

Unicorn  
+  
NGINX



Phusion Passenger



# Dockerizing Rails & Zero Downtime Deployment

Using Docker

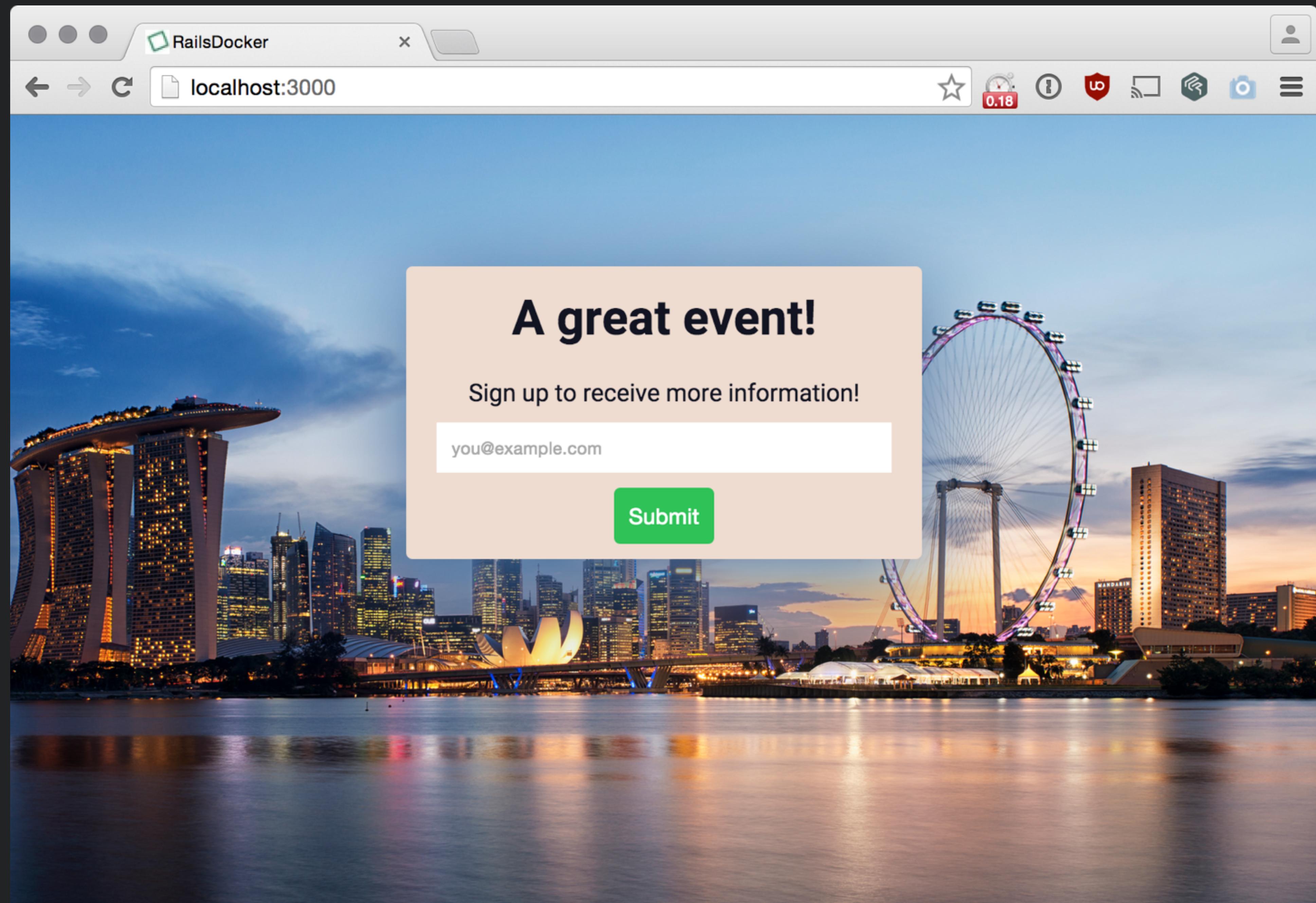


@oliver\_ponder



@giantswarm

# Lead App



# Lead App

Rails 4.2.4

MySQL

Name	Date Modified	Size	Kind
config.ru	Today 11:03	153 bytes	Document
Gemfile.lock	Today 11:03	4 KB	Document
README.rdoc	Today 11:03	296 bytes	Document
app	Today 11:04	--	Folder
assets	Today 11:04	--	Folder
controllers	Today 11:03	--	Folder
concerns	Today 11:03	--	Folder
application_controller.rb	Today 11:03	206 bytes	Ruby script
leads_controller.rb	Today 11:03	201 bytes	Ruby script
helpers	Today 11:03	--	Folder
mailers	Today 11:03	--	Folder
models	Today 11:03	--	Folder
concerns	Today 11:03	--	Folder
lead.rb	Today 11:03	35 bytes	Ruby script
views	Today 11:03	--	Folder
bin	Today 11:03	--	Folder
config	Today 11:03	--	Folder
db	Today 11:03	--	Folder
lib	Today 11:03	--	Folder
log	Today 11:03	--	Folder
public	Today 11:03	--	Folder
test	Today 11:03	--	Folder
vendor	Today 11:04	--	Folder
Gemfile	Today 11:03	393 bytes	TextEd...ument
Rakefile	Today 11:03	249 bytes	TextEd...ument

```
1 Rails.application.routes.draw do
2   resources :leads
3   root 'leads#new'
4 end
5
```

config/routes.rb

```
1< class LeadsController < ApplicationController
2
3  def new
4    @lead = Lead.new
5  end
6
7< def create
8  @lead = Lead.create(params.permit(:email))
9  redirect_to root_path
10 end
11
12 def index
13  @leads = Lead.all
14 end
15 end
```

app/controllers/leads\_controller.rb

```
1< class LeadsController < ApplicationController
2
3  def new
4    @lead = Lead.new
5  end
6
7< def create
8  @lead = Lead.create(params.permit(:email))
9  redirect_to root_path
10 end
11
12 def index
13  @leads = Lead.all
14 end
15 end
```

app/controllers/leads\_controller.rb

```
1< class LeadsController < ApplicationController
2
3  def new
4    @lead = Lead.new
5  end
6
7< def create
8  @lead = Lead.create(params.permit(:email))
9  redirect_to root_path
10 end
11
12 def index
13  @leads = Lead.all
14 end
15 end
```

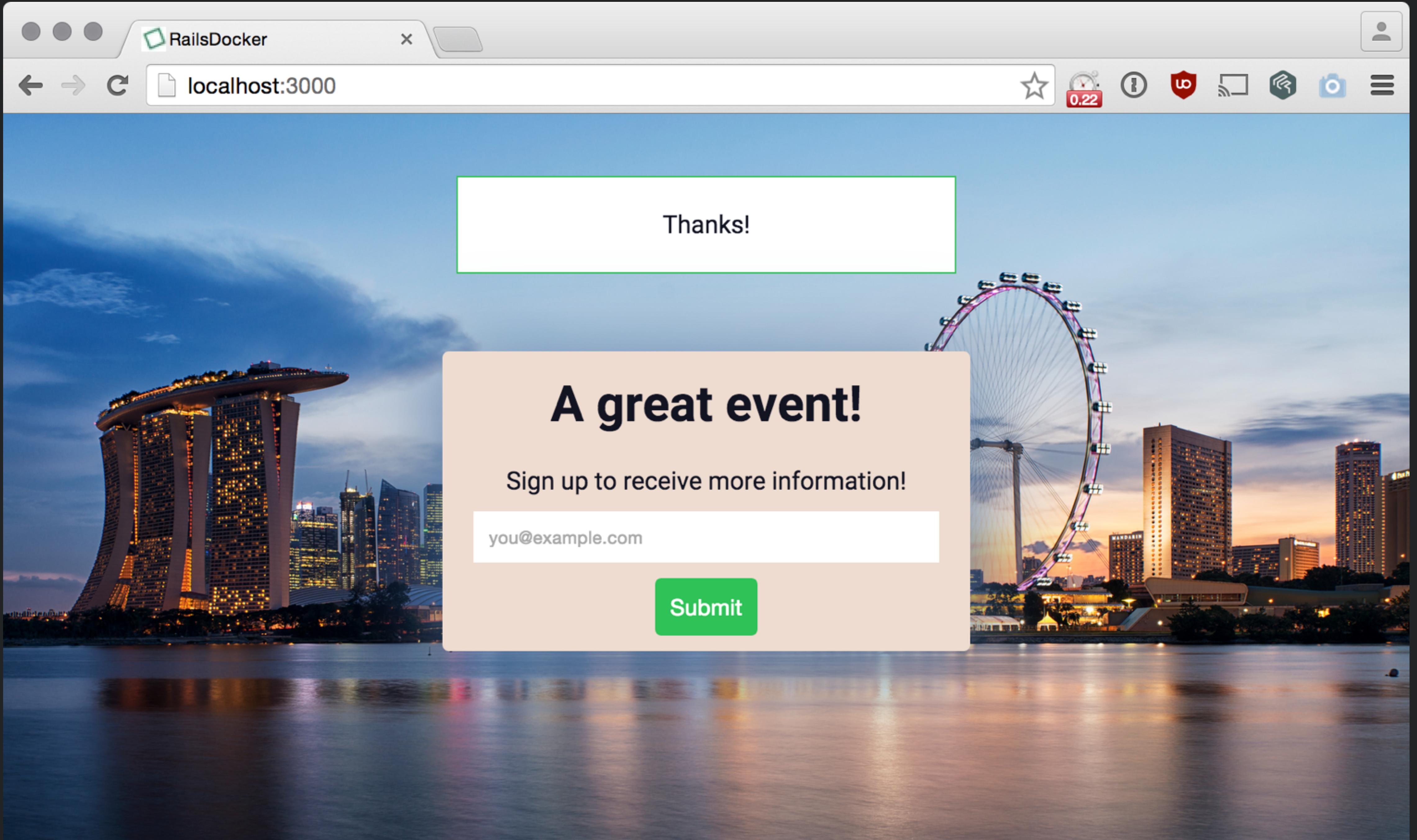
app/controllers/leads\_controller.rb

```
1< class LeadsController < ApplicationController
2
3  def new
4    @lead = Lead.new
5  end
6
7< def create
8  @lead = Lead.create(params.permit(:email))
9  redirect_to root_path
10 end
11
12 def index
13  @leads = Lead.all
14 end
15 end
```

