Real Project along with Docker
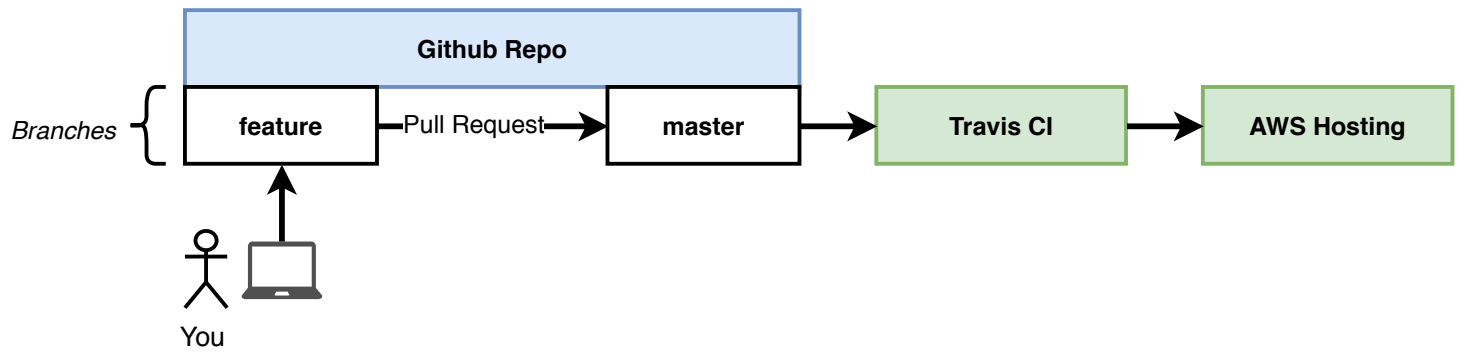
Github Repo

Branches {
feature —Pull Request→ master → **Travis CI** → **AWS Hosting**

You

Flow

| Dev | Create/change features |
| | Make changes on a non-master branch |

Push to github

↓

Create Pull Request to merge with master

↓

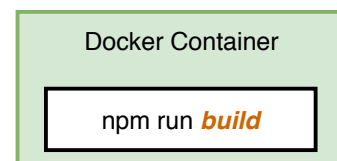| Test | Code pushed to Travis CI |
| | Tests run |

↓

Merge PR with master

↓

Prod	Code pushed to Travis CI
	Tests run
	Deploy to AWS Elastic Beanstalk

Docker Container

npm run *start*
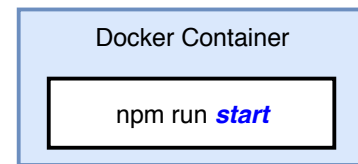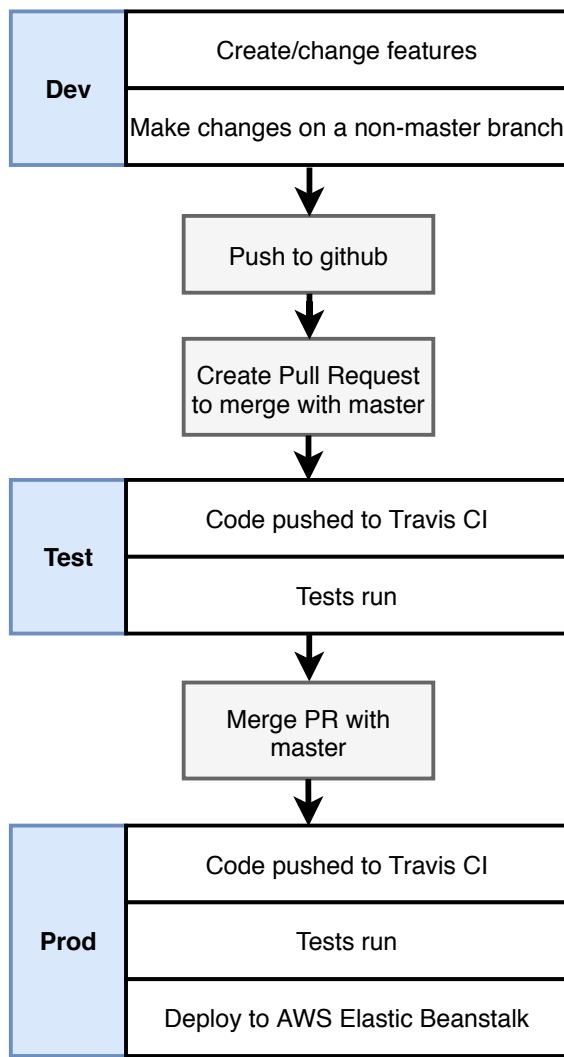
Dockerfile.dev

In Production...

