DOCKER E DOCKER-COMPOSSE PARTE 3

Douglas Nassif Roma Junior

/douglasjunior

/in/douglasjunior

smarppy.com

douglas@smarppy.com

Slides: https://git.io/fpgYO

AGENDA

- Services
- Seu primeiro `docker-composse.yml`
- Execute seu balanceamento de carga
- Orquestrando os seus containers
- Referências

SERVICES

SERVICES

- Em uma aplicação distribuída, diferentes partes são chamadas de "serviços" (ou *services*).

- Por exemplo, em um serviço de compartilhamento de vídeos você provavelmente possui:
  - Serviço para armazenar os dados da aplicação em um banco de dados.
  - Serviço para decodificação de vídeo em segundo plano após o *upload*.
  - Serviço para o front-end.
  - E assim por diante...

SERVICES

- No **Docker**, os serviços são representados por **Containers** em execução.
- Um **Serviço** roda apenas uma **Imagem**, e também define a maneira como a imagem é executada: quais portas devem ser usadas, quantas réplicas do container devem ser executadas, e assim por diante.
- O dimensionamento de um serviço altera o número de instâncias de containers que executam a aplicação, atribuindo mais recursos computacionais ao serviço em execução.
- Felizmente, é muito fácil de definir, executar e escalar serviços com o Docker, apenas é necessário criar um `docker-composse.yml`.

SEU PRIMEIRO DOCKER-COMPOSSE.YML

SEU PRIMEIRO DOCKER-COMPOSSE.YML

- O arquivo `docker-composse.yml` é do tipo YAML, e define como os containers do Docker devem se comportar em execução.

- Crie um arquivo chamado `docker-composse.yml` em qualquer diretório desejado, e copie o conteúdo do link à seguir:

- https://gist.github.com/douglasjunior/08550396924348c0bb02277ee44f4cdf

SEU PRIMEIRO DOCKER-COMPOSSE.YML

- O arquivo que você criou contem as seguintes configurações:
  - Baixar a imagem publicada na aula anterior à partir do "registry" público.
  - Executar 5 instâncias desta imagem como um **serviço** chamado **web**, limitando cada uma para ocupar no máximo 10% do CPU e 50MB de RAM.
  - Reiniciar o container imediatamente após qualquer falha.
  - Mapear a porta 4000 da máquina para a porta 80 do container.
  - Configura o serviço **web** para compartilhar a porta 80 através de uma rede de "balanceamento de carga" (load-balancer), chamada **webnet**.
  - Define a rede **webnet** com as configurações padrões. (que é uma rede de sobreposição com balanceamento de carga)

EXECUTE SEU BALANCEAMENTO DE CARGA

EXECUTE SEU BALANCEAMENTO DE CARGA

- Antes de executar nossa aplicação, primeiro precisamos iniciar o Docker Swarm*.

```
$ docker swarm init
```

- Então, basta executar o Docker Stack**:

```
$ docker stack deploy -c docker-compose.yml meuapp-balanceado
```

- Para consultar os serviços em execução:

```
$ docker service ls
```

- Para consultar os containers:

```
$ docker container ls
```

* https://docs.docker.com/engine/swarm/
** https://www.mundodocker.com.br/tag/docker-stack/

EXECUTE SEU BALANCEAMENTO DE CARGA

- **Para remover o serviço completamente:**

```
$ docker stack rm meuapp-balanceado
```

- **Para interromper o Swarm.**

```
$ docker swarm leave --force
```

ORQUESTRANDO OS SEUS CONTAINERS

ORQUESTRANDO OS SEUS CONTAINERS

- Uma etapa faltante em nosso projeto é a integração do container da aplicação com o Redis para que o contador de visitas funcione adequadamente.

