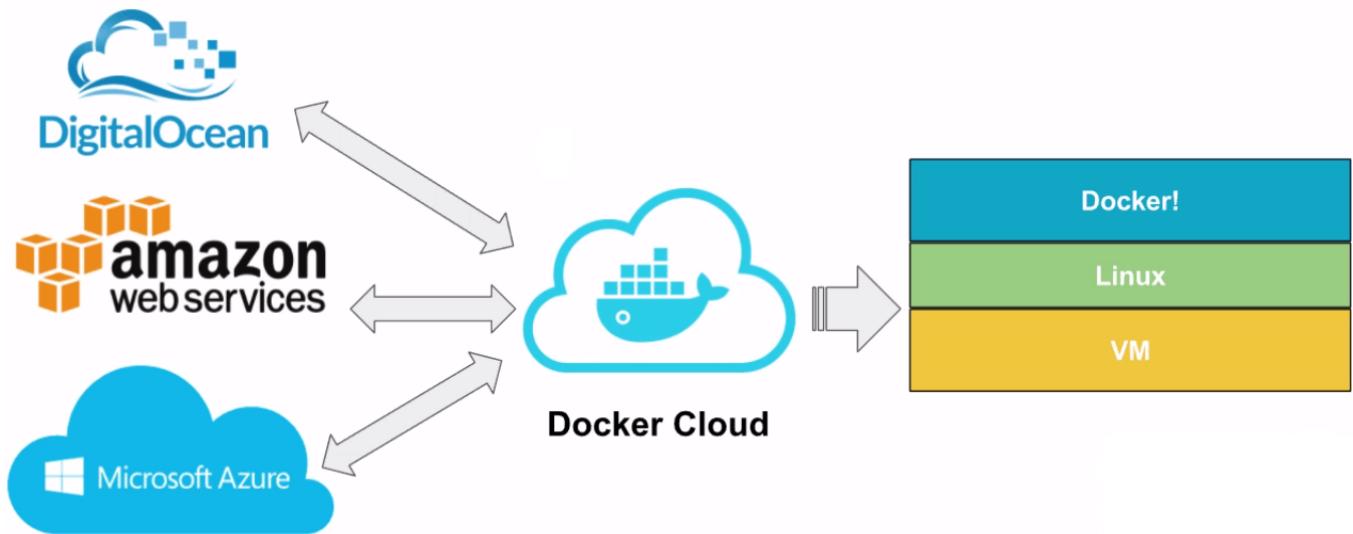


05

Deployando um Hello World no Docker Cloud

Transcrição

Lembrando do diagrama:



A primeira parte já fizemos, que é ligar o Docker Cloud à nossa conta do AWS, faltando falarmos para o Docker Cloud que ele pode criar uma máquina virtual, instalar um Linux e subir um Docker nela.

No **Docker Cloud**, fazemos isso acessando o menu lateral da esquerda, e clicando em **Nodes**, que é onde terá um **Docker Host** instalado, para subirmos os nossos *containers*. E conforme formos criando um *node*, veremos que na verdade ele é uma máquina virtual que está no AWS.

Criando um node

Para criar um *node*, clicamos no botão **Create**, onde seremos redirecionados para a seguinte página:

The screenshot shows the 'Node Clusters / Wizard' interface. The left panel contains fields for 'NAME' (Node cluster name), 'LABELS' (Select...), and 'PROVIDE' (Select...). The right panel shows a 'Number of nodes' slider set to 1, with a 'node' label below it. A 'Launch node cluster' button is located at the bottom right.

- No campo **NAME**, colocamos um nome para o nosso *node*
- Em **LABELS**, podemos deixar em branco mesmo
- Em **PROVIDER**, colocamos quem irá prover o *node* para nós, no nosso caso o **Amazon Web Services**
- Em **REGION**, podemos colocar a mesma região da nossa conta do AWS
- Em **VPC**, deixamos o valor **Auto**
- Por fim, em **TYPE/SIZE**, precisamos ter atenção, pois para **contas gratuitas**, só podemos utilizar dois tipos de instâncias, a **t2.micro** e a **t2.nano**, onde temos 750 horas por mês sem nenhuma cobrança para nós. Então, vamos escolher a instância **t2.micro**.

Node Clusters / Wizard

NAME

LABELS

PROVIDER

REGION

VPC

TYPE/SIZE

Node disk size

 GB

Number of nodes

 node

Launch node cluster

À direita, configuramos quanto queremos de disco e quantos *nodes* nós queremos. Mais uma vez, como estamos com uma conta gratuita, vamos utilizar o mínimo, isto é, **60GB** e **1 node**. Assim, podemos clicar em **Launch node cluster**.