app/controllers/leads\_controller.rb

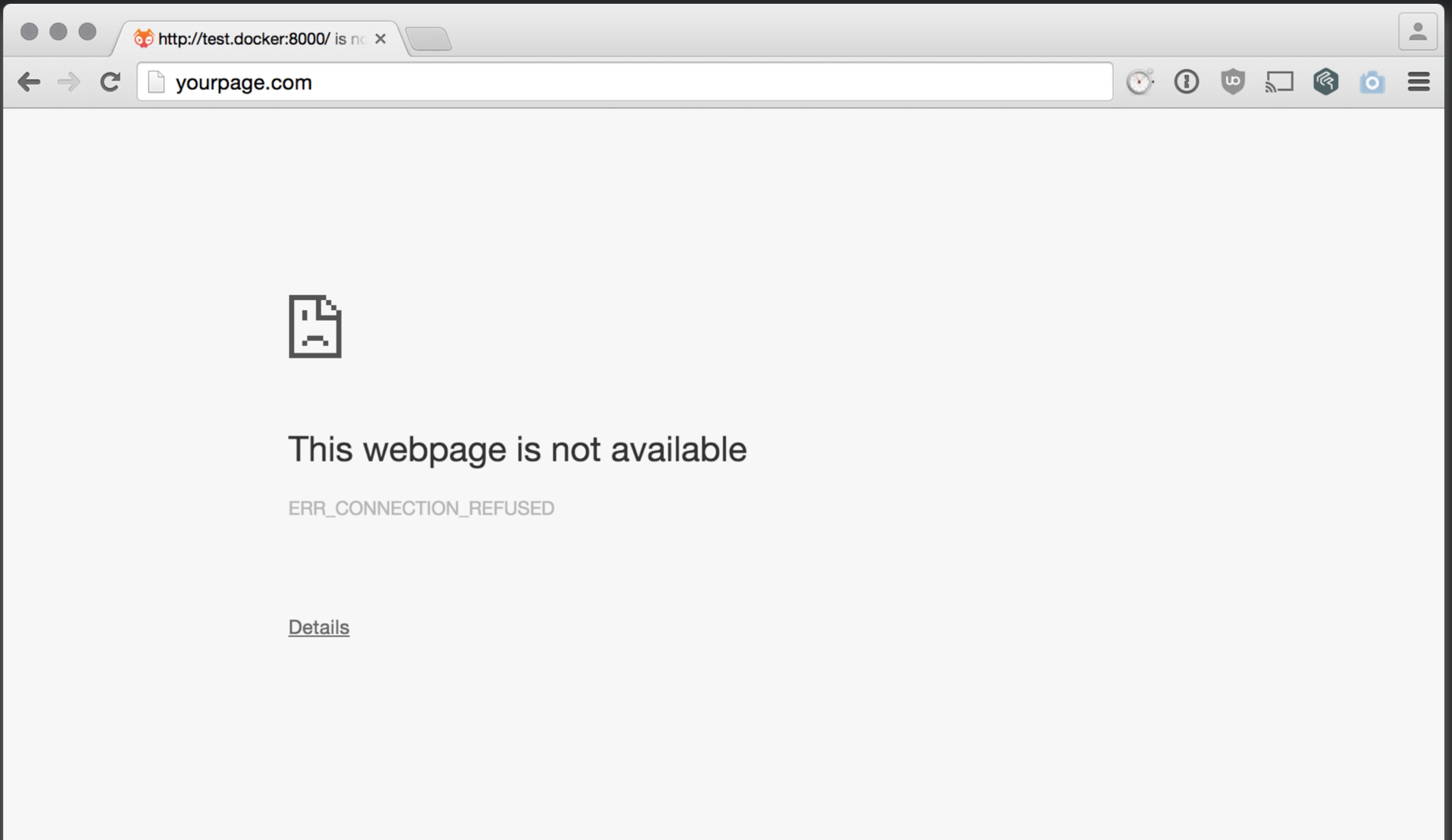
```
1 development:  
2   adapter: mysql2  
3   database: sg_meetup  
4   pool: 5  
5   username: root  
6   host: localhost
```

