

Você sabia que seu material didático é interativo e multimídia? Ele possibilita diversas formas de interação com o conteúdo, a qualquer hora e de qualquer lugar. Mas na versão impressa, alguns conteúdos interativos são perdidos, por isso, fique atento! Sempre que possível, opte pela versão digital. Bons estudos!

Sistemas Distribuídos

Simulando sistemas distribuídos com Docker

Unidade 3 - Seção 3

Nesta webaula, apresentaremos os comandos Docker que podem ser utilizados em sistemas distribuídos, por exemplo, comandos para containers, cluster e de monitoramento. Vale salientar que a leitura do livro didático é indispensável, uma vez que os assuntos são tratados de forma aprofundada.


Docker

É uma famosa plataforma genérica de containerização, conceito parecido com virtualização, porém considerado “mais leve” que este. Atualmente, containers são populares devido à facilidade e à flexibilidade que advêm de seu uso. Assim, chegou o momento de aprendermos a utilizar essa famosa ferramenta.


Instalação do Docker

Todo o procedimento de instalação deve ser feito com um usuário com permissões de “administrador”. No caso que será apresentado a seguir, utilizaremos o sistema operacional Ubuntu (versão mais atualizada 14.04.5 LTS) e o “root”, por meio do comando “sudo su”. Em resumo, o processo de instalação é bem simples, sendo somente necessário avançar as seguintes etapas:

- 1. Antes de instalar a ferramenta, devemos remover versões anteriores do Docker que possam estar instaladas, utilizando o seguinte comando:



```
sudo apt-get remove docker docker-engine docker.io
```



Caso não tenha nenhuma versão instalada, será exibida a mensagem que foi impossível encontrar o pacote docker-engine.

Iniciando e testando o Docker

Para ver se o Docker foi instalado corretamente, devemos iniciar o serviço do Docker e, após isso, verificar se ele está em execução. Para tanto, é necessário utilizar os seguintes comandos:

```
sudo      service docker start  
service docker status
```

Agora que instalamos e verificamos seu funcionamento, o sistema está apto a receber as especificidades que queremos criar. Para isso, é possível usar o **Docker Swarm**. Nesse cenário é possível agrupar vários hosts em um mesmo pool de recursos, o que facilita o **deploy de containers** (DIEDRICH, 2018).

Após a criação do nosso login na plataforma, teremos acesso a uma interface para criação de instâncias como clusters e nós.

Por meio dos comandos que serão apresentados a seguir, podemos simular algumas características de sistemas distribuídos com Docker.

Comando executado “fora” do(s) nó(s)

Um dos comandos importantes é o de **criação de um novo manager para o nosso cluster**. Através dele, criamos um manager chamado de “mestre”:

```
docker-machine create --driver virtualbox mestre
```

Ao executarmos esse comando, o começo dele significa que estamos criando uma nova máquina através do Docker (“docker-machine create”).

Comando executado “dentro” do(s) nó(s) manager

O comando a seguir pode ser utilizado para iniciar o cluster através do framework swarm, de gerenciamento de containers.

```
docker swarm init --advertise-addr < IP DO MESTRE >
```

O comando a seguir pode ser utilizado para iniciar o cluster através do framework swarm, de gerenciamento de containers.

“node ls”

“inspect”

“ps”

“update”

Podemos consultar os nós que fazem parte do cluster utilizando o comando “node ls”:

```
docker node ls
```

Comando executado “dentro” do(s) nó(s) worker

O comando a seguir pode ser utilizado para adicionar um nó escravo (worker) ao nosso cluster. Para isso, devemos acessar o nó escravo que queremos adicionar a um cluster e executar o comando a seguir, passando o token de segurança e o IP do nó mestre (manager), seguido por porta, conforme a sintaxe representada no comando:



```
docker swarm join --token < IP do mestre:Porta >
```

Comando executado “dentro” de cada UM dos nós (managers e workers)

Quando executamos o comando Docker “system prune” com o parâmetro “all” dentro de um nó, ele será responsável por apagar/deletar tudo o que foi feito dentro do mesmo. Conforme observamos a seguir sua utilização:

```
docker system prune -all
```

Para terminar os estudos desta webaula, vamos aprender a orquestrar o servidor web Apache em um cluster simples. Primeiramente, você deve estar logado na plataforma de “playground” do Docker. Em seguida, será necessário:

- 
1. Criar um cluster com 3 nós, que serão suficientes para analisarmos nosso cluster sem comprometer a usabilidade da plataforma de testes do Docker. Sendo assim, você deve adicionar 3 nós através do botão “Add new instance”.
- 

Esta sequência de comandos que utilizamos para orquestrar um servidor web Apache em 3 nós é a configuração utilizada em sistemas distribuídos para que os acessos há um website sejam balanceados. Vale ressaltar que, caso ocorra algum problema em um dos nós que mantém a aplicação, o outro nó assume a execução.

Concluindo esta webaula, esperamos que você tenha tido um bom proveito dos conteúdos apresentados!
Bons estudos!



Vídeo de encerramento

Para visualizar o vídeo, acesse seu material digital.