- Para isso, você pode adicionar a definição de um serviço extra em seu `docker-compose.yml`:

https://gist.github.com/douglasjunior/2771da8e9f7f29c0365a45df3af38ce7

ORQUESTRANDO OS SEUS CONTAINERS

- Novamente, antes de executar nossa aplicação precisamos iniciar o Docker Swarm.

```
$ docker swarm init
```

- Então, basta executar o Docker Stack:

```
$ docker stack deploy -c docker-compose.yml meuapp-balanceado
```

- Para consultar os serviços em execução:

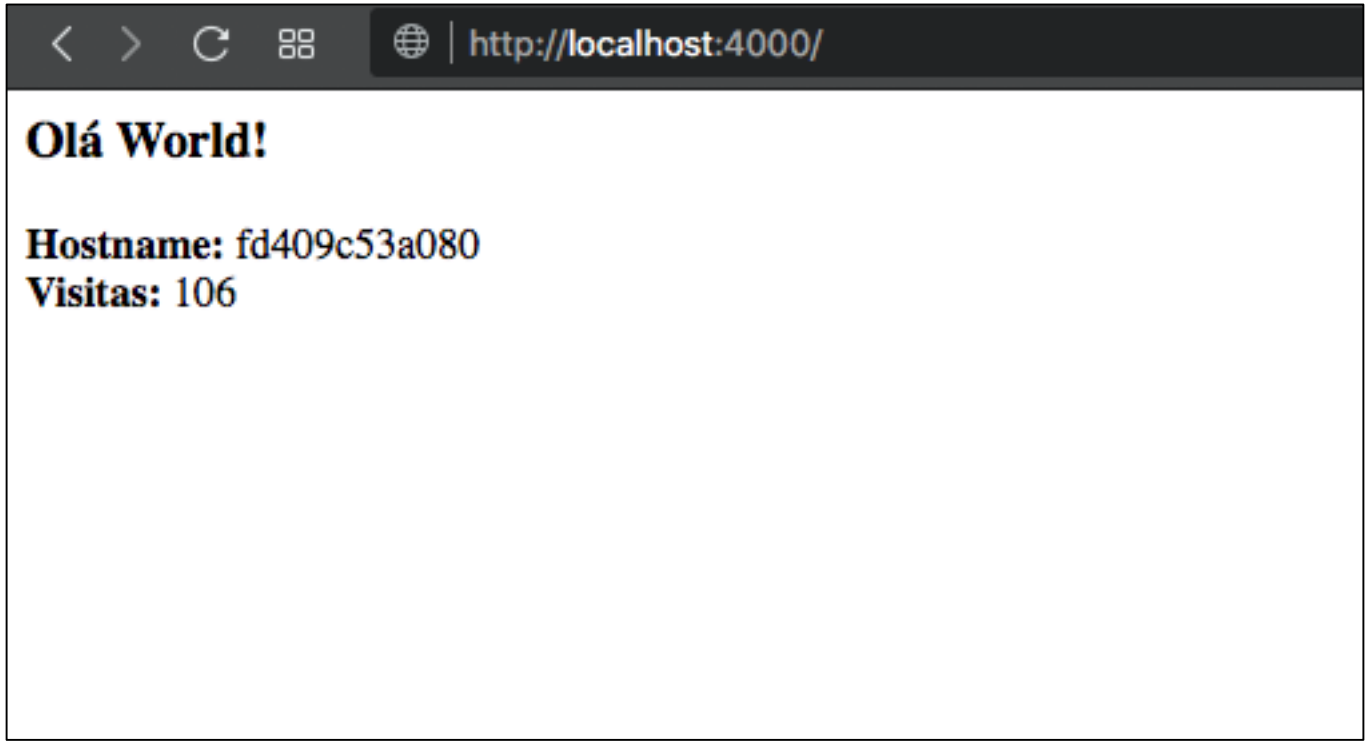
```
$ docker service ls
```

- Para consultar os containers:

```
$ docker container ls
```

ORQUESTRANDO OS SEUS CONTAINERS

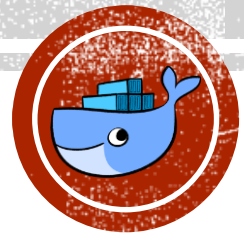
- Agora ao acessar o navegador você deve visualizar o contador de visitas em funcionamento.



REFERÊNCIAS

- Docker Swarm - https://docs.docker.com/engine/swarm/
- Docker Stack - https://www.mundodocker.com.br/tag/docker-stack/
- Redis - https://redis.io/

DÚVIDAS?



Douglas Nassif Roma Junior

/douglasjunior

/in/douglasjunior

smarppy.com

douglas@smarppy.com

Slides: https://git.io/fpgYO