About Me

- Go Frendi Gunawan
- AI Engineer at Kata.ai
- Ex Teacher at STIKI
- Initiator at Malang Cloud Facebook Group
- Writer at stalchmst
- Netizen at twitter: @gofrendiasgard
- Mere Mortal

Docker

- Container for your application
- Each container are isolated from each other
- Problem to solve: "It works on my machine"

Why Docker

- Isolated environment
- Easy to reproduce
- The facto standard for docker-compose, docker-swarm, and kubernetes

Docker: Some Terminologies

- Image: Definition of a container
  - Written as `Dockerfile`
  - Get list of local images: `docker images`
  - Pull image from registry: `docker pull [OPTIONS] NAME[:TAG|@DIGEST]`
  - Create image: `docker build [OPTIONS] PATH | URL | -`
  - Push image to registry: `docker push [OPTIONS] NAME[:TAG]`
- Image Registry: Place to host docker-images
  - Public: https://hub.docker.com/
  - Private: Usually used internally. Gitlab provide one: https://docs.gitlab.com/ee/user/packages/container_registry/
- Container
  - Get list of running container: `docker ps`
  - Get list of all container: `docker ps -a`
  - Pull and start container: `docker run [OPTIONS] IMAGE [COMMAND] [ARG...]`
  - Start container: `docker start [OPTIONS] CONTAINER [CONTAINER...]`

Dockerfile Example

```
# Parent image
FROM golang:1.13

# The working directory in our container is is `/go/src/app`
WORKDIR /go/src/app
# Copy everything in this directory to container's working directory
COPY . .

# Prepare go binary
RUN go get -d -v ./...
RUN go install -v ./...
RUN go build -o app

# Documentation, so that your infra-engineer
# doesn't have to dive into the code or guess which port is exposed by default
EXPOSE 3000

# when you run the container, this command will be executed
CMD ./app
```

You can build the image by performing:

```
docker build -t stalchmst/awesome-app .
```

Working with Docker

```
# In your computer

# You build an image
docker build -t stalchmst/awesome-app .

# You try to create container named "naruto" based on the image and start it.
# Your app listen to port 3000, but for some reason you want to access it
# from port 8686 of your host.
docker run --name naruto -p 8686:3000 -d stalchmst/awesome-app

# You have verify that the image works as expected,
# now you are ready to push it into public registry
docker push -t stalchmst/awesome-app
```

```
# In someone else's computer

# Can you guess what is he doing?
docker run --name sasuke -p 8080:3000 -d stalchmst/awesome-app
docker run --name sakura -e MODE=verbose -p 8081:3000 -d stalchmst/awesome-app
```

Docker Compose

Your containers work together for a greater good.

Docker Compose (The Philosophy)

- Single Responsibility Principle
- You can't predict the future
- *Bersatu kita teguh, bercerai kita runtuh*

  Instead of creating single container with so many responsibilities, it is better to spawn a lot of containers that can be scaled/maintained independently to each other

What Docker Compose is for

- Prepare development environment as quick as possible (i.e: mysql, redis, rabbit-mq pre-configured)
- Perform integration testing
- Deliver POC
- Deployment: At some point it works and it is enough (i.e: You don't always need kubernetes)

What Docker Compose is not for

- Advance orchestration (i.e: you will net spreadsheet to write down every exposed port and assign those ports manually)
- Auto scale up/down
- Deploy in multi-cluster

Docker Compose: Before we start

- In the face of ambiguity, refuse the temptation to guess
- Official image containers usualy provide good documentation.
  - Here is Mysql's: https://hub.docker.com/_/mysql/
  - Here is Nginx's: https://hub.docker.com/_/nginx/
- Want to dive into the running container? `docker exec -it <container-name> /bin/bash`

Docker Compose: How it looks like

```yaml
version: "3"
services:

  # we define "mysql" service
  mysql:
    # the service container's name should be "stalchmst-mysql"
    container_name: stalchmst-mysql
    # the service is based on "mysql" image
    image: mysql
    # in case of something funny happened, restart
    restart: always
    # We want host's "./mysql/init/init.sql" to be linked to container's
    # docker-entrypoint-initdb.d/init.sql
    volumes:
      - "./mysql/init/init.sql:/docker-entrypoint-initdb.d/init.sql"
    # set some environment variable for the container
    environment:
      - MYSQL_ROOT_PASSWORD=toor
      - MYSQL_DATABASE=db
      - MYSQL_USER=user
      - MYSQL_PASSWORD=password
    # We don't expose MySQL port's to the host, because we don't need to.
    # All container defined in docker-compose.yaml can talk to each other
    # using their default ports. (MySQL's is 3306)
```

