ThoughtWorks®

RubyConf São Paulo 2016

WHY USE DOCKER AND COMPOSE IN YOUR CI?

Carla Suárez
@carlast22

What are we going to talk about?

- CD and CI?
- Docker and Compose
  - What are them?
  - Benefits
  - Difference with other techs
- Recommendations

Hi! / Oi!

Journalist

Developer at ThoughtWorks for 2 years and a half

Python, Ruby, HTML, CSS

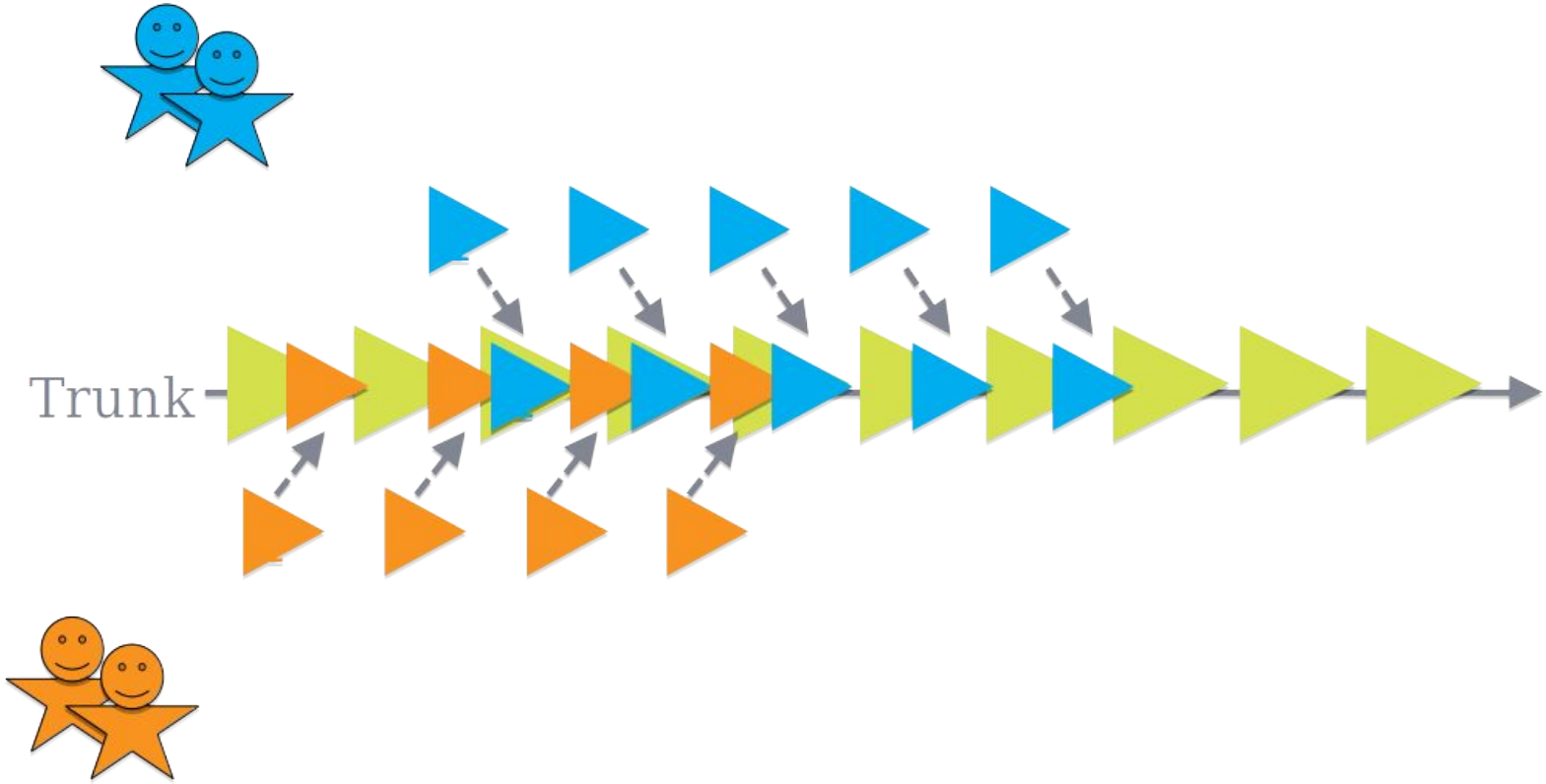
Learning about Devops

ThoughtWorks®

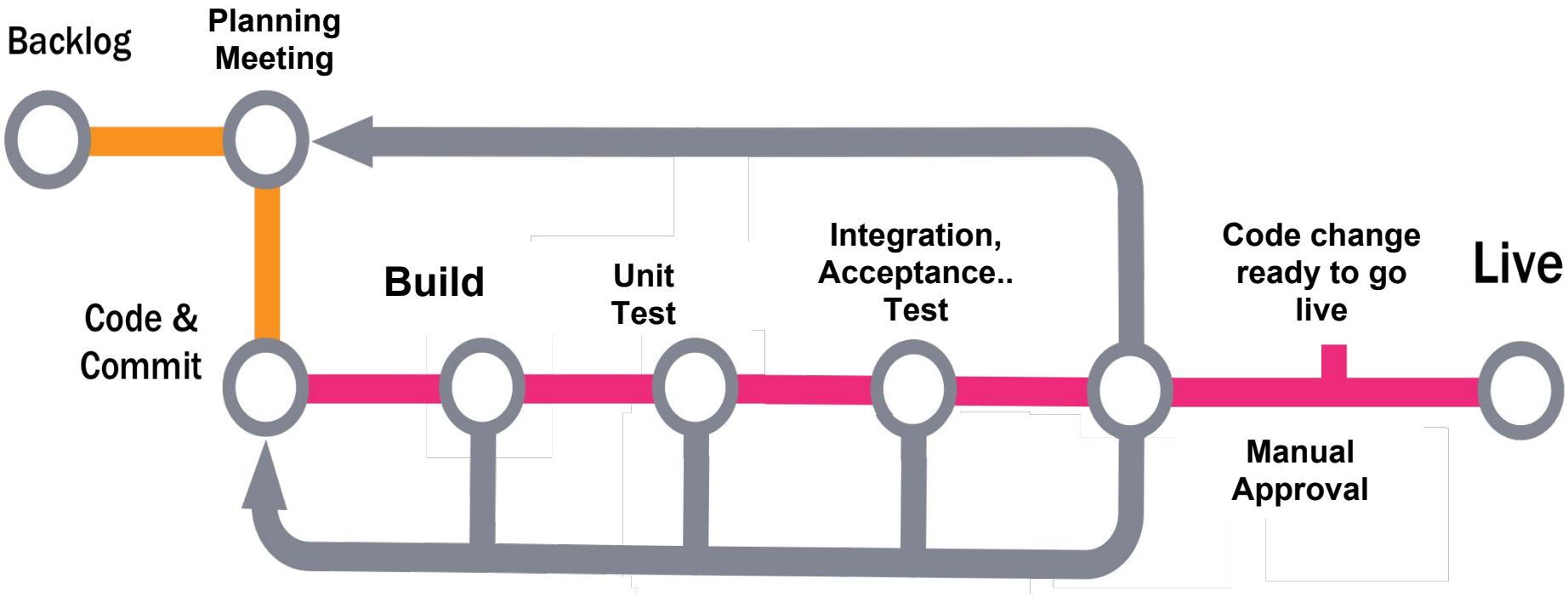
CI and CD?

What is this? Concepts and importance in deployment

Trunk based development



Here is where CI go into scene



The practices

- Automate the build
- Every commit should build on an integration machine
- Test in a clone of the production environment
- Make it easy for anyone to get the latest executable
- Automate deployment

Continuous Deployment is closely related to Continuous Integration and refers to the release into production of software that passes the automated tests.

What is CD?

“Continuous Delivery is a software development discipline where you build software in such a way that the software can be released to production at any time”

- Martin Fowler.

Main principles and practices

- Create a repeatable, reliable way to release software
- Everybody is responsible for the delivery process
- Automate almost everything

ThoughtWorks®

Docker and Compose

What are and how can it help you in CD?

Docker

- Docker is an open-source platform to package and distribute applications quickly
- It create lightweight and portable containers to run on any machine.
- It is independent of the Linux distribution of the host machine

