



# Docker Swarm

Infraestrutura clusterizada para micro e pequenas empresas

# Agenda

- Sobre mim;
- docker-compose up **vs** docker stack deploy;
- Escalabilidade e clusterização;
- Proposta de arquitetura para micro e pequenas empresas;
- Conclusão;

Sobre mim

docker-compose up

**vs**

docker stack deploy

```
version: "3.0" # docker-compose up
services:
  db:
    image: postgres:12.2
    environment:
      - POSTGRES_PASSWORD=ej
      - POSTGRES_USER=ej
      - POSTGRES_DB=ej
    volumes:
      - "backups:/var/lib/postgresql/data"
  server:
    container_name: server
    build:
      context: ../
      dockerfile: docker/Dockerfile
    volumes:
      - ../ej-server/
    env_file:
      - variables.env
    ports:
      - 8000:8000
    entrypoint: ["/bin/bash", "docker/run_local_server.sh"]
  backups:
```

```
db:
  image: postgres:12.2
  environment:
    - POSTGRES_PASSWORD=ej
    - POSTGRES_USER=ej
    - POSTGRES_DB=ej
  volumes:
    - "backups:/var/lib/postgresql/data"
```

```
server:
  container_name: server
  build:
    context: ../
    dockerfile: docker/Dockerfile
  volumes:
    - ../ej-server/
  env_file:
    - variables.env
  ports:
    - 8000:8000
  entrypoint: ["/bin/bash", "docker/run_local_server.sh"]
```

**docker-compose up -f compose.yml**

```
version: "3.0" # docker stack deploy
services:
  db:
    image: postgres:12.2
    environment:
      - POSTGRES_PASSWORD=ej
      - POSTGRES_USER=ej
      - POSTGRES_DB=ej
    volumes:
      - "backups:/var/lib/postgresql/data"
  server:
    image: https://registryprivado.com.br/ej-server:latest
    volumes:
      - code:/ej-server/
    env_file:
      - variables.env
    ports:
      - 8000:8000
    entrypoint: ["/bin/bash", "docker/run_local_server.sh"]
volumes:
  backups:
  code:
```

```
server:
  image: https://registryprivado.com.br/ej-server:latest
  volumes:
    - code:/ej-server/
  env_file:
    - variables.env
  ports:
    - 8000:8000
  entrypoint: ["/bin/bash", "docker/run local server.sh"]
```

1. **docker swarm init**
2. **docker stack deploy -f docker-stack.yml**

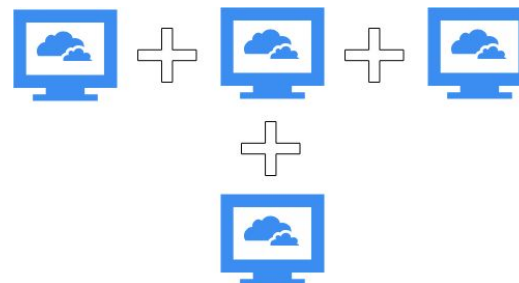
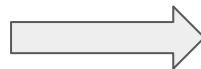
Escalabilidade e  
clusterização

