MATHEUS PAVANI

PÓS TECH

SOFTWARE ARCHITECTURE

DOCKERIZAÇÃO

AULA 07

SUMÁRIO

O QUE VEM POR AÍ?

Nesta aula, vamos aprender como trabalhar com Docker em ambientes cloud, mais especificamente, com a AWS. Imagine só poder fazer tudo o que foi feito anteriormente, só que em um ambiente cloud?

Com isso, vamos entender um pouco como utilizar a ferramenta ECS nesta jornada.

Ao final da disciplina, temos um desafio que faz parte da sua jornada de aprendizado, e você pode acessá-lo na área de atividades da plataforma FIAP ON.

Este desafio vai te ajudar a praticar os conhecimentos adquiridos durante as aulas e te preparar para o projeto da fase!

HANDS ON

Agora veremos, na prática, como subir containers e imagens Docker utilizando recursos da AWS. Isso fará parte de um conhecimento que simulará como podemos juntar EC2, ECR e o ECS!

A seguir, temos uma ideia de código que foi desenvolvido na videoaula, mas é importante que, ao olhar para ele, seja possível você desenvolver e trabalhar de maneira livre. Isso significa que é interessante mexer no código e praticar, explorar as funcionalidades e parâmetros. Por isso, as documentações são tão importantes!

SAIBA MAIS

Quando trabalhamos com tecnologia, é muito comum chegarmos ao ponto de começar a trabalhar em ambientes produtivos e controlados, e isso envolve o que chamamos de cloud.

Existem diversas clouds onde podemos construir aplicações, subir vm's e muitos mais!

Uma das mais utilizadas e que com certeza mostra para o que veio, é a AWS, que é a cloud da Amazon, já que AWS significa Amazon Web Services.

Só que a questão aqui é: como posso relacionar tudo que aprendi em Dockerização até o momento, com uma cloud AWS? Será que conseguimos construir uma imagem e 'subir' esse contêiner em uma máquina virtual adequada?

A resposta para essas perguntas é: SIM!

O primeiro passo para iniciarmos essa trajetória é criar uma conta na AWS <https://aws.amazon.com/>.

Você deverá seguir todos os passos e, um detalhe importante, é: cadastrar um cartão de crédito válido. Mas, acalme-se, pois a Amazon, assim como outras empresas que oferecem serviço cloud, precisam verificar a veracidade de suas informações, e nada será cobrado cartão, a não ser que você utilize os serviços pagos da plataforma! Só que aqui, utilizaremos serviços que não cobram nada por isso, desde que você siga determinadas especificações.

Caso você tenha dúvidas de como fazer tudo isso, a AWS disponibiliza documentações incríveis em um o passo a passo. Um exemplo é como criar máquinas virtuais utilizado a ferramenta EC2, cujo arquivo PDF você encontra aqui: <https://docs.aws.amazon.com/pt_br/codedeploy/latest/userguide/codedeploy-user.pdf#instances-ec2-create>.

Pois bem! Aqui vamos partir do princípio de que temos, de fato, uma instância do EC2 criada, e vamos construir nosso Dockerfile lá dentro.

Só que antes de tudo isso, precisamos instalar o Docker na instância e para isso utilizaremos uma máquina do tipo t2.micro com a imagem do Linux, da própria AWS.

Para começar, acesse a máquina dentro do EC2, e a visão será parecida com esta:

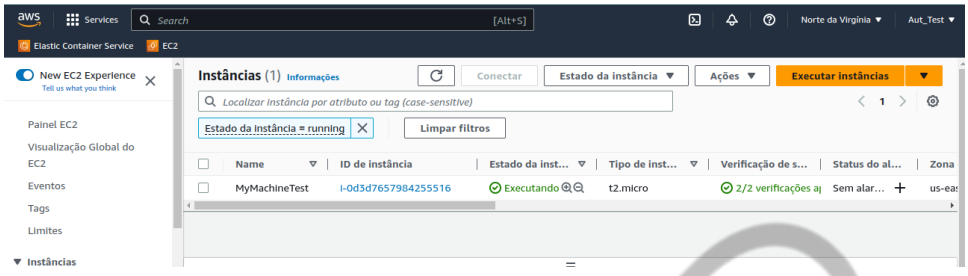

Figura 1 – EC2
Fonte: Elaborado pelo autor (2023)

No campo “ações”, conecte na máquina e, logo em seguida, abrirá um terminal muito parecido com o que temos no computador:

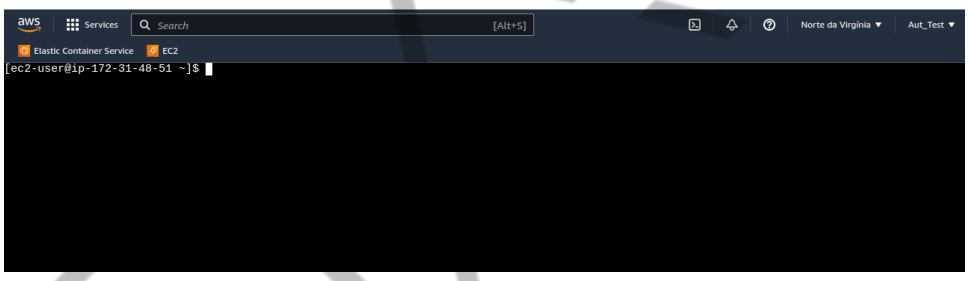

Figura 2 – Terminal EC2
Fonte: Elaborado pelo autor (2023)

