Jorge Dias

Docker on Rails

About me

Devops @ Xing / Development & Testing infrastructure

Github @diasjorge / Twitter @dias_jorge

Docker introduction

Docker - An open platform for distributed applications for developers and sysadmins.

Docker enables apps to be quickly assembled from components

Eliminates the friction between development, QA, and production environments.

Run the same app, unchanged, on laptops, data center VMs, and any cloud.

Docker Basics Demo

Our test app: https://github.com/copycopter/copycopter-server

Dockerizing our applications dependencies

```
docker run -d --name copycopter-postgres -e POSTGRES_USER=copycopter -e POSTGRES_PASSWORD=copycopter -p 5432:5432 postgres
```

5e70c8a494ea731e9d365662d8009b6e0e5fb3fca277263dbe7dbbec97823cd3

```
docker ps
```

```
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                    PORTS                    NAMES
5e70c8a494ea        postgres:latest     "/docker-entrypoint.…"   2 days ago          Up Less than a second     0.0.0.0:5432->5432/tcp   copycopter-postgres
```

Updating our config/database.yml

```
development: &default
  adapter: postgresql
  database: copycopter_development
  min_messages: WARNING
  host: <%= ENV['POSTGRES_HOST'] %>
  username: <%= ENV['POSTGRES_USER'] %>
  password: <%= ENV['POSTGRES_PASSWORD'] %>

test: &test
  <<: *default
  database: copycopter_test
```

Let's run our application locally with our changes

Dockerizing the app

Create our Dockerfile

```
FROM ruby:2.1.5
RUN apt-get update &&     DEBIAN_FRONTEND=noninteractive apt-get install -y     nodejs     libqt4-dev &&     rm -rf /var/lib/apt/lists/*
RUN usr/sbin/useradd --create-home --home-dir /app --shell /bin/bash copycopter
WORKDIR /app
COPY Gemfile* /app/
RUN bundle install
ADD . /app
RUN chown -R copycopter:copycopter /app
USER copycopter
CMD ["rails", "server"]
```

```
docker build -t copycopter-app .
```

```
Sending build context to Docker daemon 4.231 MB
Sending build context to Docker daemon
Step 0 : FROM ruby:2.1.5
 ---> 2555e4979e69
Step 1 : RUN apt-get update &&     DEBIAN_FRONTEND=noninteractive apt-get install -y     nodejs     libqt4-dev &&     rm -rf /var/lib/apt/lists/*
 ---> Using cache
 ---> 144fc3b2021b
Step 2 : RUN usr/sbin/useradd --create-home --home-dir /app --shell /bin/bash copycopter
 ---> Using cache
 ---> aef3c31b9b19
Step 3 : WORKDIR /app
 ---> Using cache
 ---> 8ebe02812d9e
Step 4 : COPY Gemfile* /app/
 ---> Using cache
 ---> b0da3d4fb821
Step 5 : RUN bundle install
 ---> Using cache
 ---> a10705098e5e
Step 6 : ADD . /app
 ---> Using cache
 ---> f3ebfb7860c7
Step 7 : RUN chown -R copycopter:copycopter /app
 ---> Using cache
 ---> 80d612aee4f9
Step 8 : USER copycopter
 ---> Using cache
 ---> 2a6591b4cfc6
Step 9 : CMD rails server
 ---> Using cache
 ---> caf23490eebe
Successfully built caf23490eebe
```

Let's run our application inside docker

Fig introduction

Fast, isolated development environments using Docker

Creating our fig.yml

```
web:
  build: .
  ports:
    - "3000:3000"
  links:
    - postgres
  volumes:
    - ".:/app"
  environment:
    POSTGRES_HOST: 'postgres'
    POSTGRES_PASSWORD: 'copycopter-dev'
    POSTGRES_USER: 'copycopter'

postgres:
  image: 'postgres'
  environment:
    POSTGRES_USER: 'copycopter'
    POSTGRES_PASSWORD: 'copycopter-dev'
```

Let's run our application using fig

That's all, thanks!