Escalabilidade  
vertical

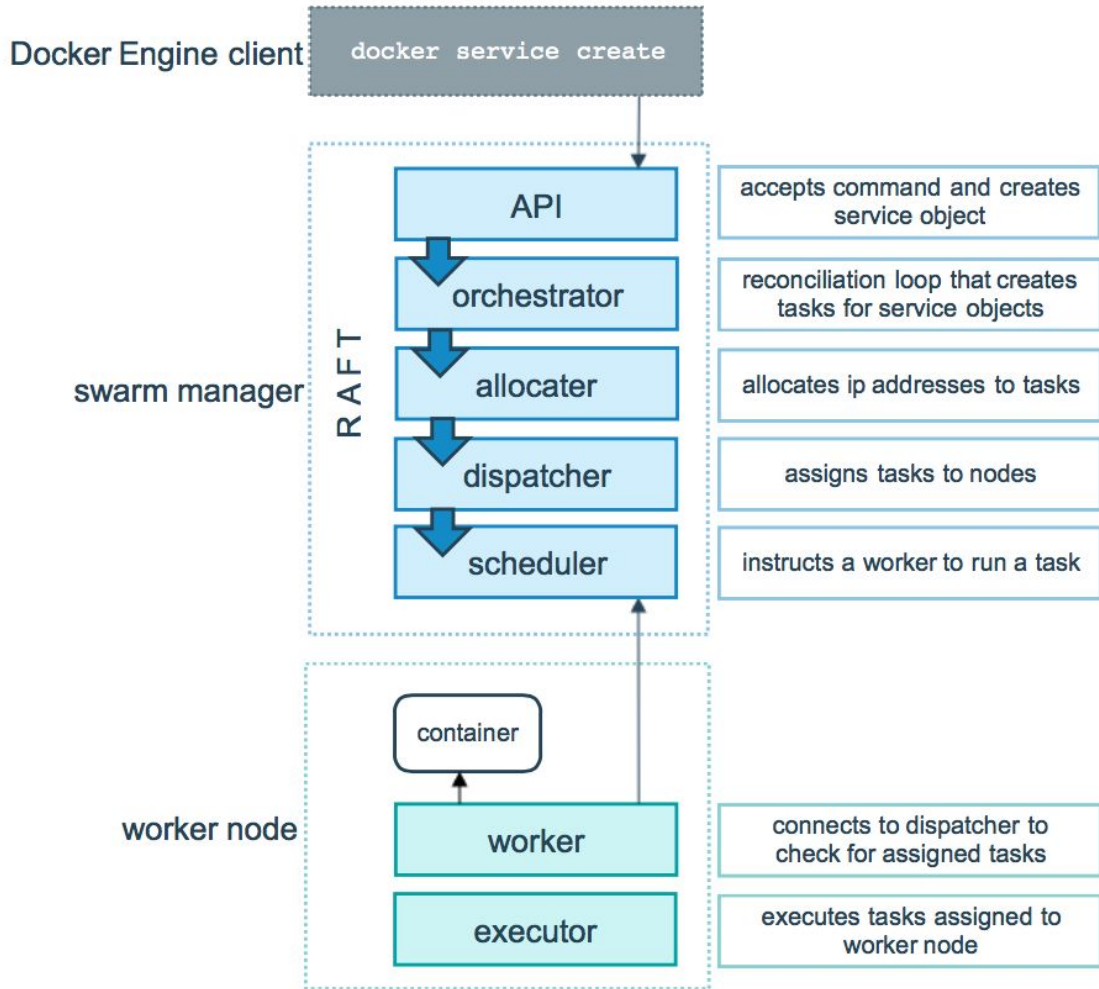


**VS**

Escalabilidade  
horizontal







## Swarm Manager

1. Manter o estado do cluster
2. Agendamento de serviços
3. Servir a api http do swarm

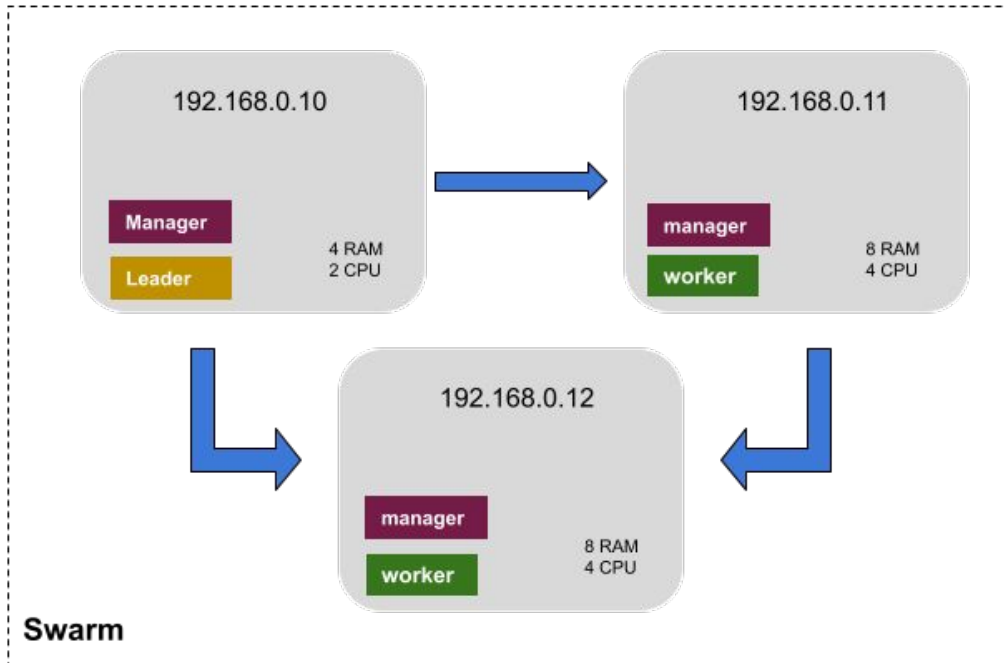
## Swarm Worker

1. executar as tarefas agendadas pelos managers

Proposta de arquitetura  
para micro e pequenas  
empresas

	<b>docker- compose up</b>	<b>docker stack deploy</b>
permite criar serviços	sim	sim
permite criar volumes do tipo bind	sim	não
permite criar volumes nomeados	sim	sim
permite comunicação entre os serviços	sim	sim
permite distribuir serviços em mais de uma máquina	não	sim
permite escalar a aplicação em mais de uma máquina	não	sim
permite fazer balanceamento de carga dos serviços	não	sim
permite gerenciar diferentes máquinas virtuais de forma unificada	não	sim

Imaginemos uma aplicação composta por uma API feita em Django, frontend feito em React e banco de dados postgresql.



- Um cluster com N managers tolera a perda de no máximo  $(N-1)/2$  managers.

### Cenário 1

API rodando na máquina 192.168.0.11, banco de dados e frontend rodando na máquina 192.168.0.12