Ao finalizar o treinamento, não esqueça de desligar todos os nodes do Docker Cloud, pois enquanto eles estiverem rodando, horas vão sendo consumidas, podendo gerar cobranças futuras.

Com isso, o *node* será *deployado*, mas clicando nele e acessando a aba **Timeline**, podemos ver tudo o que está sendo feito, conforme for sendo executado no AWS. Quando terminar o *deploy*, o *node* irá para o estado de **DEPLOYED**.

Node Clusters

Quando criamos um *node*, um nó, na verdade é criado por debaixo dos panos um **node cluster**, que é como se fosse uma coleção de vários *nodes*, onde podemos ter diversas máquinas virtuais falando uma com a outra, ou vários *nodes* em uma mesma máquina, mas como estamos fazendo uma aplicação simples, que não precisa de nada muito pesado, criamos um *node* só, então no nosso *cluster* só deve ter um nó rodando.

Subindo uma aplicação no Docker Cloud

Como experimento, vamos subir uma aplicação no Docker Cloud. Para isso, no menu lateral da esquerda, vamos clicar em **Services** e em seguida no botão **Create**. Temos três abas, na primeira (um ícone de foguete) temos algumas aplicações

padrões, na segunda (um ícone de planeta) podemos buscar alguma imagem no Docker Hub, e na terceira (um ícone de bloco de notas) temos acesso diretamente às nossas imagens, sem precisarmos buscá-las.

Como é a nossa primeira vez, vamos utilizar a imagem padrão **dockercloud/hello-world**. Ao clicar nela, vamos para uma página de configuração, onde detalharemos alguns campos:

- **IMAGE**: a imagem que vamos utilizar e a sua versão
- **SERVICE NAME**: o nome do serviço
- **NICKNAME**: um apelido opcional para o serviço
- **CONTAINERS**: a quantidade de *containers* que queremos no serviço
- **AUTORESTART**: temos três opções, nunca reiniciar o serviço, somente reiniciar o serviço quando o mesmo falha, ou reiniciar sempre o serviço
- **AUTODESTROY**: temos três opções, nunca destruir o serviço, somente destruir o serviço quando o mesmo executar com sucesso, ou destruir sempre o serviço

Entre várias outras opções que o **Docker Cloud** nos disponibiliza, onde podemos configurar tudo manualmente em uma interface bem intuitiva. Há algumas outras opções que já conhecemos, como **NETWORK**, **ENTRYPPOINT**, que é o comando que executará quando o *container* rodar, além de também podermos limitar a CPU e a memória para o *container* utilizar.

Em **Ports**, liberamos portas do *container*. Essa imagem utilizar a porta **80**, que nem podemos alterar, e para acessá-la de fora, devemos torná-la pública, marcando a *checkbox* **Published**. Com isso, nós de fora podemos acessar esse *container*, basicamente a mesma coisa que fazíamos na nossa máquina, em que colocávamos uma porta externa para acessarmos uma porta do *container*. E podemos escolher a porta do *node* que será atrelada à porta **80** do *container*, mas vamos deixar com o valor **dynamic**, para o próprio Docker escolher a porta.

Tudo o que fizemos na linha de comando, podemos fazer no Docker Cloud, mas com uma interface gráfica. Agora não precisamos configurar mais nada, então vamos clicar em **Create & Deploy**.

Quando o serviço estiver no estado de **RUNNING**, para acessá-lo, vamos até a sessão **Endpoints**, e vemos o endereço gerado pelo Docker Cloud. Ao clicar nesse endereço, vemos a seguinte página:



Hello world!

My hostname is hello-world-docker-cloud-1

Ou seja, estamos executando um *container* em uma máquina do AWS, tudo orquestrado pelo **Docker Cloud**. Em nenhum momento tivemos que acessar o AWS para criar uma máquina, instalar um sistema, o Docker, etc, já que tudo foi gerenciado pelo **Docker Cloud**.

Na página **Services**, temos todos os detalhes do serviço que criamos, das configurações que fizemos. E em **Containers**, também no menu lateral da esquerda, podemos ver o *container* criado.

Verificando a instância criada

Se formos no AWS, acessar a página do **EC2**, que são as máquinas virtuais da Amazon, e clicar em [Running Instances](https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceId) (<https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:sort=instanceId>), podemos ver a instância criada pelo Docker Cloud no momento em que criamos o *node*, com todas as configurações necessárias, inclusive subindo um Docker lá dentro. E quando criamos o serviço, o Docker Cloud foi até a instância, executou os comandos para levantar um *container* com a imagem que pedimos e fez toda essa configuração para nós.

Vimos como *deployas* uma imagem, as suas configurações, então no próximo vídeo veremos como trabalhar com mais serviços, fazendo o *deploy* da aplicação que fizemos na aula anterior.