Online Workshop

Docker & Kubernetes

Session #3 | @inicopaez

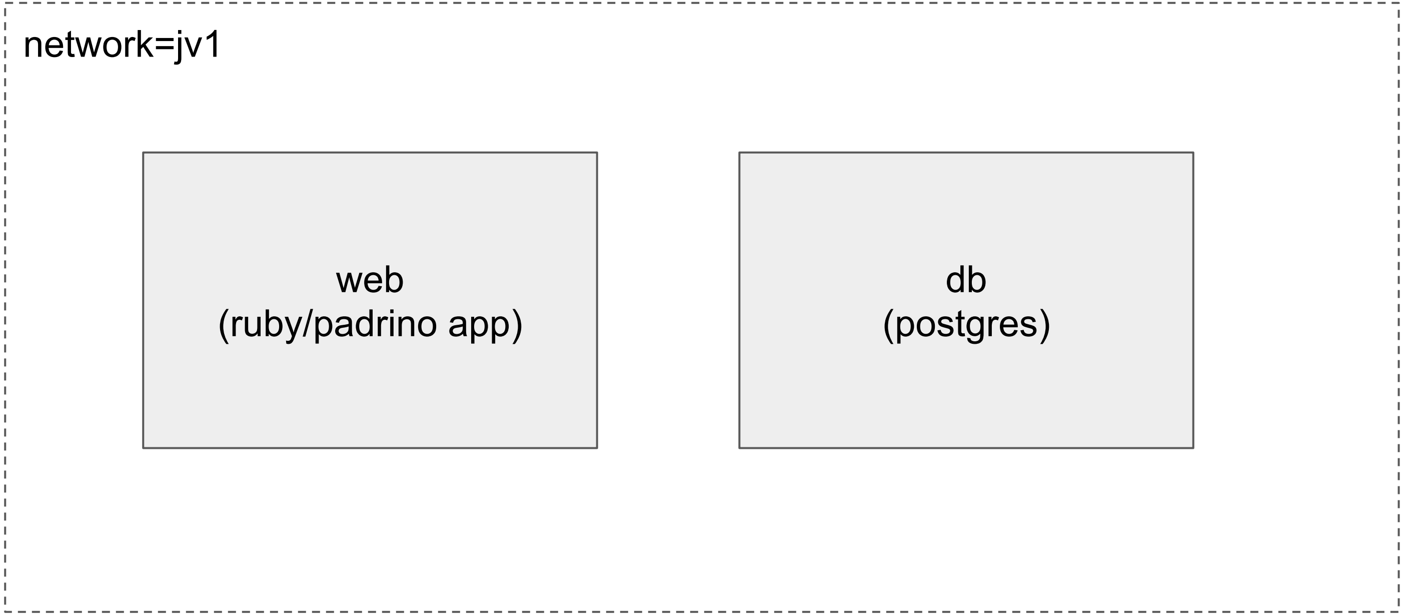
Session 3 Agenda

- Working with multiple containers
- Docker-compose fundamentals
- Docker-compose use cases