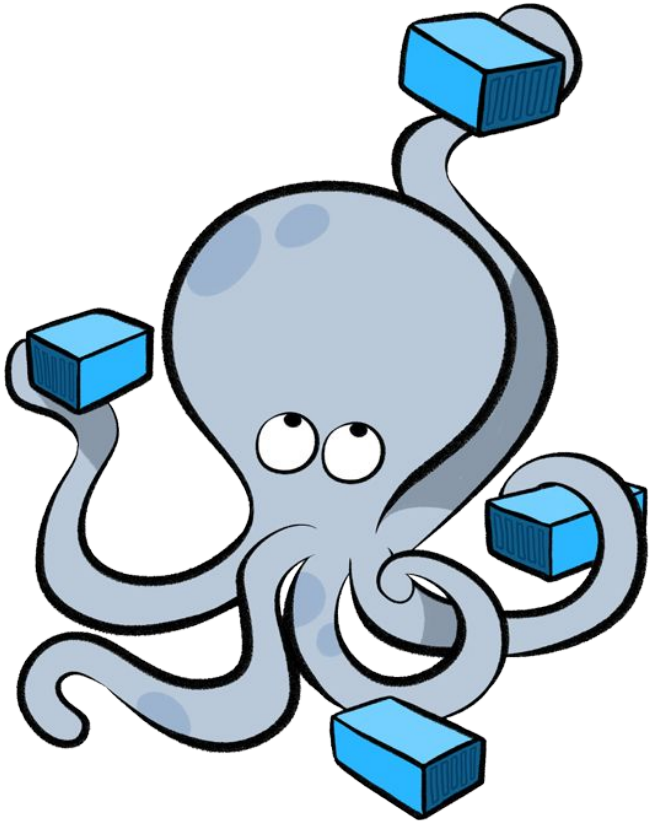
Dockerfile

```
FROM ruby:2.3-alpine

WORKDIR /app
COPY Gemfile /app
RUN bundle install
COPY . /app
EXPOSE 9292

CMD ["rackup", "--host", "0.0.0.0"]

```

Gemfile

```
source 'https://rubygems.org'

gem 'cuba', '~> 3.8'
gem 'tilt', '~> 2.0', '>= 2.0.5'
gem 'redis', '~>3.2'
```