Docker Container

npm run *build*

Dockerfile.prod

Specify **the dockerfile for development**

| docker | **build** | **-f** Dockerfile.dev | . |
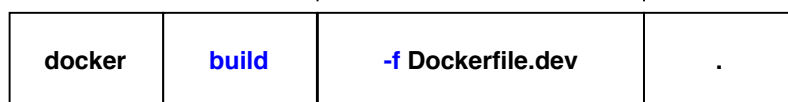
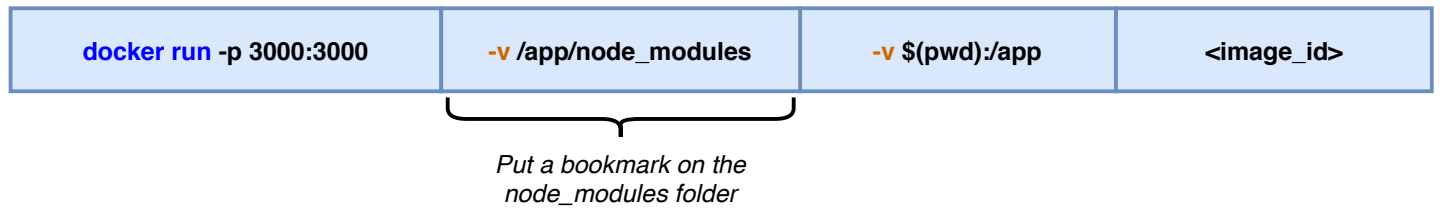Specify directory of files/folders to use for the build

**In development,
to make code changes into inside of container**

*reference folder :
Map the pwd into
the '/app' folder*

| docker run -p 3000:3000 | -v /app/node_modules | -v $(pwd):/app | <image_id> |

*Put a bookmark on the
node_modules folder*

docker-compose.yml

| version: | '3' |

| services: |

| web |

| build: |

| context: | . |

| dockerfile: | Dockerfile.dev | build 'Dockerfiile.dev'

| ports: |

| - | 3000:3000 |

| volumns: |

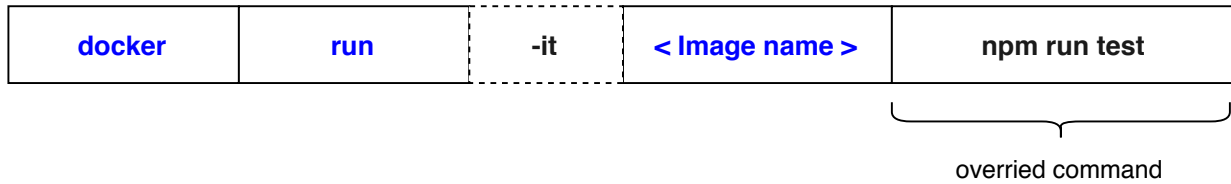| - | /app/node_modules | '/app/node_modules' is a placeholder

| - | .:/app | '/app' inside the container
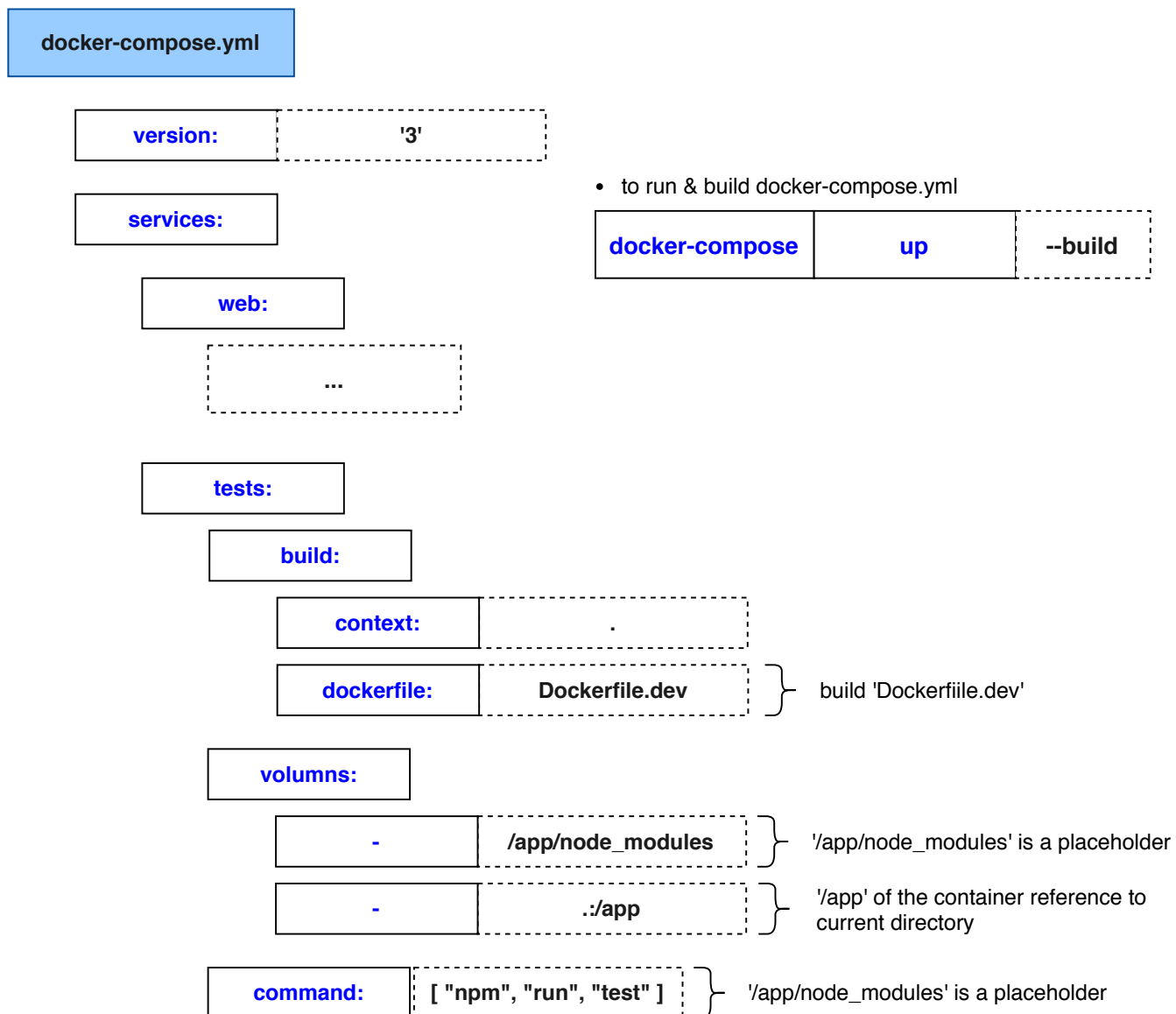**reference to** outside current directory

