

NODE.JS

CI/CD

PIPELINES

Mohammad-Ali A'râbi
Docker All-Hands #5



TABLE OF CONTENTS

01

INTRO

We're going to cook
a Node.js exec
Docker image

02

INGREDIENTS

Node.js, Docker,
and 2 teaspoons
database mocks

03

BUILD

Let the TS dough
rest and ferment
into JS

04

TEST

Test away: the
dough, the cook,
the end result

05

PUBLISH

It's pu(bli)sh:
You publish by
pushing

01

INTRODUCTION

FROM `node:17-alpine`
LABEL maintainer "aerabi"





MOHAMMAD-ALI A'RÂBI

Software Engineer, Mathematician,
Iranian, based in Germany:

- Writing code since 2004
- Docker user since 2015
- Docker Community Leader of
Switzerland

Tweet me [@MohammadAliEN](#)

GitHub, Medium, LinkedIn [@aerabi](#)

BUILD NODE.JS FOR PRODUCTION



NODE.JS

A backend
project written
in TypeScript



DOCKER

An executable
Node.js backend
application
Docker image



02

INGREDIENTS

Node.js, GitLab-CI,
Gorgonzola, and Pesto

TECHNOLOGIES

1. Node.js
2. TypeScript
3. Express.js (a web framework)
4. MongoDB (a database)
5. Jest/Jasmine (a testing framework)
6. Docker-Compose
7. GitLab-CI

03

BUILD

Trim your beard while
the project is building



BUILD PHASES



PRE-TRANSPILE

You want to rebuild the routes?



TRANSPILE

Make it JS, minify, uglify, binify



DOCKER IMAGE

Put minified uglified binified into Alpine

BUILD STEPS

Create package.json commands:

- build:pre
- build:transpile
- build:prod
- build:docker

BUILD COMMANDS

package.json

```
{
  "scripts": {
    "build:pre": "npm run build:routes && npm run build:swagger",
    "build:transpile": "tsc -p src",
    "build:prod": "npm run build:webpack && npm run build:pkg",
    "build:docker": "docker build -t aerabi/website-node .",
    "publish:docker": "docker push aerabi/website-node"
  }
}
```

BUILD DOCKERFILE

```
FROM node:16 as builder
```

```
WORKDIR /usr/src/app
```

```
COPY . /usr/src/app/
```

```
RUN npm install
```

```
RUN npm run build:pre &&
```

```
    npm run build:transpile &&
```

```
    npm run build:prod
```

```
RUN chmod +x bin/website-node
```

```
FROM aerabi/base:latest as runner
```

```
COPY --from=builder /usr/src/app/bin/website-node /opt/website-node
```

```
CMD ["/opt/website-node"]
```

CI/CD: DOCKER BUILD JOB

```
# .gitlab-ci.yml
```

```
build:docker:
  stage: build
  image: aerabi/docker:19.03.12-node
  services:
    - docker:19.03.12-dind
  before_script:
    - docker info
    - docker login -u $DOCKER_USER -p $DOCKER_PASSWORD
  script:
    - npm run build:docker
    - npm run publish:docker
  when: manual
```



04

TESTING

You have an extra space
in your line

INTEGRATION TEST

Docker-Compose

NODE.JS



MONGODB REPLICA SET

BUILD COMMANDS

package.json

```
{
  "scripts": {
    "build:pre": "npm run build:routes && npm run build:swagger",
    "build:transpile": "tsc -p src",
    "build:prod": "npm run build:webpack && npm run build:pkg",
    "build:docker": "docker build -t aerabi/website-node .",
    "test:int": "docker-compose up --exit-code-from node node",
    "publish:docker": "docker push aerabi/website-node"
  }
}
```


PIPELINE

BUILD:PRE

Integration test
needs it



BUILD:DOCKER

Create the Docker
image



TEST:INT

Docker-Compose in
Docker



DEPLOY

Let the clients
enjoy it



OTHER TESTS

- code format
- commit lint
- test coverage
- dependency check
- package lock check



**ALMOST
THERE**

PUBLISH

Give all your bugs to
the end user

05



PUBLISH, CHECK, ROLLOUT



PUBLISH

Push the Docker image to registry



PREVIEW

Create a preview environment



CHECK

Do more manual or automated tests



ROLLOUT

Rollout or deploy the image

TIPS



SAVE TIME

You can share artifacts



TEST AWAY

The more tests the merrier



FAIL FAST

Optimize the pipeline DAG



PARALLELIZE

Your jobs can run in parallel



KEEP SHORT

Keep it under 10 minutes



SPLIT

Put your CI conf in multiple files

REPO



The slides + more material will be
available on

github.com/aerabi/docker-2022

gitlab.com/aerabi/docker-2022

ALTERNATIVE RESOURCES



RESOURCES

GITLAB-CI

- [Docker-in-Docker](#)
- [Environments](#)

DOCKER

- [Multi-stage builds](#)

MORE TALKS BY DOCKER COMMUNITY

- [Docker image optimization workshop](#)
- [Hackdockerfest: Security](#)



THANKS!

**DO YOU HAVE ANY
QUESTIONS?**

mohammad-ali@aerabi.com

Twitter @MohammadAliEN

GitHub, LinkedIn @aerabi

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon** and infographics & images by **Freepik**