つづく

Docker Compose: How it looks like (Continued)

```yaml
nginx:
  container_name: stalchmst-nginx
  image: nginx
  restart: always
  volumes:
    - "./nginx/conf.d:/etc/nginx/conf.d"
    - "./static-content:/usr/share/nginx/html"
  ports:
    - "80:80"
  environment:
    - NGINX_PORT=80

api:
  container_name: stalchmst-api
  # this one is different, we build the image based on the source code
  # in `api` directory, and we don't put it on the registry.
  build: api
  restart: always
  environment:
    - MYSQL_CONNECTION_STRING=user:password@tcp(stalchmst-mysql:3306)/db
    - MYSQL_MAX_CONNECT_ATTEMPT=50
  depends_on:
    - mysql
```

Docker Compose, Up and Down

Start all services

```
docker-compose up

docker-compose up -d # run everything in the background
docker-compose up --build -d # build image and run everything
                             # in the background
```

Stop all services

```
docker-compose down
```

Our Project (UI)

Tmplt

A template you can use to generate documents. Choose one of the template, put your data, and impress your boss

Template Code

pantun-receh

Template

```
Jalan jalan ke kota {{.Kota}}
Jangan lupa membeli {{.Barang}}
Buat apa ke kota {{.Kota}}
Kalau cuma membeli {{.Barang}}
```