config/database.yml



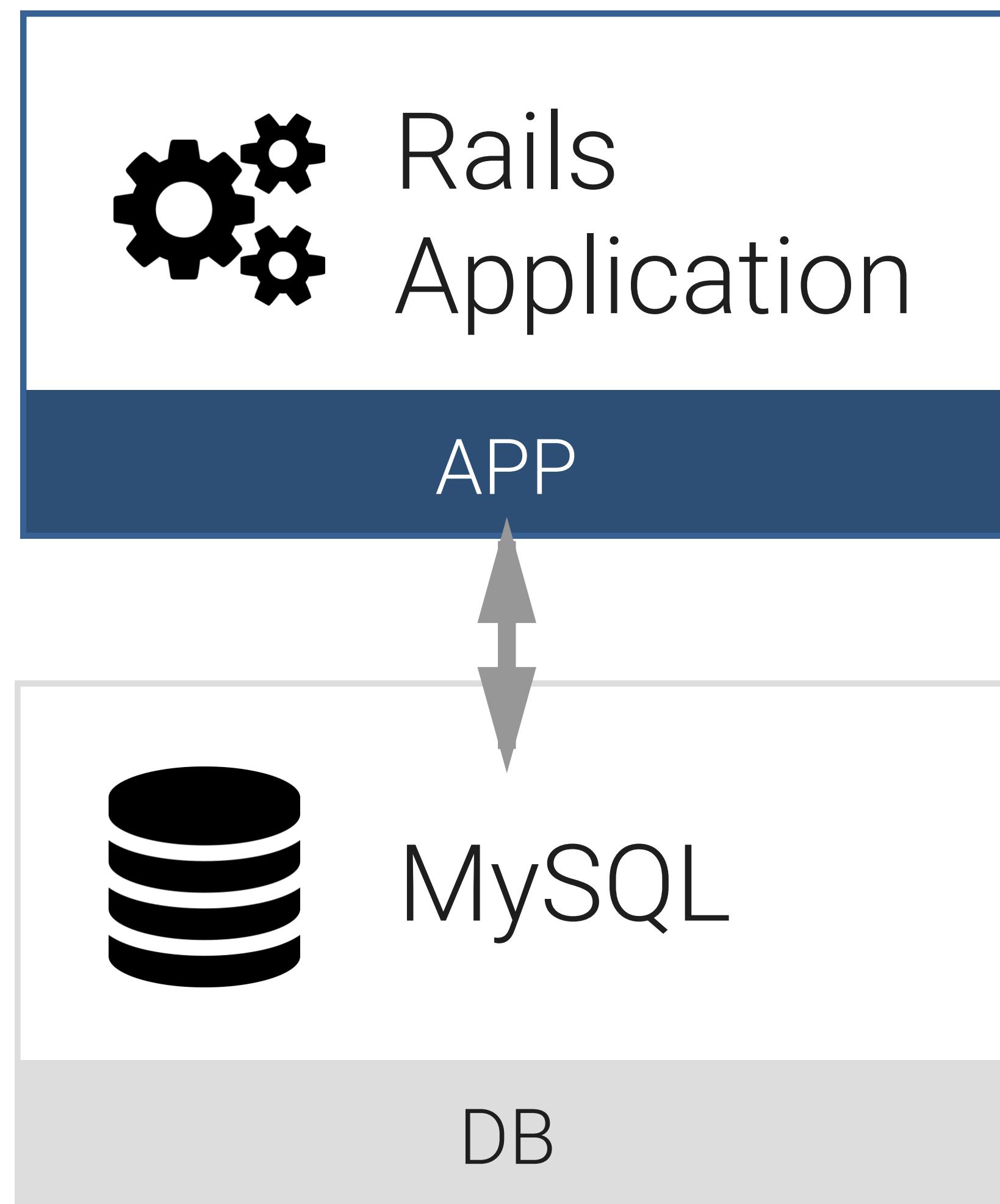
cap production deploy



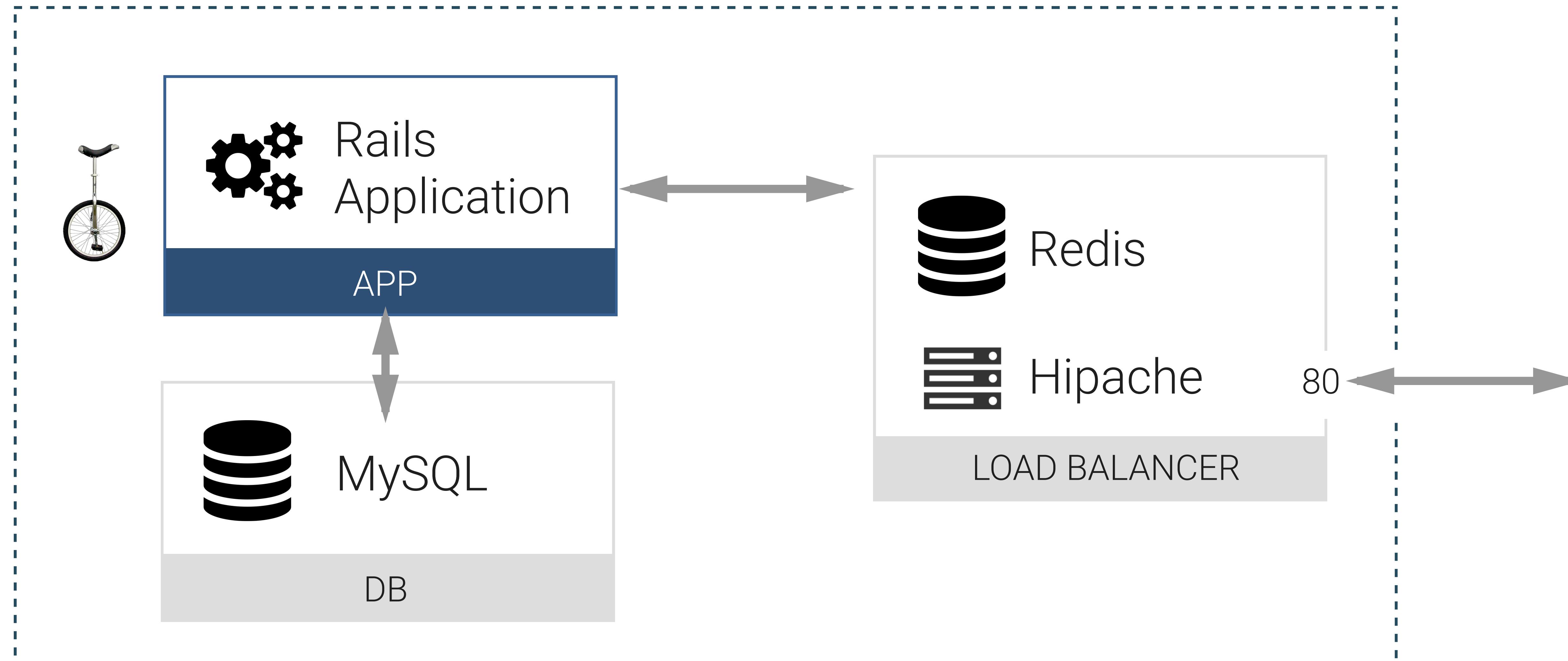




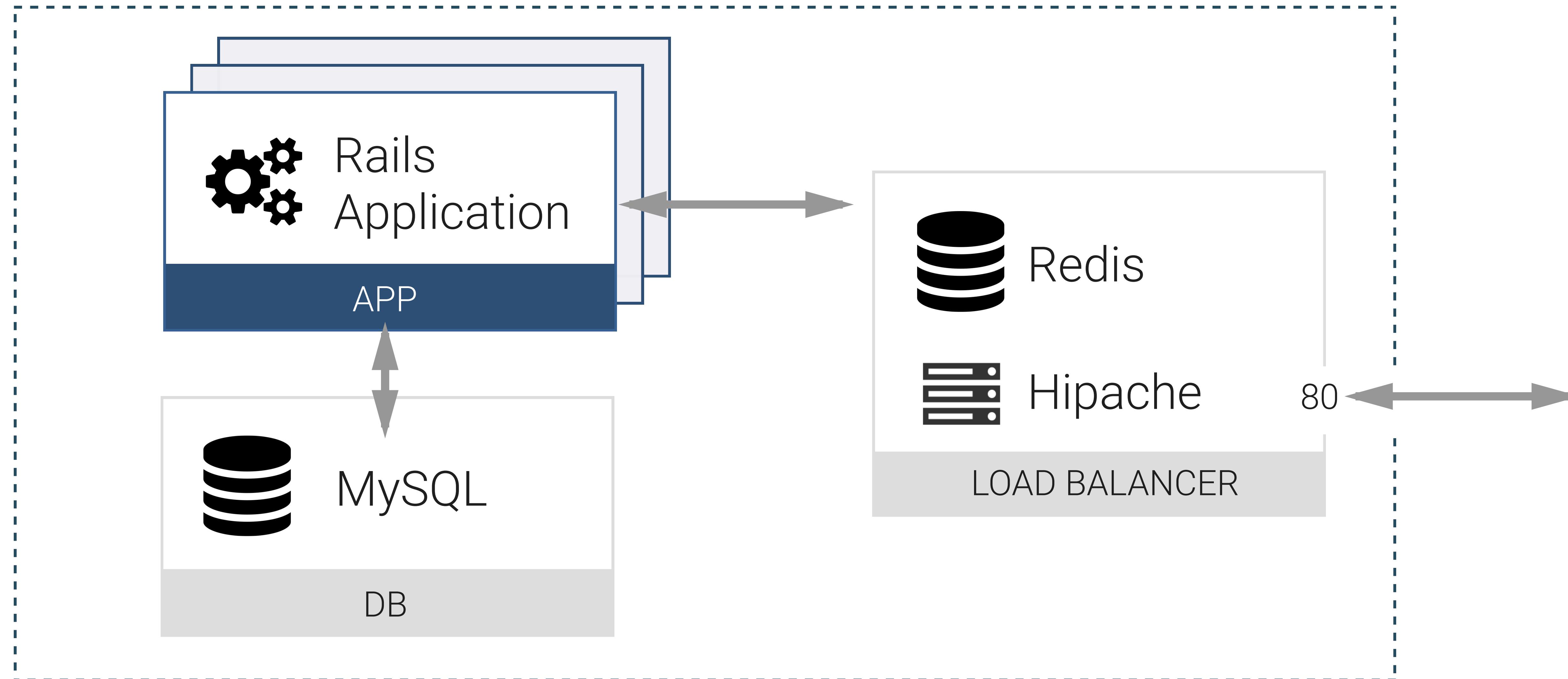




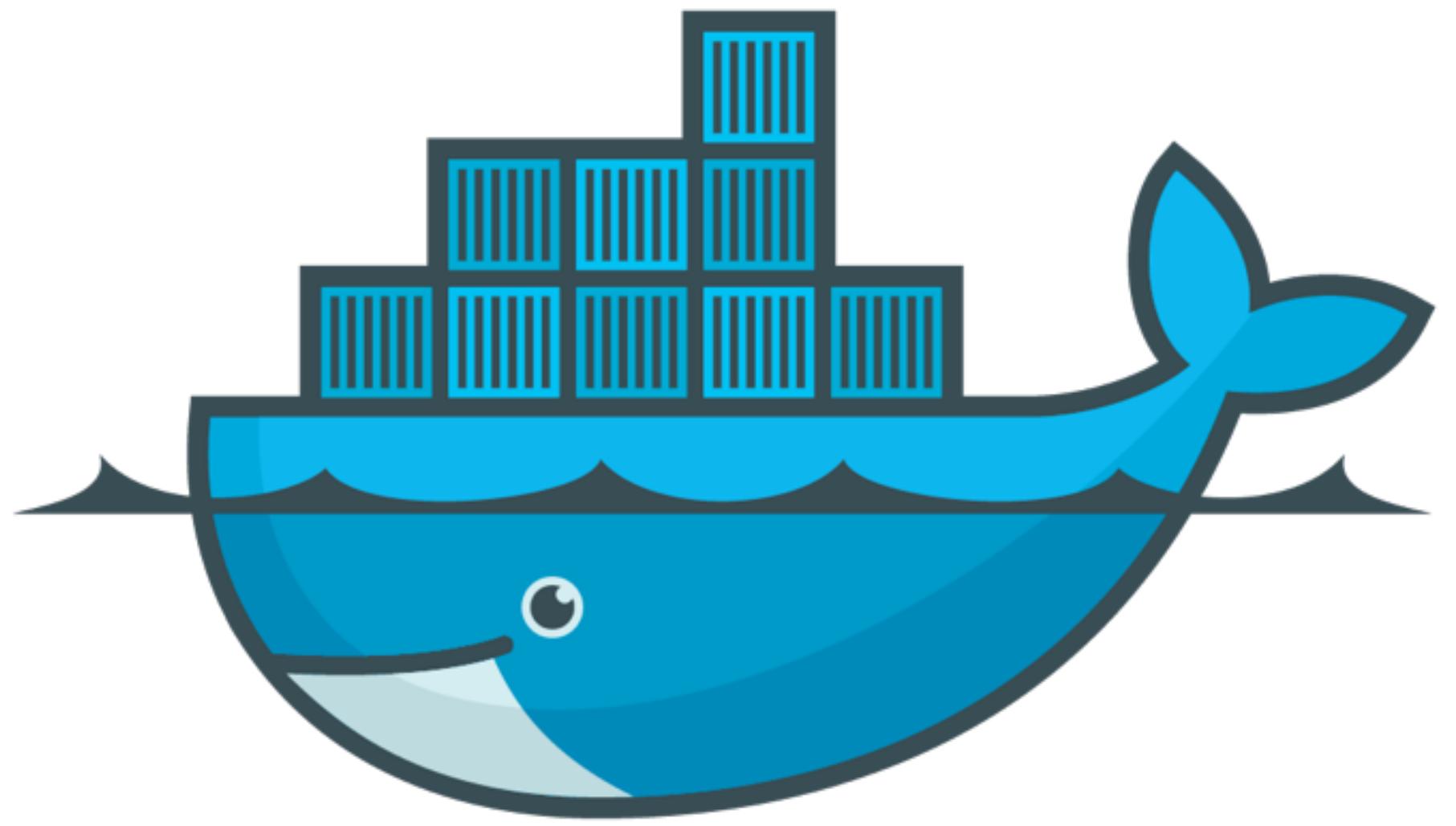
Application Architecture



Application Architecture



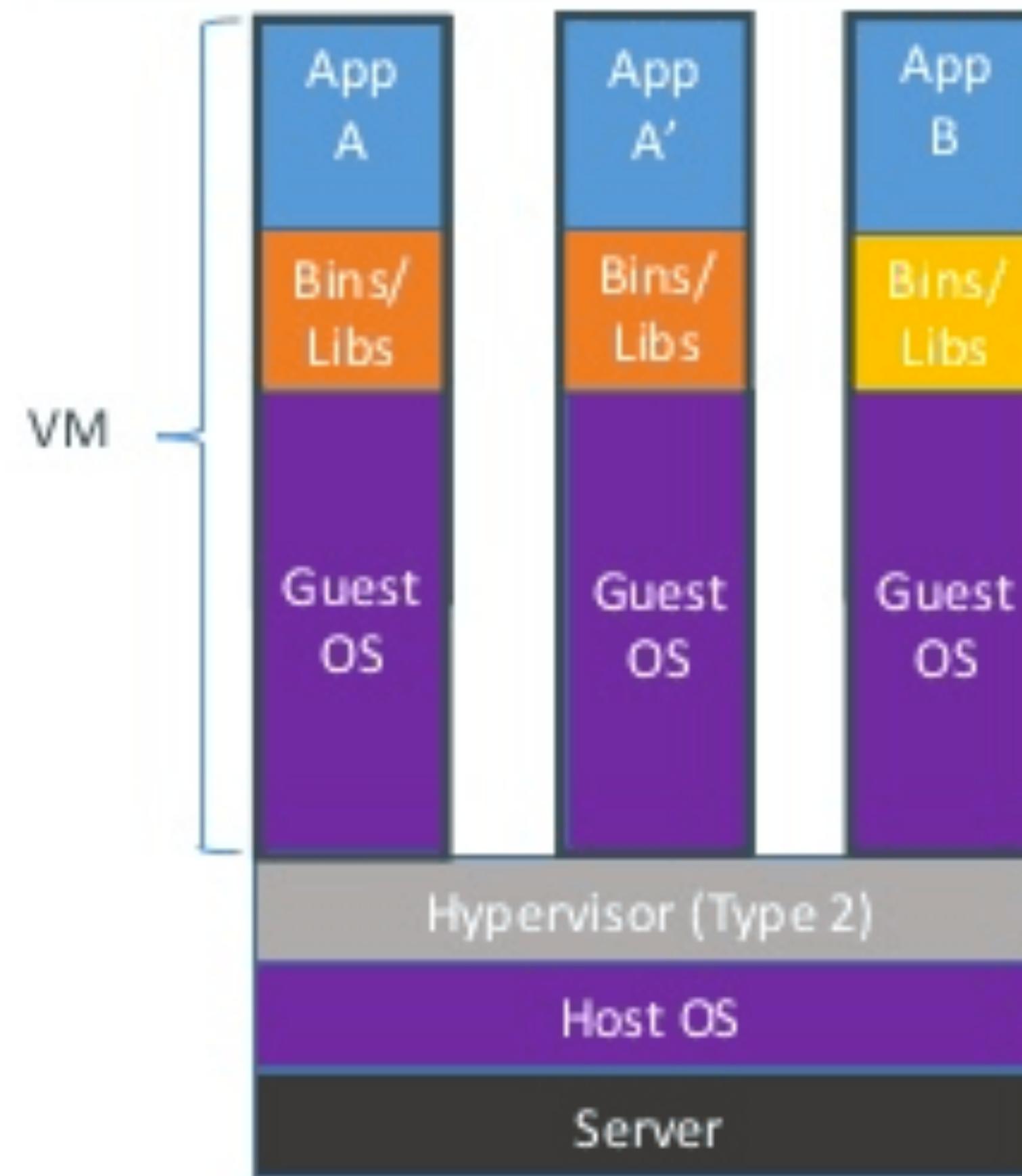
Application Architecture



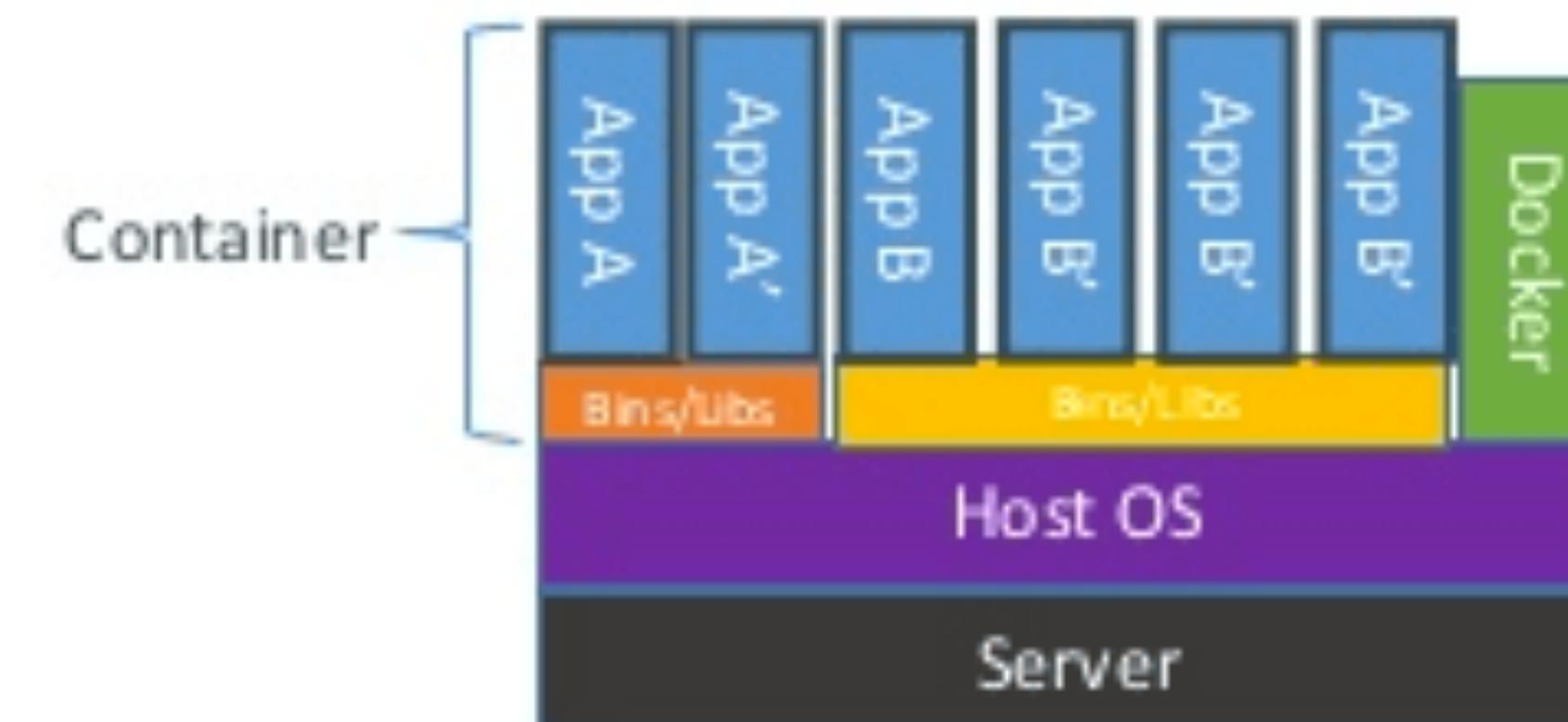
docker

# Containers vs. VMs

---



Containers are isolated,  
but share OS and, where  
appropriate, bins/libraries

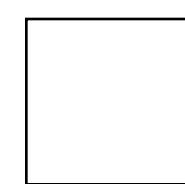
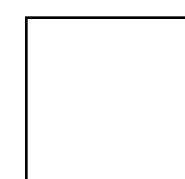
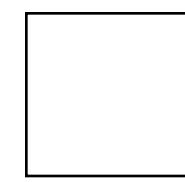
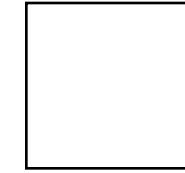
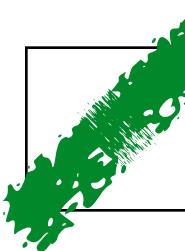


# Containerization Checklist

- install docker
- Dockerfile
- docker-compose.yml
- fix database.yml
- startup script

<https://docs.docker.com/mac/started/>

# Containerization Checklist



install docker

Dockerfile

docker-compose.yml

fix database.yml

startup script

# Dockerfile

Name	Date Modified	Size	Kind
config.ru	Today 11:03	153 bytes	Document
Gemfile.lock	Today 11:19	4 KB	Document
README.rdoc	Today 11:03	296 bytes	Document
app	Today 11:04	--	Folder
bin	Today 11:03	--	Folder
config	Today 11:03	--	Folder
db	Today 11:03	--	Folder
lib	Today 11:03	--	Folder
log	Today 11:03	--	Folder
public	Today 11:03	--	Folder
test	Today 11:03	--	Folder
tmp	Today 11:19	--	Folder
vendor	Today 11:04	--	Folder
Dockerfile	Today 14:26	Zero bytes	TextEd...ument