In test,

- to test docker once

docker	run	-it	< Image name >	npm run test

overried command

- to test docker according to **changed test source codes**

docker-compose	up

build docker-compose and run

docker	ps

Identify **the running container ID** by another terminal

docker	exec	-it	< container ID >	npm run test

overried command
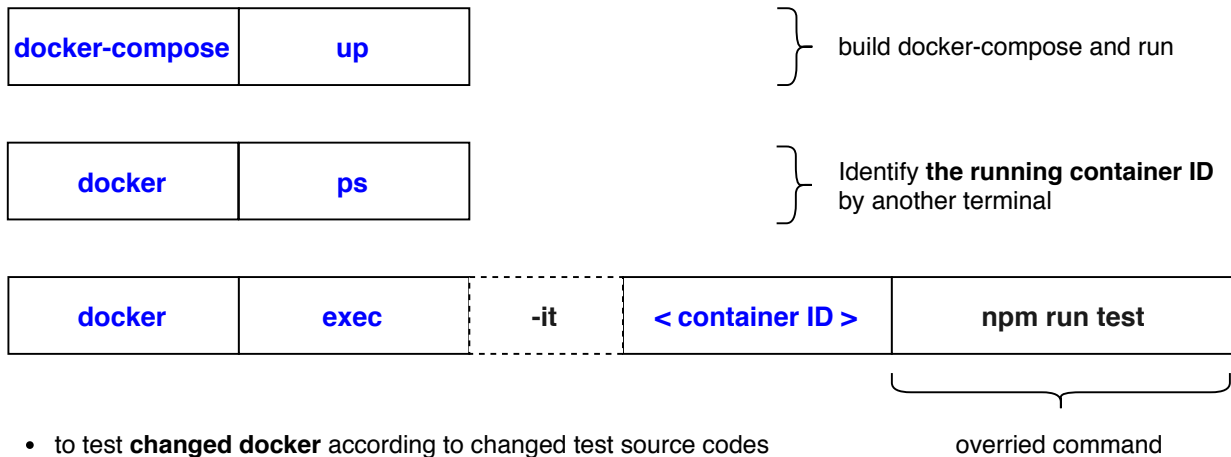
- to test **changed docker** according to changed test source codes

docker-compose.yml

version:	'3'

services:

- to run & build docker-compose.yml

docker-compose	up	--build

    web:

      ...

    tests:

      build:

context:	.

dockerfile:	Dockerfile.dev

build 'Dockerfiile.dev'

      volumns:

-	/app/node_modules

'/app/node_modules' is a placeholder

-	.:/app

'/app' of the container reference to current directory

command:	["npm", "run", "test"]