Data

```
{
  "Kota": "Jakarta",
  "Barang": "Jenang"
}
```

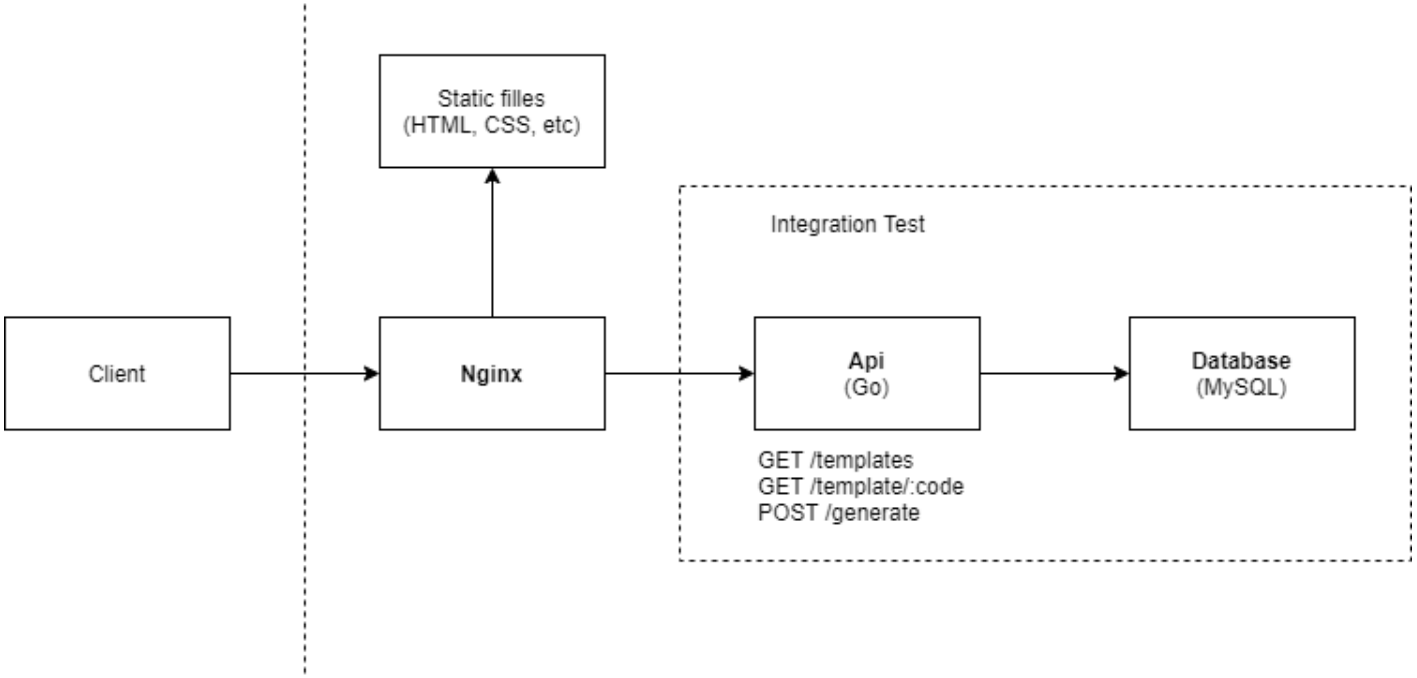
Generate and copy to clipboard

Generated Document

```
Jalan jalan ke kota Jakarta
Jangan lupa membeli Jenang
Buat apa ke kota Jakarta
Kalau cuma membeli Jenang
```

.:: Stalchmst, 2020 ::.

Our Project (Architecture)



Docker Compose for testing: Directory structure

```
api
├── docker-compose.yml
├── generator.go
├── generator_test.go
├── go.mod
├── go.sum
├── main.go
├── mysql-test-init
│   └── init.sql
├── template.go
├── template_test.go
└── test.sh
```

Docker Compose for testing: docker-compose.yml

```yaml
version: "3"
services:

  mysql:
    container_name: stalchmst-mysql-test
    image: mysql
    restart: always
    volumes:
      - "./mysql-test-init/init.sql:/docker-entrypoint-initdb.d/init.sql"
    ports:
      - "3307:3306"
    environment:
      - MYSQL_ROOT_PASSWORD=toor
      - MYSQL_DATABASE=db
      - MYSQL_USER=user
      - MYSQL_PASSWORD=password
```

Docker Compose for testing: test.sh

```
#!/bin/sh

# setup
echo ">> Setting up testing environment"
docker-compose down
docker-compose up --build -d

# waiting
echo ">> Wait MySQL to be ready"
while ! docker exec stalchmst-mysql-test mysql -u user -ppassword -e "USE db;"; do
  sleep 1 # wait for 1 second to try again
  echo ">> Retry..."
done

# test
echo ">> Start testing"
env MYSQL_CONNECTION_STRING="user:password@tcp(localhost:3307)/db" MYSQL_MAX_CONNE

# tear down
echo ">> Clean up testing environment"
# TODO: when something goes wrong and you suspect MySQL has
# something to do with the failure, uncomment this one
docker-compose down
echo ">> Done"
```

Docker Compose for deployment

- CI/CD: buddy works https://app.buddy.works/
- Deployment machine: Vultr with ssh access https://www.vultr.com/
- The Source Code: https://github.com/goFrendiAsgard/stalchmst
- Typical CI/CD steps:
  - Run test
  - Build docker images
  - Publish images to the registry *(We skip this one)*
  - Deploy

Docker Compose for deployment: CI/CD

EXECUTED REVISION

#f1fd1a3: better tmplt (1 commit) master

ACTIONS PERFORMED IN THIS EXECUTION

EXECUTED ACTION	STATUS	
Prepare Environment	00:00	Logs
[stalchmst.com] Execute: git pull origin master	00:04	Logs

Raw logs Fullscreen

```
Resolving host stalchmst.com...
Connected. Executing commands...
git pull origin master
From github.com:goFrendiAsgard/stalchmst
 * branch            master     -> FETCH_HEAD
   31cbfa9..f1fd1a3  master     -> origin/master
Updating 31cbfa9..f1fd1a3
Fast-forward
 api/generator.go             | 28 ++++++++++++++++++
 api/generator_test.go        | 27 +++++++++++++++++
 api/main.go                  | 15 ++--------
```

EXECUTE

Edit this action

[stalchmst.com] Execute: ./test.sh	00:25	Logs
[stalchmst.com] Execute: docker-compose up -d	00:38	Logs

META

Docker Compose for deployment: CI/CD

Execute: git pull origin master

```
git pull origin master
```

Execute: ./test.sh

```
cd api
./test.sh
```

Execute: docker-compose up -d

```
docker-compose down --rmi=local
docker-compose up -d
```

Bonus: Can I Use Docker for Development Environment

Development inside the container is possible, but not everyone agree with this:

https://code.visualstudio.com/docs/remote/containers

That's All

Q/A?

Join Malang Cloud Facebook Group