Gemfile	Today 11:03	393 bytes	TextEd...ument
Rakefile	Today 11:03	249 bytes	TextEd...ument

```
1 |FROM ubuntu:14.04  
2  
3 CMD ["echo", "Hello World"]
```

## Dockerfile

```
1 FROM ruby:2.2.3
2
3 RUN apt-get update -qq && apt-get install -y build-essential
4
5 RUN apt-get install -y libmysqlclient-dev
6 RUN apt-get install -y libxml2-dev libxslt1-dev
7 RUN apt-get install -y nodejs
8 RUN apt-get install -y redis-tools
9 RUN apt-get install -y netcat
10
11 ENV APP_HOME /myapp
12 RUN mkdir $APP_HOME
13 WORKDIR $APP_HOME
14
15 # Add Gemfile here, before the app files
16 # That way we can cache the bundle install step
17 ADD Gemfile* $APP_HOME/
18 RUN bundle install
19
20 ADD . $APP_HOME
21
22 CMD ["./start_rails"]
```

```
1 FROM ruby:2.2.3
2
3 RUN apt-get update -qq && apt-get install -y build-essential
4
5 RUN apt-get install -y libmysqlclient-dev
6 RUN apt-get install -y libxml2-dev libxslt1-dev
7 RUN apt-get install -y nodejs
8 RUN apt-get install -y redis-tools
9 RUN apt-get install -y netcat
10
11 ENV APP_HOME /myapp
12 RUN mkdir $APP_HOME
13 WORKDIR $APP_HOME
14
15 # Add Gemfile here, before the app files
16 # That way we can cache the bundle install step
17 ADD Gemfile* $APP_HOME/
18 RUN bundle install
19
20 ADD . $APP_HOME
21
22 CMD ["./start_rails"]
```

```
1 FROM ruby:2.2.3
2
3 RUN apt-get update -qq && apt-get install -y build-essential
4
5 RUN apt-get install -y libmysqlclient-dev
6 RUN apt-get install -y libxml2-dev libxslt1-dev
7 RUN apt-get install -y nodejs