Homework recap

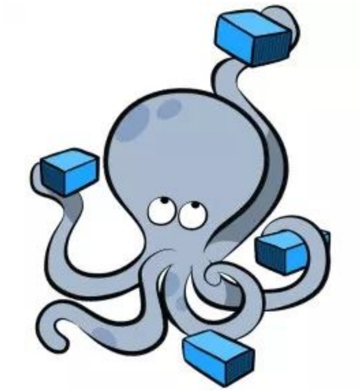
Ejercicio imagen RedHat

Job Vacancy Application

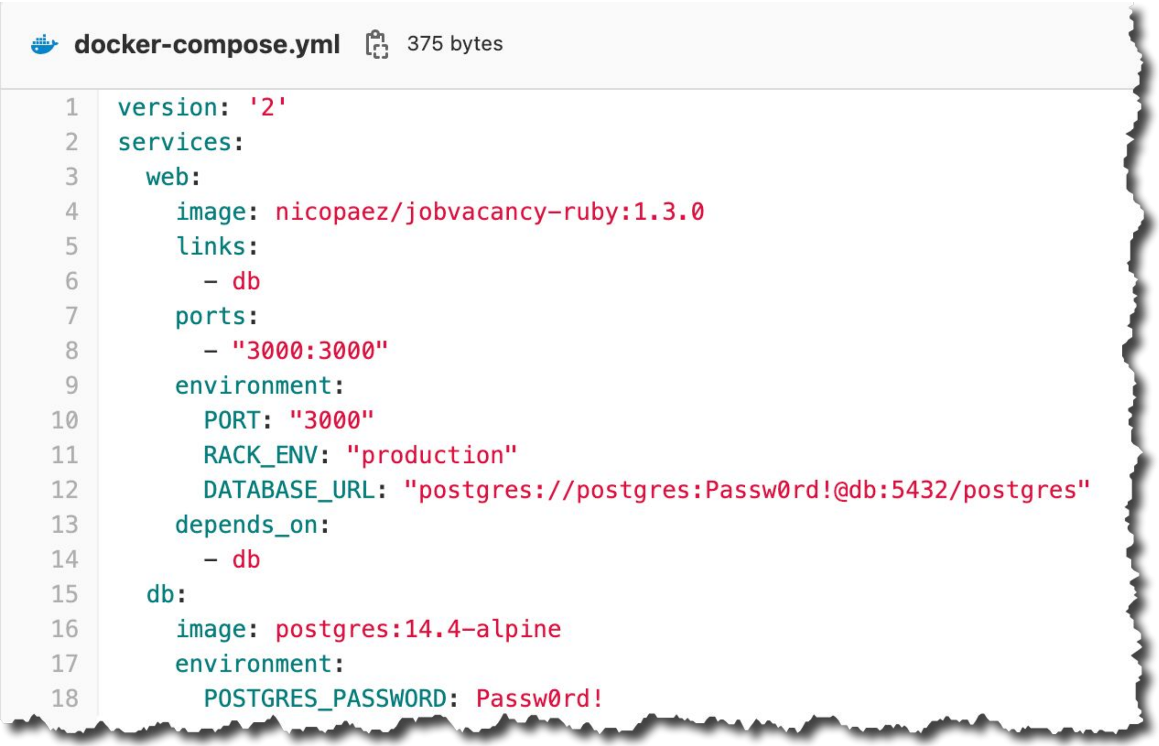


https://gitlab.com/-/snippets/2376003/raw/main/plain-docker.sh

Docker Compose



Job Vacancy on docker-compose



docker-compose.yml 375 bytes

```yaml
version: '2'
services:
  web:
    image: nicopaez/jobvacancy-ruby:1.3.0
    links:
      - db
    ports:
      - "3000:3000"
    environment:
      PORT: "3000"
      RACK_ENV: "production"
      DATABASE_URL: "postgres://postgres:Passw0rd!@db:5432/postgres"
    depends_on:
      - db
  db:
    image: postgres:14.4-alpine
    environment:
      POSTGRES_PASSWORD: Passw0rd!
```

https://gitlab.com/-/snippets/2376003/raw/main/docker-compose.yml `docker-compose`

vs.

`docker compose`

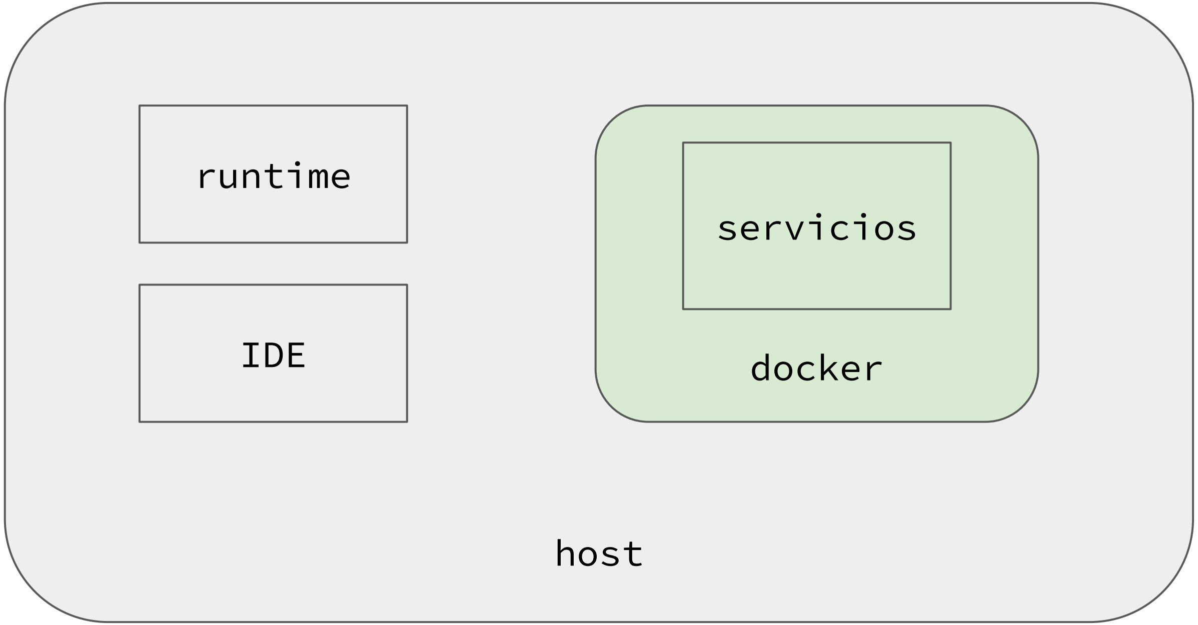
https://github.com/docker/awesome-compose