A partir daqui, devemos seguir uma sequência de comandos para instalar o Docker e, como sempre, devemos fazer um update no sistema:

```
sudo yum update –y
```

Após essa ação, vamos instalar o pacote do Docker engine mais recente:

```
sudo amazon-linux-extras install docker
```

Aqui, estamos supondo que a sua máquina seja um sistema Linux do tipo Amazon Linux 2 AMI.

Vamos iniciar o serviço do Docker com o comando:

```
sudo service docker start
```

Uma boa alternativa, mas que é totalmente opcional, é fazer com que o daemon do Docker seja iniciado toda vez que reiniciarmos o sistema. Para isso, usamos o comando:

```
sudo systemctl enable docker
```

Vamos adicionar o user ec2-user para o grupo Docker, fazendo com que possamos executar os comandos Docker sem a necessidade do sudo na frente:

```
sudo usermod –a  -G docker ec2-user
```

Agora, para saber se está sendo possível executar os comandos sem o “sudo”, teste com o seguinte comando:

```
sudo docker info
```

Agora temos uma máquina Linux e com Docker, e podemos trabalhar com o nosso Dockerfile.

Para criar um arquivo Dockerfile, existem várias maneiras diferentes, mas para facilitar, aconselho criar um diretório de trabalho e depois executar o comando:

```
touch Dockerfile
```

Esse comando cria o arquivo, mas não o executa. Agora, precisamos editar nosso Dockerfile e, para isso, podemos usar um dos dois comandos abaixo:

```
sudo vi Dockerfile

sudo vim Dockerfile
```

Após isso, você terá acesso ao arquivo e poderá começar a edição. Para isso, quando você for acessar no arquivo, aperte a tecla “i” no teclado para começar a “inserir” informações. Não vamos nos limitar sobre o que vamos escrever lá, aqui estamos preocupados apenas em fazer esse Dockerfile para que posteriormente essa imagem seja útil dentro do ECS.

Um exemplo de Dockerfile pode ser:

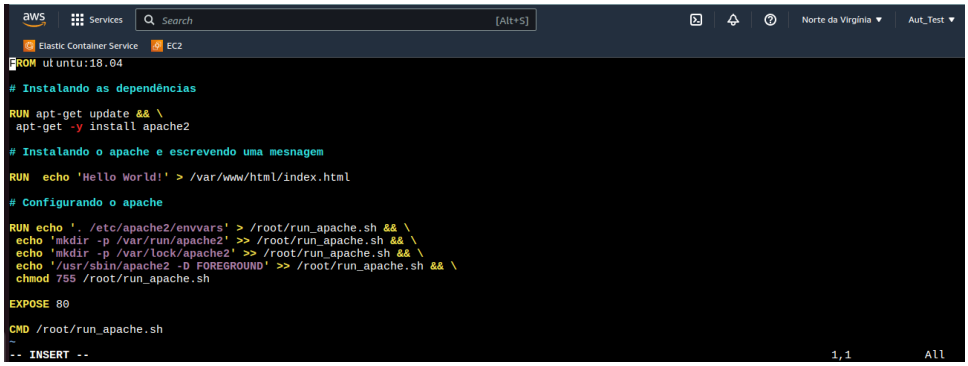

Figura 3 – Dockerfile
Fonte: Elaborado pelo autor (2023)

Depois de criar o arquivo, é hora de fazer o build da nossa imagem. Para isso rode o comando:

docker build –t teste .

Após isso, podemos criar um comando fazendo o contêiner rodar:

docker run –t –i –p 80:80 teste

Agora você poderá ver uma app web no ar, desde que seu grupo de segurança permita que a porta 80 esteja liberada.

Algumas aulas atrás, tratamos do conceito de Registry, e na AWS não poderia faltar! Ele se chama Elastic Container Registry ou mais conhecido como ECR, onde todas as nossas imagens serão armazenadas para uso futuro em ferramentas como ECS.

Pois bem! Aqui já podemos criar nosso primeiro repositório com o comando:

aws ecr create-repository --repository-name hello-repository --region region

Os pontos em vermelho destacam o nome que você gostaria de dar para o repositório e qual região você está utilizando (comumente estamos na região us-east-1).