8 RUN apt-get install -y redis-tools
9 RUN apt-get install -y netcat
10
11 ENV APP_HOME /myapp
12 RUN mkdir $APP_HOME
13 WORKDIR $APP_HOME
14
15 # Add Gemfile here, before the app files
16 # That way we can cache the bundle install step
17 ADD Gemfile* $APP_HOME/
18 RUN bundle install
19
20 ADD . $APP_HOME
21
22 CMD ["./start_rails"]
```

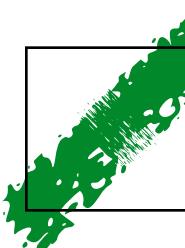
```
1 FROM ruby:2.2.3
2
3 RUN apt-get update -qq && apt-get install -y build-essential
4
5 RUN apt-get install -y libmysqlclient-dev
6 RUN apt-get install -y libxml2-dev libxslt1-dev
7 RUN apt-get install -y nodejs
8 RUN apt-get install -y redis-tools
9 RUN apt-get install -y netcat
10
11 ENV APP_HOME /myapp
12 RUN mkdir $APP_HOME
13 WORKDIR $APP_HOME
14
15 # Add Gemfile here, before the app files
16 # That way we can cache the bundle install step
17 ADD Gemfile* $APP_HOME/
18 RUN bundle install
19
20 ADD . $APP_HOME
21
22 CMD ["./start_rails"]
```

```
1 FROM ruby:2.2.3
2
3 RUN apt-get update -qq && apt-get install -y build-essential
4
5 RUN apt-get install -y libmysqlclient-dev
6 RUN apt-get install -y libxml2-dev libxslt1-dev
7 RUN apt-get install -y nodejs
8 RUN apt-get install -y redis-tools
9 RUN apt-get install -y netcat
10
11 ENV APP_HOME /myapp
12 RUN mkdir $APP_HOME
13 WORKDIR $APP_HOME
14
15 # Add Gemfile here, before the app files
16 # That way we can cache the bundle install step
17 ADD Gemfile* $APP_HOME/
18 RUN bundle install
19
20 ADD . $APP_HOME
21
22 CMD ["./start_rails"]
```