Compose use cases

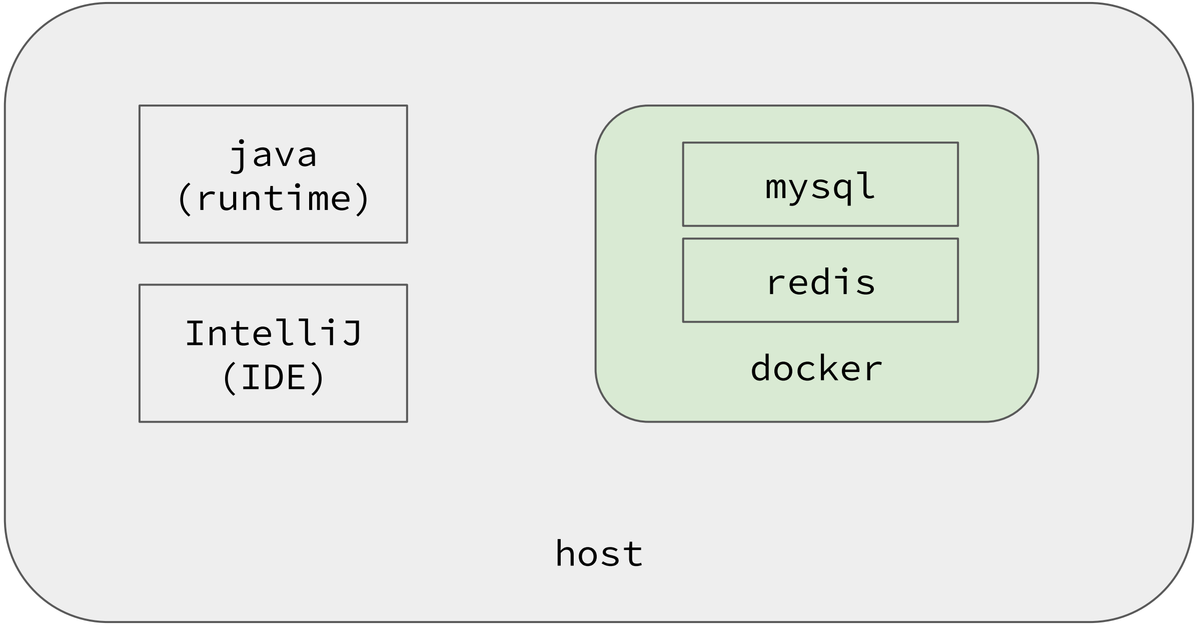
- Development: clone & run experience
- Testing: end-to-edge testing

clone & run

Development docker-compose 1



Development con docker-compose 1



java
(runtime)

IntelliJ
(IDE)