Docker Compose

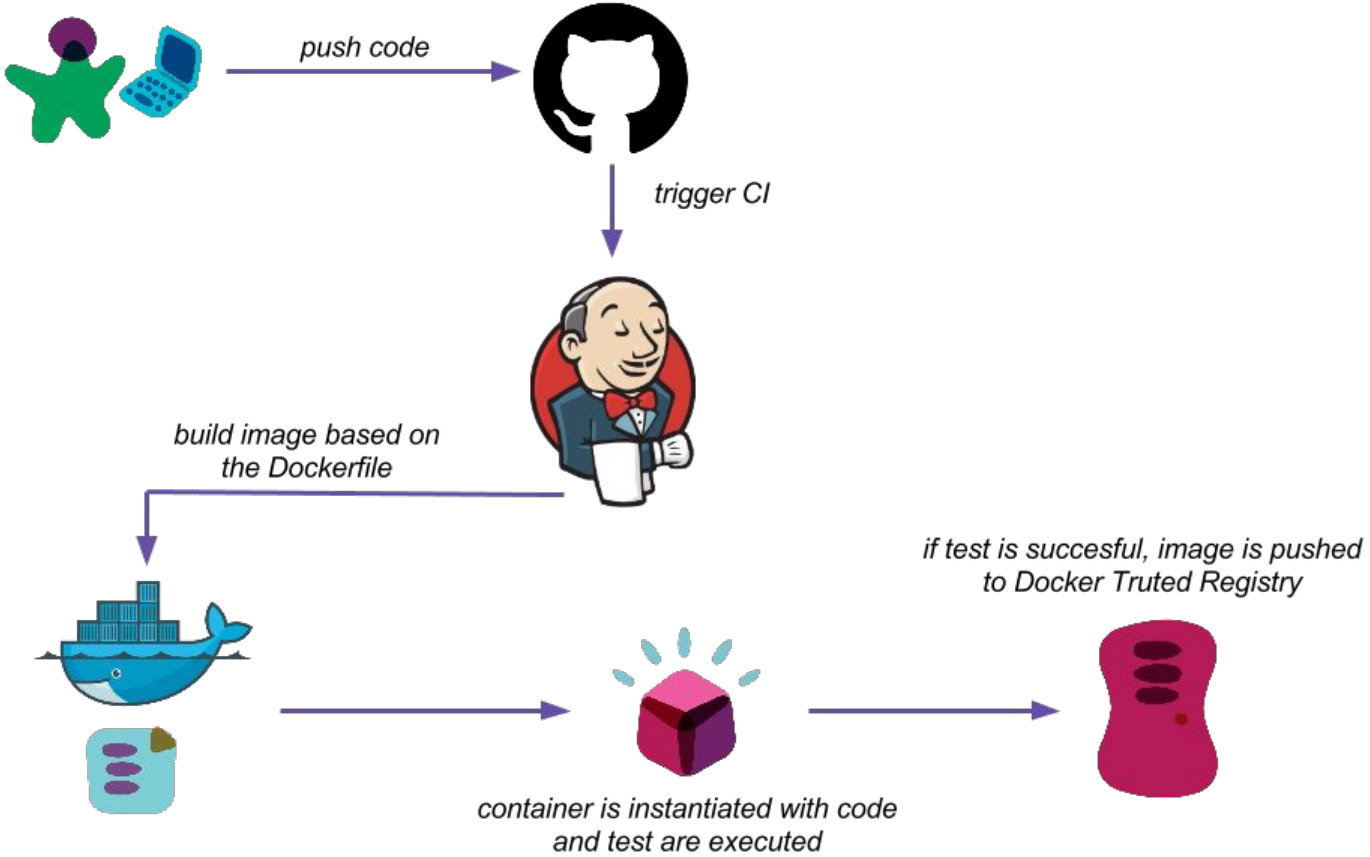


- Compose is a tool for the definition and implementation of multi-Docker container applications
- Through a configuration file and a single command, create and start all predefined services

docker-compose.yml

```yaml
version: '2'

services:
  app:
    build: .
    ports:
     - "9292:9292"
    links:
     - redis
  db:
    image: redis

```

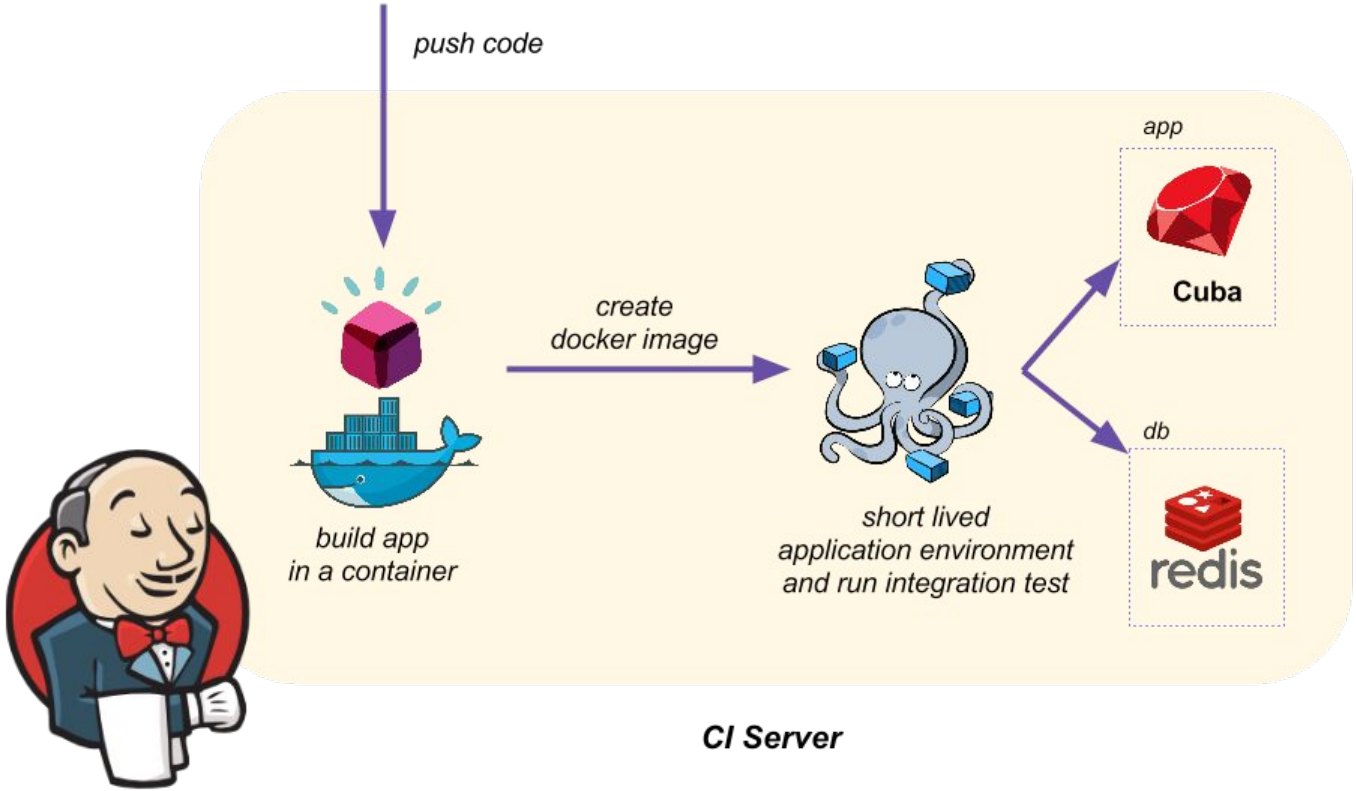
CI Workflow


push code
trigger CI
build image based on the Dockerfile
if test is succesful, image is pushed to Docker Truted Registry
container is instantiated with code and test are executed