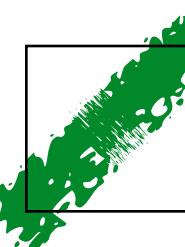
```
1 FROM ruby:2.2.3
2
3 RUN apt-get update -qq && apt-get install -y build-essential
4
5 RUN apt-get install -y libmysqlclient-dev
6 RUN apt-get install -y libxml2-dev libxslt1-dev
7 RUN apt-get install -y nodejs
8 RUN apt-get install -y redis-tools
9 RUN apt-get install -y netcat
10
11 ENV APP_HOME /myapp
12 RUN mkdir $APP_HOME
13 WORKDIR $APP_HOME
14
15 # Add Gemfile here, before the app files
16 # That way we can cache the bundle install step
17 ADD Gemfile* $APP_HOME/
18 RUN bundle install
19
20 ADD . $APP_HOME
21
22 CMD ["./start_rails"]
```

```
1 FROM ruby:2.2.3
2
3 RUN apt-get update -qq && apt-get install -y build-essential
4
5 RUN apt-get install -y libmysqlclient-dev
6 RUN apt-get install -y libxml2-dev libxslt1-dev
7 RUN apt-get install -y nodejs
8 RUN apt-get install -y redis-tools
9 RUN apt-get install -y netcat
10
11 ENV APP_HOME /myapp
12 RUN mkdir $APP_HOME
13 WORKDIR $APP_HOME
14
15 # Add Gemfile here, before the app files
16 # That way we can cache the bundle install step
17 ADD Gemfile* $APP_HOME/
18 RUN bundle install
19
20 ADD . $APP_HOME
21
22 CMD ["./start_rails"]
```

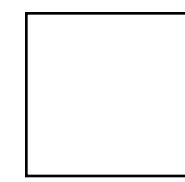
# Containerization Checklist



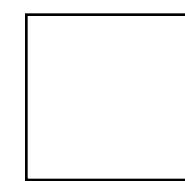
install docker



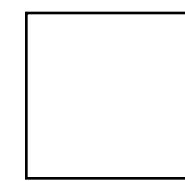
Dockerfile



docker-compose.yml



fix database.yml

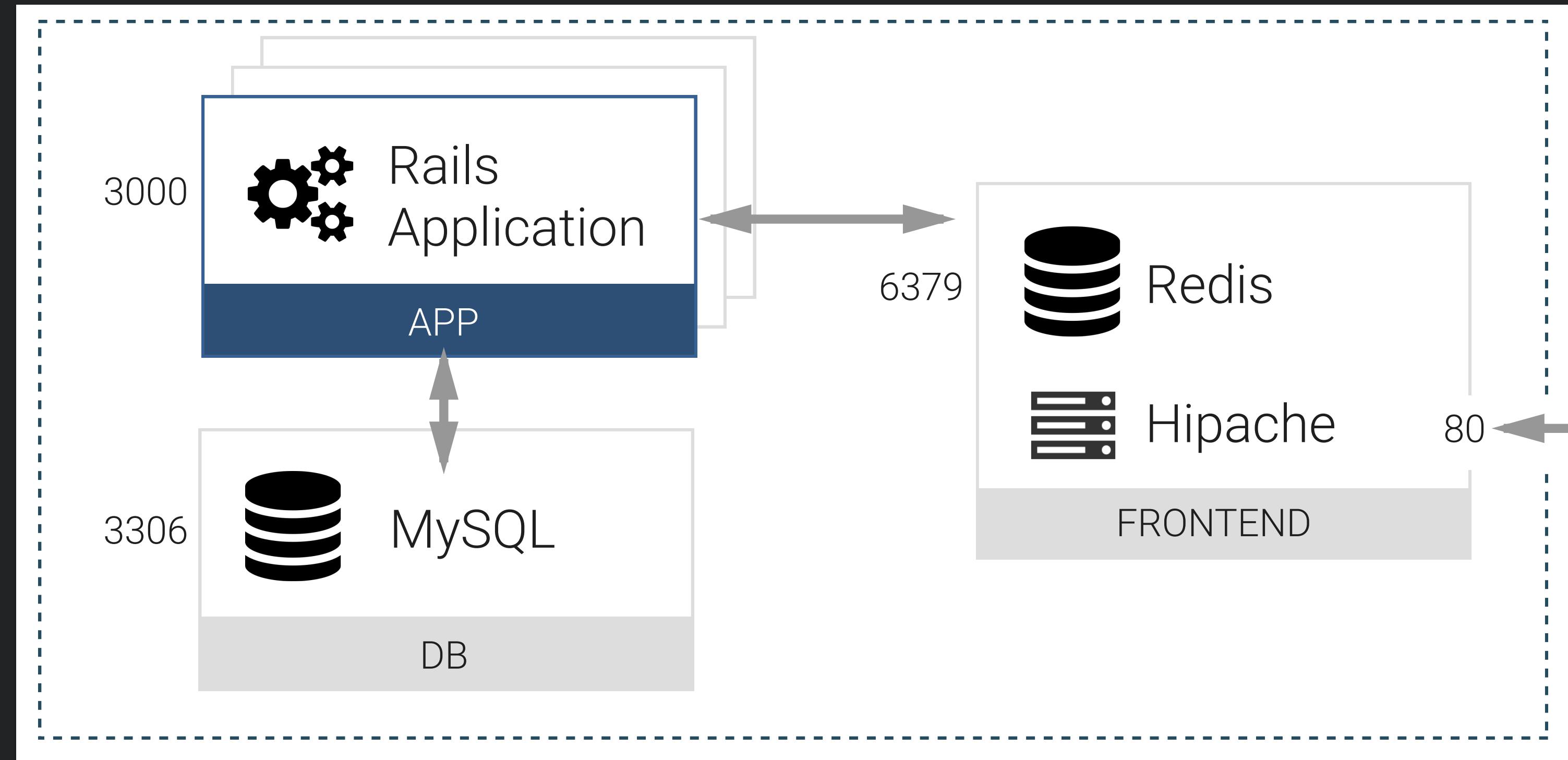


startup script

```
1  frontend:
2    image: oponder/hipache
3    ports:
4      - "80:80"
5      - "6379:6379"
6
7  app:
8    build: .
9    environment:
10   - DOMAIN=local.docker
11  links:
12   - db
13   - frontend
14  volumes:
15   - .:/myapp
16  ports:
17   - '3000'
18
19  db:
20    image: mysql:5.7.8
21    ports:
22    - "3306:3306"
23    volumes:
24    - /var/lib/mysql
25
26    environment:
27      MYSQL_ROOT_PASSWORD: my-secret-pw
```