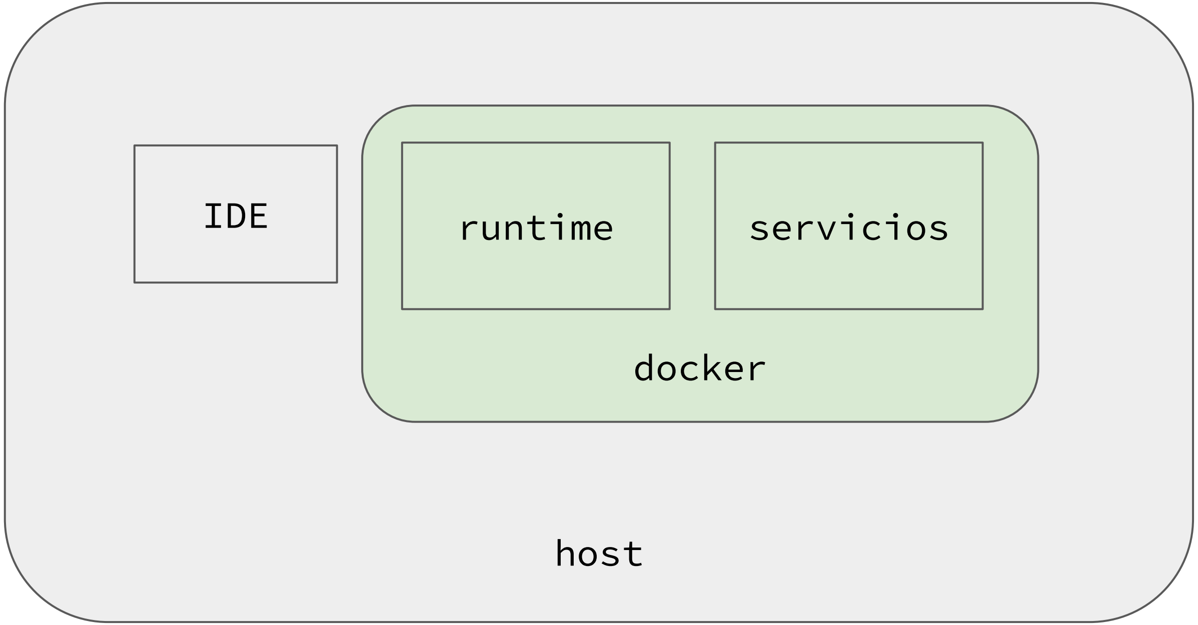
mysql

redis

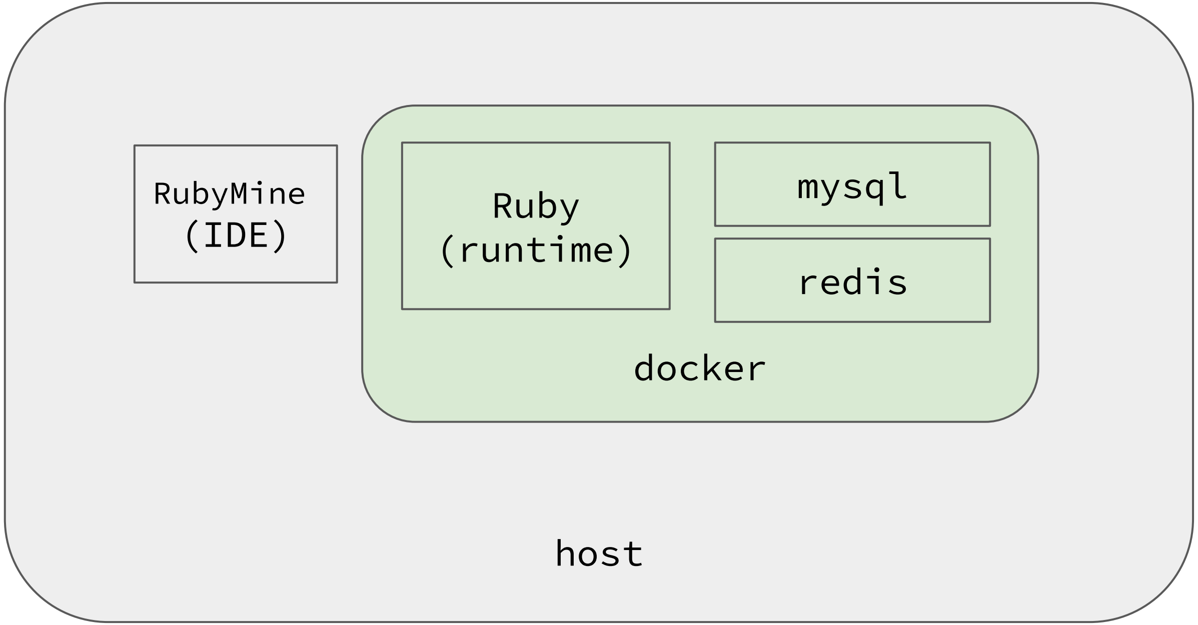
docker

host

Development docker-compose 2



Development docker-compose 2



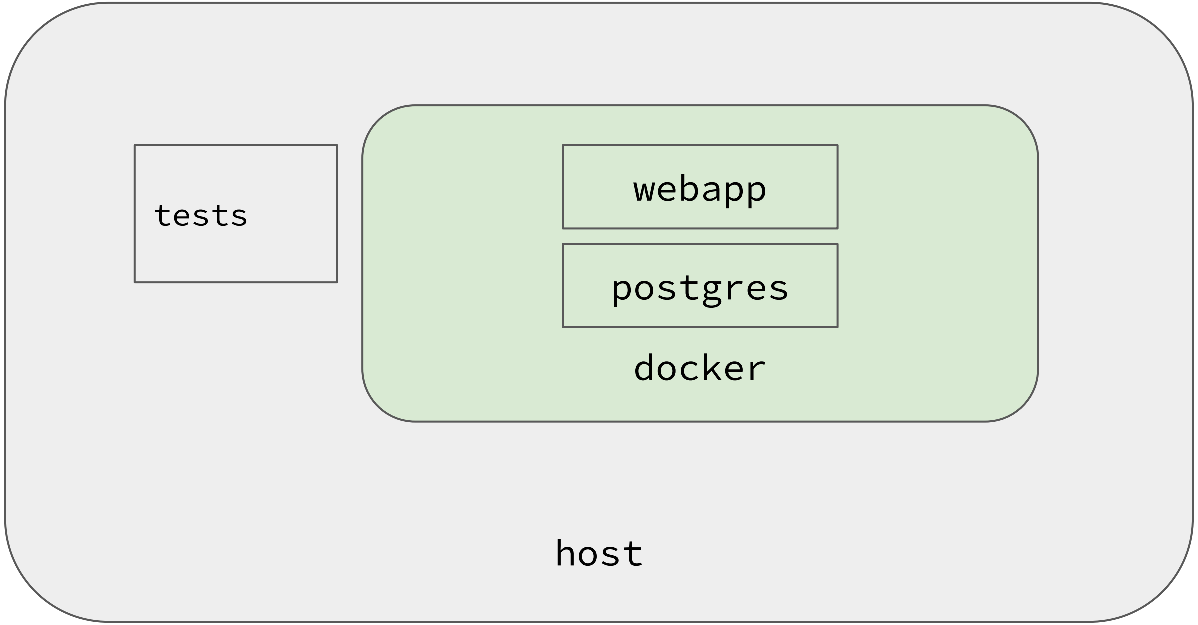
Docker-compose scenarios

https://github.com/nicopaez/webapi-example

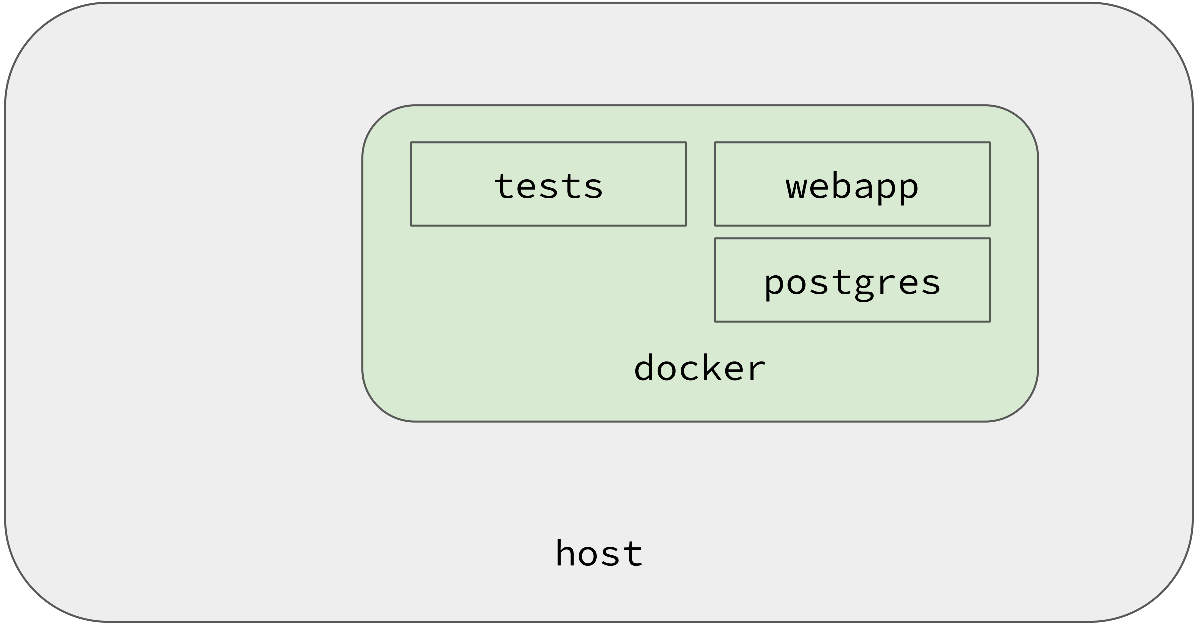
```
docker-compose up -d
```

```
docker-compose exec webapp /bin/bash
```

Testing end-2-edge



Testing end-2-edge



Demo: e2e testing sin ambiente

Demo: HomeBanking

Docker: Production challenges

Single Point of failure / Availability

Management cost

Restricted Scalability

End of Session 3