

From development to production with Docker



docker



Pablo Chico de Guzmán
Docker Inc
@chico_de_guzman

Ramiro Berrelleza
Atlassian
@rberrelleza

Docker Fundamentals

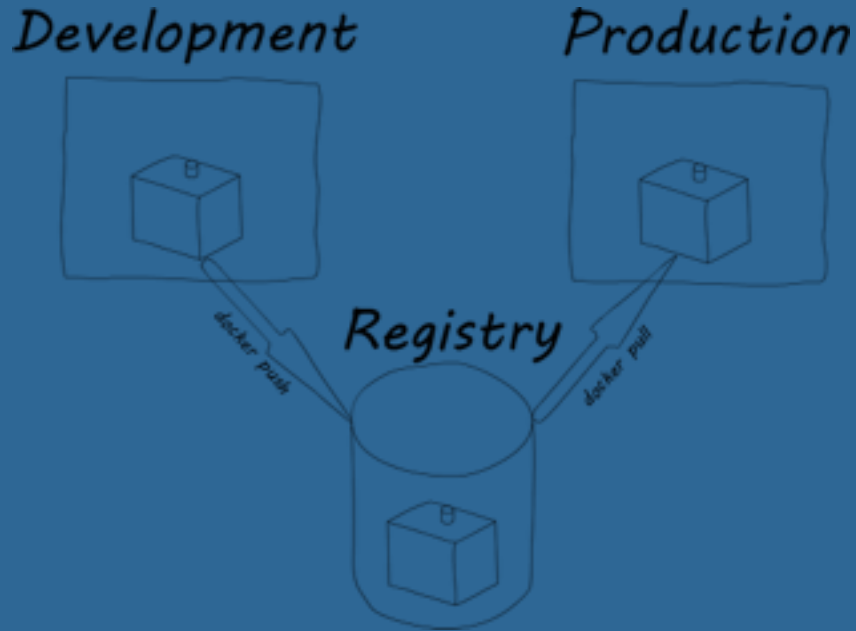
What are containers?

Isolated processes

Includes all dependencies

Easy to distribute

Build, ship and run any app, anywhere



Build

Dockerfile

Multi-stage:

- small images

- reproducible builds

Cache

Local Environment

web:

```
build: .
```

```
volumes:
```

```
- ./src
```

```
ports:
```

```
- "80:80"
```

redis:

```
image: redis
```

Continuous Integration

```
tests:  
  build: .  
  command: ./run_tests.sh
```

```
redis:  
  image: redis
```


Taking Docker to production

Prepare your dependencies

Create an standard build process for your containers

Build on CI, don't trust the developer machine

Tag your images

Have your own private repository

Start small

Create a docker-based dev environments as a way to get familiar with docker

Move some of your dev tooling to docker

Prepare your code base before heading into production

Prepare for deployments

Don't use kubernetes or mesos

Decide how to handle secrets

Decide how to handle your different environments

Prepare for deployments

Validate your processes with branched or test deployments

Have a strategy for versioning and rollbacks

Logging

Have scripts to duplicate deployments locally

To production

Deploy a single non-critical service to docker.

Get used to rolling back versions

Ramp up as you get more familiar



No one cares that you're running on containers.
Your customers get value out of your application, not the deployment model

Q & A