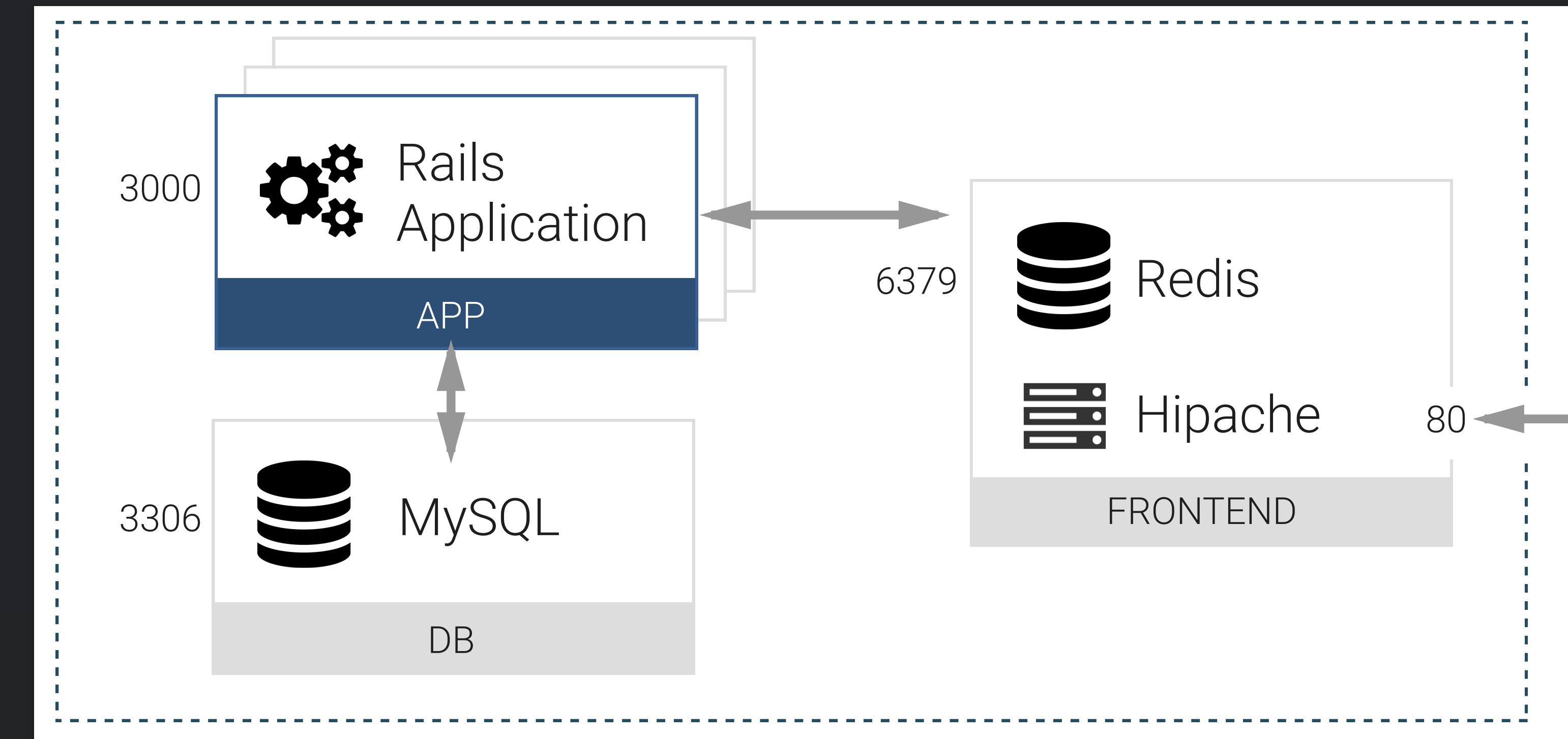
docker-compose.yml

```
1  frontend:  
2    image: oponder/hipache  
3    ports:  
4      - "80:80"  
5      - "6379:6379"  
6  
7  app:  
8    build: .  
9    environment:  
10      - DOMAIN=local.docker  
11    links:  
12      - db  
13      - frontend  
14    volumes:  
15      - ./myapp  
16    ports:  
17      - '3000'  
18  
19  db:  
20    image: mysql:5.7.8  
21    ports:  
22      - "3306:3306"  
23    volumes:  
24      - /var/lib/mysql  
25  
26  environment:  
27      MYSQL_ROOT_PASSWORD: my-secret-pw
```



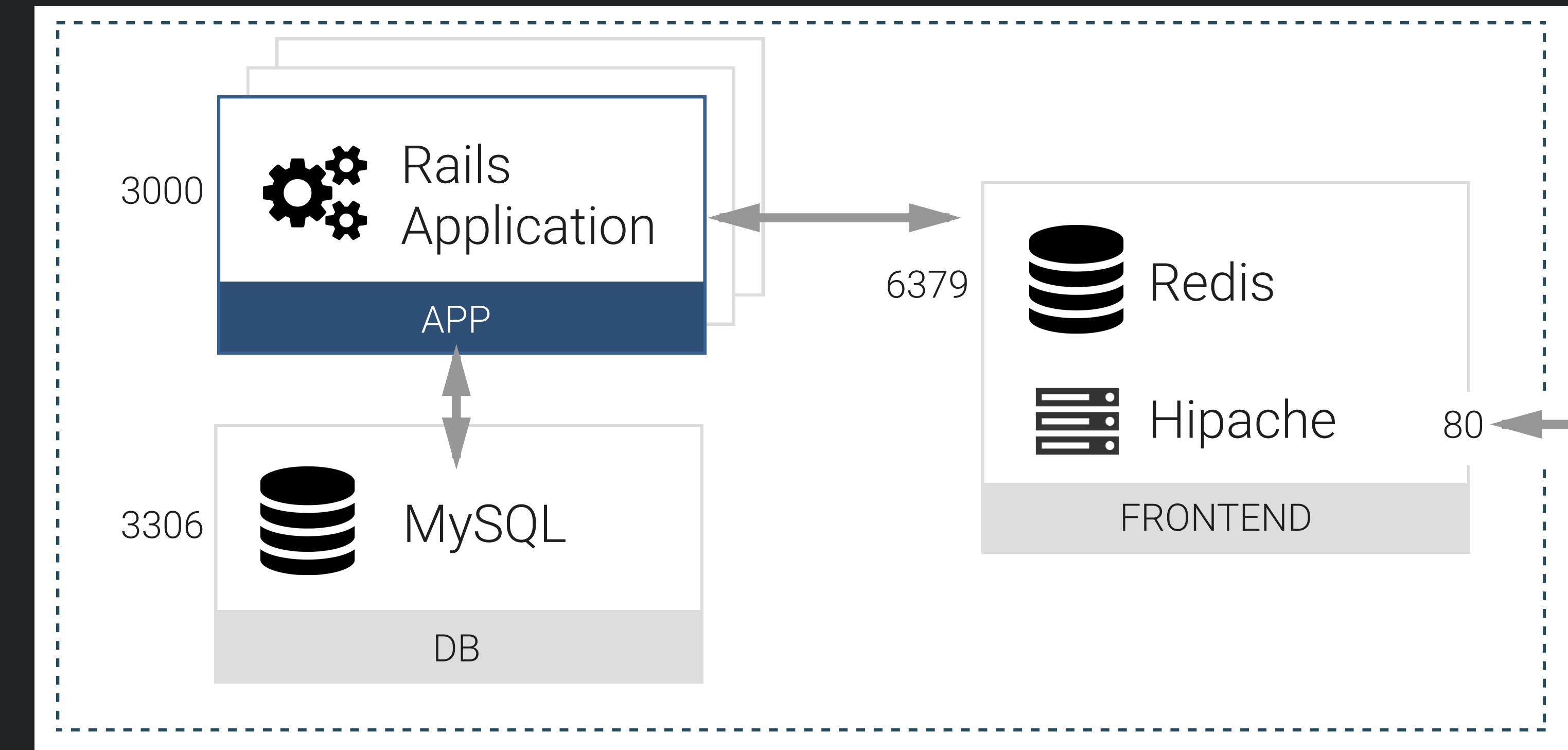
docker-compose.yml

```
1  frontend:  
2    image: oponder/hipache  
3    ports:  
4      - "80:80"  
5      - "6379:6379"  
6  
7  app:  
8    build: .  
9    environment:  
10      - DOMAIN=local.docker  
11    links:  
12      - db  
13      - frontend  
14    volumes:  
15      - ./myapp  
16    ports:  
17      - '3000'  
18  
19  db:  
20    image: mysql:5.7.8  
21    ports:  
22      - "3306:3306"  
23    volumes:  
24      - /var/lib/mysql  
25  
26  environment:  
27      MYSQL_ROOT_PASSWORD: my-secret-pw
```



docker-compose.yml

```
1  frontend:  
2    image: oponder/hipache  
3    ports:  
4      - "80:80"  
5      - "6379:6379"  
6  
7  app:  
8    build: .  
9    environment:  
10      - DOMAIN=local.docker  
11    links:  
12      - db  
13      - frontend  
14    volumes:  
15      - ./myapp  
16    ports:  
17      - '3000'  
18  
19  db:  
20    image: mysql:5.7.8  
21    ports:  
22      - "3306:3306"  
23    volumes:  
24      - /var/lib/mysql  
25  
26  environment:  
27    MYSQL_ROOT_PASSWORD: my-secret-pw
```



docker-compose.yml

```
1  frontend:  
2    image: oponder/hipache  
3    ports:  
4      - "80:80"  
5      - "6379:6379"  
6  
7  app:  
8    build: .  
9    environment:  
10      - DOMAIN=local.docker  
11  links:  
12      - db  
13      - frontend  
14    volumes:  
15      - ./myapp  
16    ports:  
17      - '3000'  
18  
19  db:  
20    image: mysql:5.7.8  
21    ports:  
22      - "3306:3306"  
23    volumes:  
24      - /var/lib/mysql  
25  
26    environment:  
27      MYSQL_ROOT_PASSWORD: my-secret-pw
```