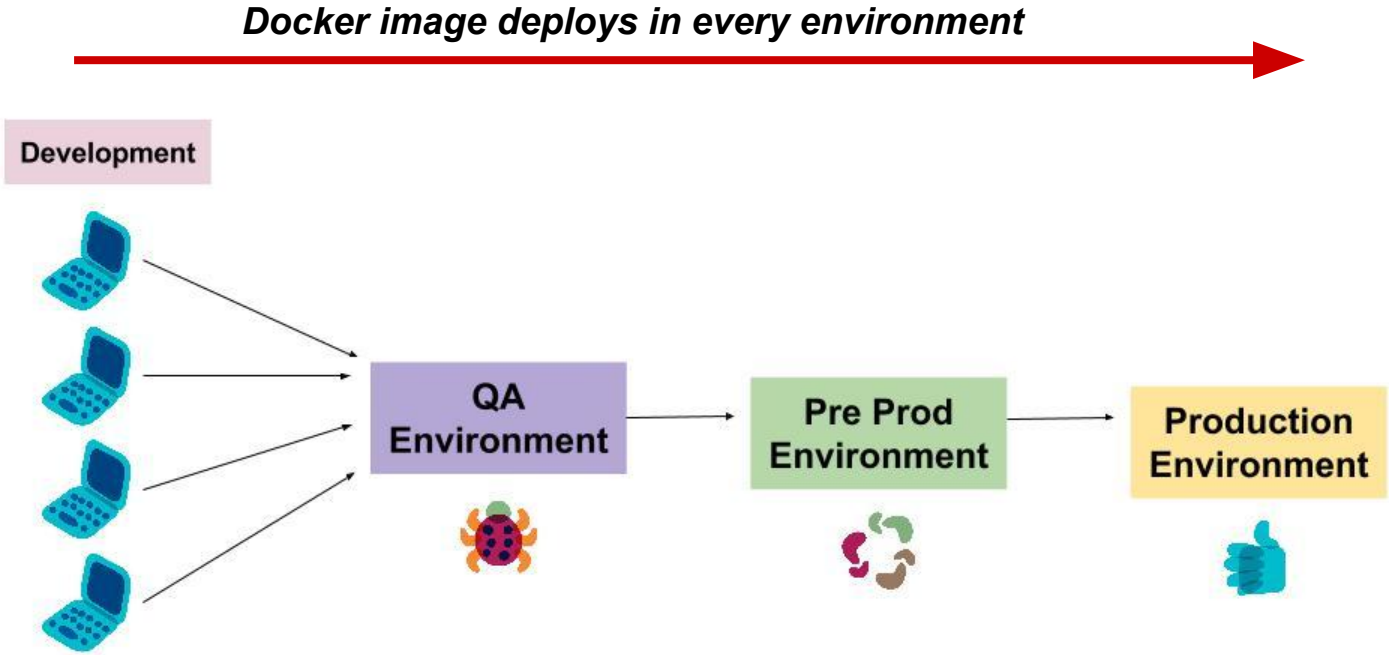
Registry

Is a server side application that stores and lets you distribute Docker images.

And Docker Compose?



The same in different environments



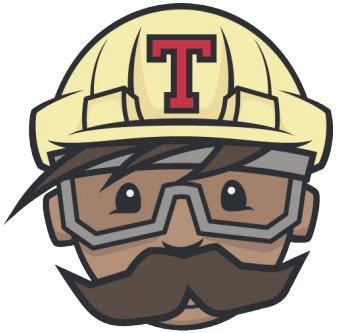
Using docker and docker compose

- Deploy all your application needs
- Automated testing
- Dedicated container for each application
- Reduce time and costs
- Stability and resilience

VMs vs. Containers

- VM is a completely virtualized environment that only abstracts the physical hardware.
- VM comes with its own BIOS, virtualized network adapters, disk storage, CPU and a complete operating system.
- Container abstraction happens at the operating system level.
- Each user shares the same operating system, kernel instance, network connection and base file system, each instance of the application will run within a separate user space.

Supports docker



snap

circleci

go®

CODESHIP

ThoughtWorks®

Recommendations

To think before choosing a technology

- Is it the best for your project requirements?
- Spike before choosing.
- Do CD, if you are not doing it yet.

Thanks/Obrigada

For feedback or questions:

csuarez@thoughtworks.com

@carlast22

ThoughtWorks®