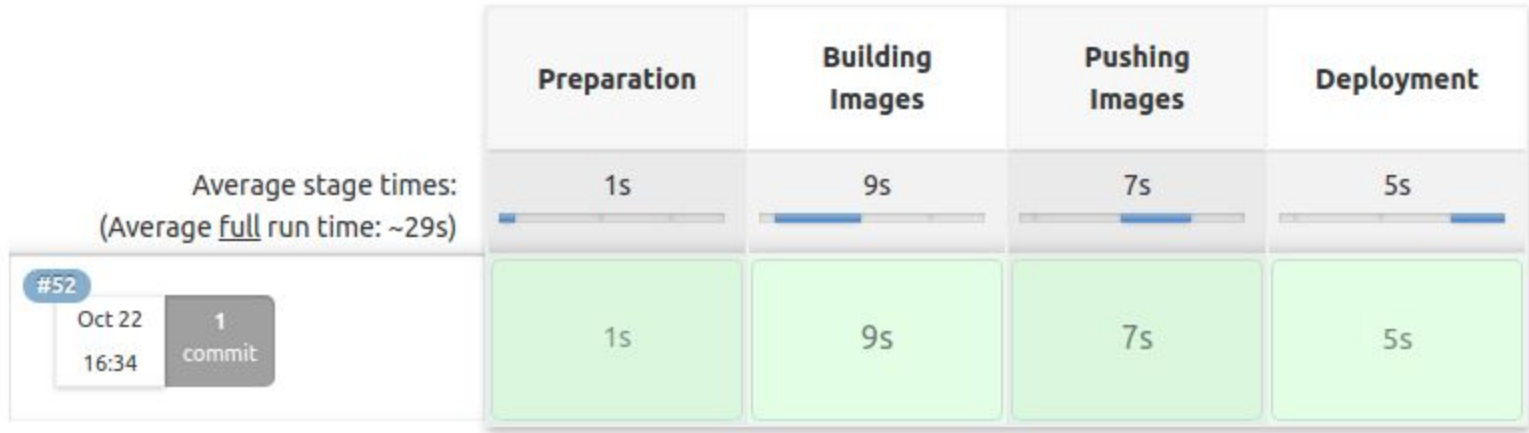


jenkins APP 4:11 PM

Slack/Jenkins plugin: you're all set on <http://3.15.23.154:8080/>

STEP 4: BUILT CI/CD PIPELINE

Stage View



STEP 5: ADDED GITHUB WEBHOOKS TO TRIGGER THE PIPELINE AND THE RESULTS ARE SENT AS NOTIFICATIONS TO A SLACK CHANNEL

Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

✓ <http://3.15.23.154:8080/github-w...> (push)

Edit

Delete



jenkins APP 4:22 PM

Building Images has Failed



jenkins APP 4:27 PM

Application has been Deployed Successfully

STEP 6: CREATED VM INSTANCE FOR THE PRODUCTION ENVIRONMENT

<input type="checkbox"/>	Status	Name ↑	Zone	Recommendations	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	✓	docker-containers	eu-west-4a			10.164.0.4 (nic0)	34.90.135.151 ↗	SSH ▾ ⋮

Configured a VM instance using google compute engine to spin up containers for the application to run.

STEP 7: DOCKER-COMPOSE FILE

```
1  version: "3"
2  services:
3    angular-app:
4      image: nadertarekcs/angular-service
5      stdin_open: true
6      ports:
7        - "80:80"
8      networks:
9        - mean-app
10   api-server:
11     image: nadertarekcs/back-end-service
12     ports:
13       - "5000:8080"
14     networks:
15       - mean-app
16 networks:
17   mean-app:
18     driver: bridge
```

This file will be used in the deployment stage in the CI/CD pipeline to spin up the containers and the database is hosted on mongoDB-Atlas.

STEP 8: HERE IS THE LIVE APPLICATION