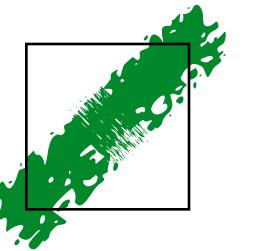
```
$ docker-compose up  
Recreating railsdocker_frontend_1...  
Recreating railsdocker_db_1...  
Creating railsdocker_app_1...  
Building app...
```

docker-compose.yml

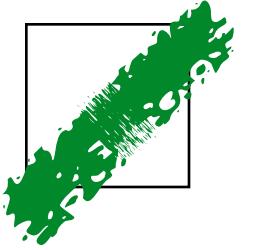
# Containerization Checklist



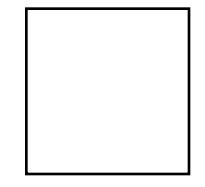
install docker



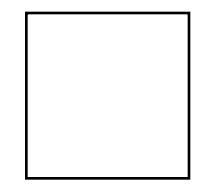
Dockerfile



docker-compose.yml



fix database.yml



startup script

```
1 development:  
2   adapter: mysql2  
3   database: sg_meetup  
4   pool: 5  
5   username: root  
6   host: localhost
```

config/database.yml

```
1 development:
2   adapter: mysql2
3   database: sg_meetup
4   min_messages: WARNING
5   pool: 5
6   username: root
7   password: <%= ENV['RAILSDOCKER_DB_1_ENV_MYSQL_ROOT_PASSWORD'] %>
8   host: <%= ENV['RAILSDOCKER_DB_1_PORT_3306_TCP_ADDR'] %>
9   port: <%= ENV['RAILSDOCKER_DB_1_PORT_3306_TCP_PORT'] %>
```

# Containerization Checklist

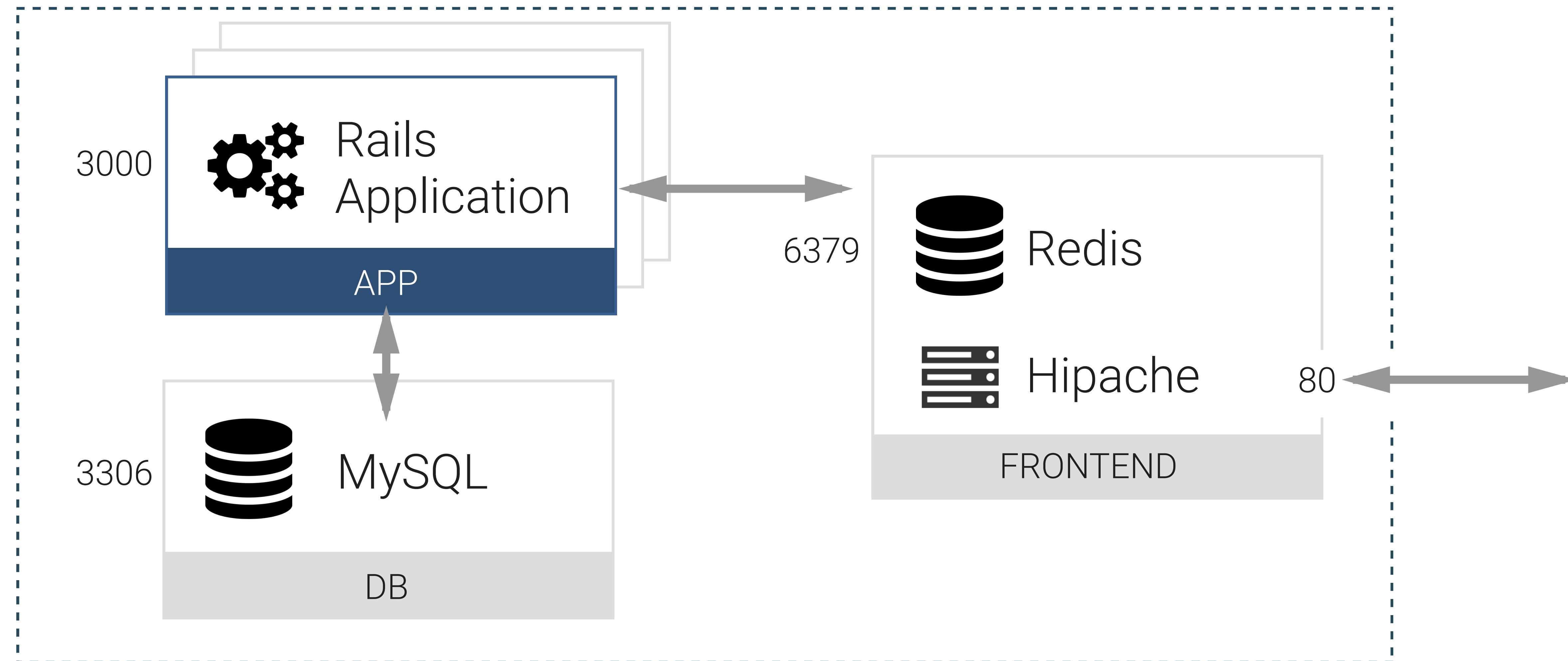
- install docker
- Dockerfile
- docker-compose.yml
- fix database.yml
- startup script

```
1 #!/bin/bash
2
3 trap 'excode=$?; cleanup; echo $excode; exit' EXIT HUP INT QUIT PIPE TERM
4
5 cleanup() {:=}
6 wait_till_port_is_open() {:=}
7 create_loadbalancer_identifier() {:=}
8 create_loadbalancer_entry() {:=}
9
10
11 main() {
12     wait_till_port_is_open frontend 6379
13     wait_till_port_is_open db 3306
14     create_loadbalancer_identifier
15
16     rake db:create db:migrate
17     rails server --port 3000 --binding 0.0.0.0 --pid /server.pid &
18
19     wait_till_port_is_open localhost 3000
20
21     create_loadbalancer_entry
22 }
23
24
25 main
26 wait $1_
```

```
1 #!/bin/bash
2
3 trap 'excode=$?; cleanup; echo $excode; exit' EXIT HUP INT QUIT PIPE TERM
4
5 cleanup() { }
6
7 wait_till_port_is_open() { }
8
9 create_loadbalancer_identifier() { }
10
11 create_loadbalancer_entry() { }
12
13
14 main() {
15     wait_till_port_is_open frontend 6379
16     wait_till_port_is_open db 3306
17     create_loadbalancer_identifier
18
19     rake db:create db:migrate
20     rails server --port 3000 --binding 0.0.0.0 --pid /server.pid &
21
22     wait_till_port_is_open localhost 3000
23
24     create_loadbalancer_entry
25 }
26
27
28 main
29 wait $1_
```

```
1 #!/bin/bash
2
3 trap 'excode=$?; cleanup; echo $excode; exit' EXIT HUP INT QUIT PIPE TERM
4
5 cleanup() {:=}
6
7 wait_till_port_is_open() {:=}
8
9 create_loadbalancer_identifier() {:=}
10
11 create_loadbalancer_entry() {:=}
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29 main() {
30     wait_till_port_is_open frontend 6379
31     wait_till_port_is_open db 3306
32     create_loadbalancer_identifier
33
34     rake db:create db:migrate
35     rails server --port 3000 --binding 0.0.0.0 --pid /server.pid &
36
37     wait_till_port_is_open localhost 3000
38
39     create_loadbalancer_entry
40 }
41
42 main
43 wait $1_
```

```
1 #!/bin/bash
2
3 trap 'excode=$?; cleanup; echo $excode; exit' EXIT HUP INT QUIT PIPE TERM
4
5 cleanup() { }
6
7 wait_till_port_is_open() { }
8
9 create_loadbalancer_identifier() { }
10
11 create_loadbalancer_entry() { }
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29 main() {
30     wait_till_port_is_open frontend 6379
31     wait_till_port_is_open db 3306
32     create_loadbalancer_identifier
33
34     rake db:create db:migrate
35     rails server --port 3000 --binding 0.0.0.0 --pid /server.pid &
36
37     wait_till_port_is_open localhost 3000
38
39     create_loadbalancer_entry
40 }
41
42 main
43 wait $1_
```



# Containerization Checklist

- install docker
- Dockerfile
- docker-compose.yml
- fix database.yml
- startup script

# **Benefits**

Portability across machines

No dependencies besides Docker

Lightweight footprint and minimal overhead

Simplified maintenance

# Feedback please!

<https://goo.gl/m7v0uk>

**Presentation feedback**

Zero Downtime Rails Deployments

Thanks for watching my presentation, I'd appreciate any feedback you have!

**\* 1. Rating questions**

	Strongly Disagree	Disagree	Neutral	Agree	Strongly agree
The presentation was easy to follow	<input type="radio"/>				
The transitions were smooth and seemed natural	<input type="radio"/>				
The topic was relevant	<input type="radio"/>				
The presenter understood the topic well	<input type="radio"/>				
The slides were interesting					