'/app/node_modules' is a placeholder

In production,

Dev Environment

Web Container

| Dev Server | *index.html* |
| | *main.js* |

Prod Environment

Web Container

| nginx | *index.html* |
| | *main.js* |
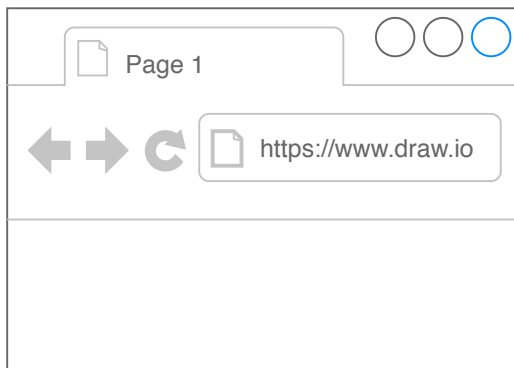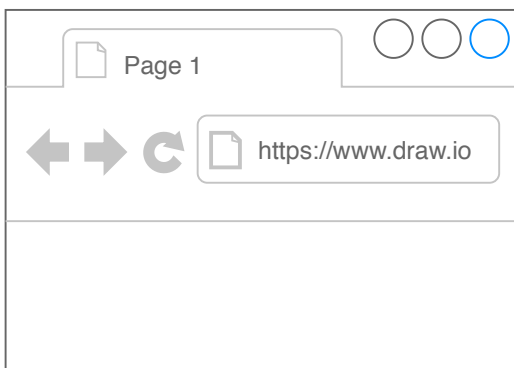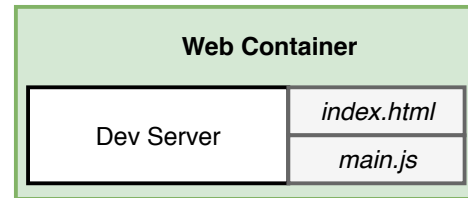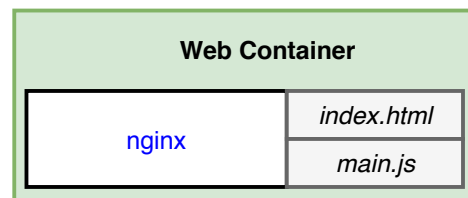
multi configuration of build for production

Build Phase

- Use node:alpine
- Copy the package.json file
- Install dependencies
- Run 'npm run build'

Run Phase

- Use nginx
- Copy over the result of 'npm run build'
- Start nginx

```
unnecessary
to production
```

needed in
production
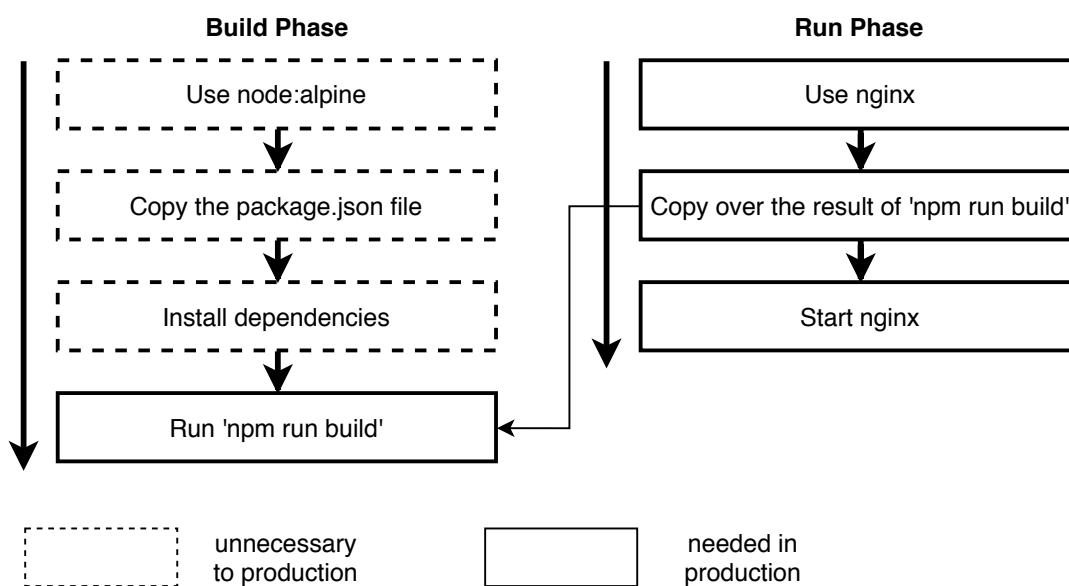
Dockerfile for multi step builds

FROM	node:alpine	as	builder

build stage name

WORKDIR	'/app'

COPY	package.json .

RUN	npm install

COPY	. .

RUN	npm run build

folder inside of **source container**

FROM	nginx

COPY	--from=builder	/app/build	/usr/share/nginx/html

copy from the stage 'builder'

folder inside of **target container**