### Cenário 2

API rodando de forma replicada nas máquinas 192.168.0.11 e 192.168.0.12, banco de dados e frontend rodando na máquina 192.168.0.12

### Cenário 3 - Escala Horizontal

API rodando de forma replicada nas máquinas 192.168.0.11 e 192.168.0.12, banco de dados e frontend rodando na máquina 192.168.0.12

Dashboard

Registries

APPLICATIONS

Stacks

Services

Tasks

INFRASTRUCTURE

Networks

Nodes

DATA

Volumes

Secrets

Configs

Total (5)

SHOW FILTERS

gitlab-runner  
173.255.192.24

docker 20.10.3

READY MANAGER DRAIN



ip-172-31-6-251  
18.222.20.172

docker 19.03.12

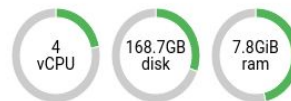
READY WORKER ACTIVE



main-server  
72.14.178.49

docker 19.03.12

READY LEADER MANAGER ACTIVE



rasa-dev  
172.104.193.76

docker 20.10.3

READY MANAGER ACTIVE



worker1  
23.239.28.249

docker 19.03.12

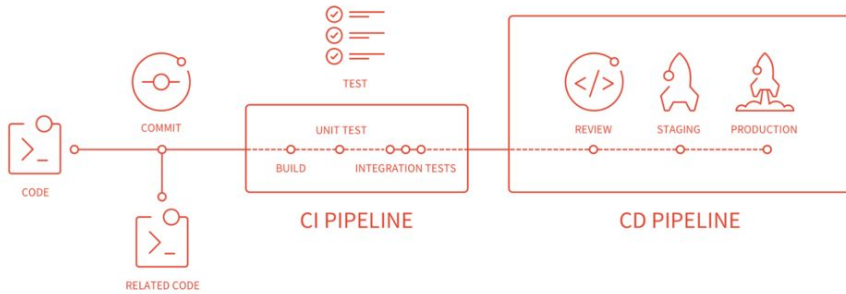
READY WORKER ACTIVE



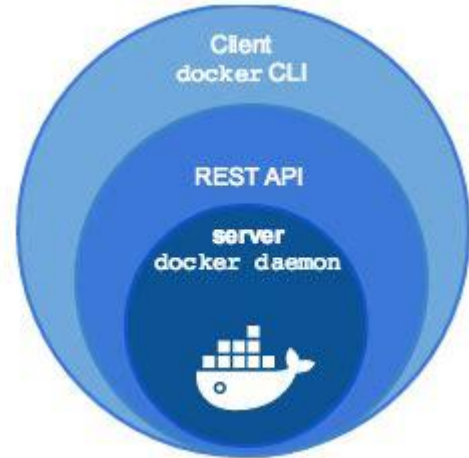
Deploy contínuo  
e  
monitoramento



# GitLab CI



# ZABBIX





# Obrigado!



@mr\_davidcarlos



@mr\_davidcarlos

<https://pencillabs.com.br/>