Em seguida, é interessante marcar a imagem utilizando o que fizemos anteriormente:

```
docker tag test aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository
```

Agora, podemos autenticar:

```
docker login -u AWS -p $(aws ecr get-login-password --region region) aws_account_id.dkr.ecr.region.amazonaws.com
```

E podemos finalmente subir no ECR:

```
docker push aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository
```

Após a criação do nosso registry, precisamos entender como criar o cluster para gerenciar nossos contêineres, e é aí que o ECS entra em cena! Para acessar o ECS, basta você digitar na barra de pesquisa o nome dele e clicar no resultado. Assim que for feito, chegaremos à esta tela:

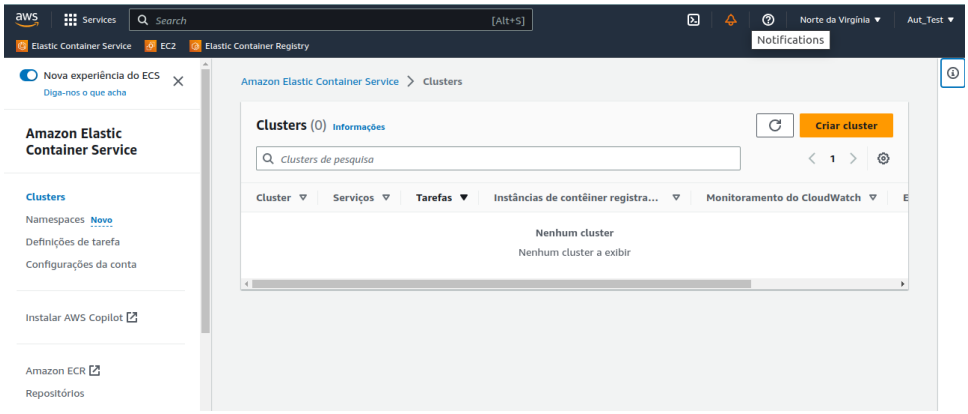

Figura 4 – ECS
Fonte: Elaborado pelo autor (2023)

Clique em ‘criar cluster’ para ser redirecionado para esta tela:

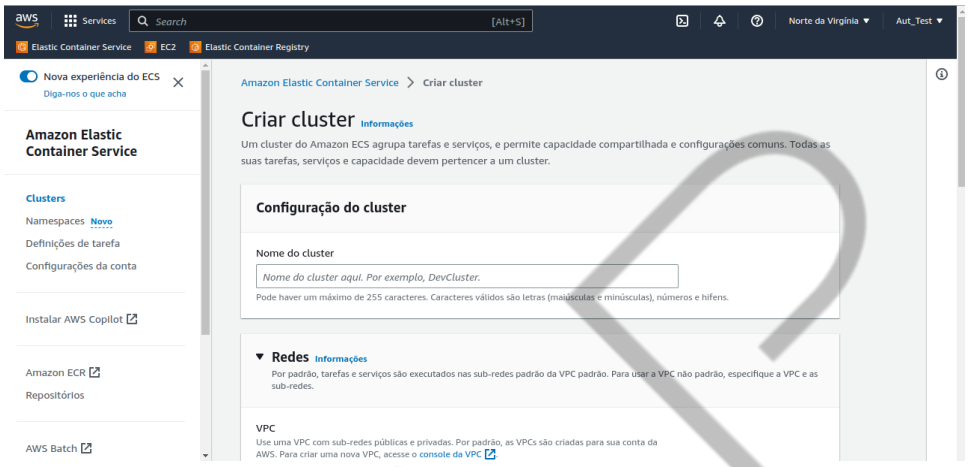

Figura 5 – ECS configuration
Fonte: Elaborado pelo autor (2023)

Aqui devemos configurá-lo. O ideal é que você atrele esse cluster com uma intância EC2 (para economizar recursos, utilize uma t2.micro e um amazon linux 2).

Após algum tempo, seu cluster estará criado:

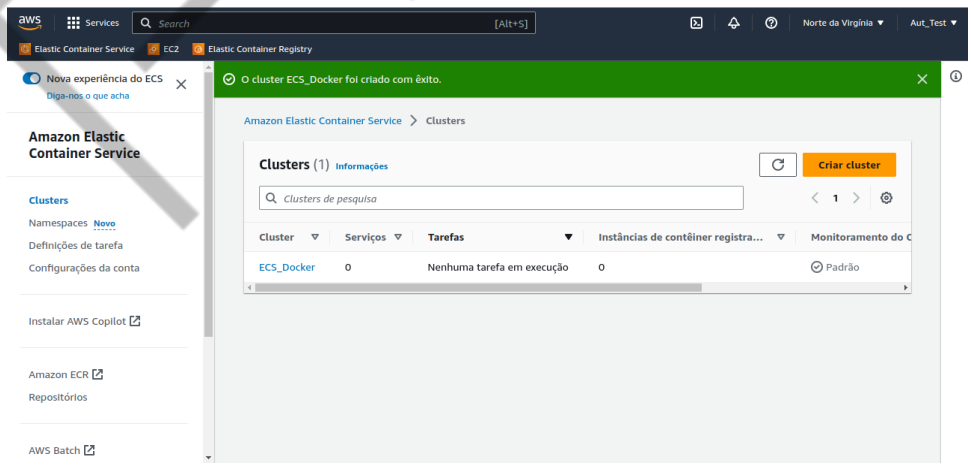

Figura 6 – Cluster criado
Fonte: Elaborado pelo autor (2023)

Entrando no cluster, aparecerá a opção de criar tarefas, onde será possível criar uma tarefa:

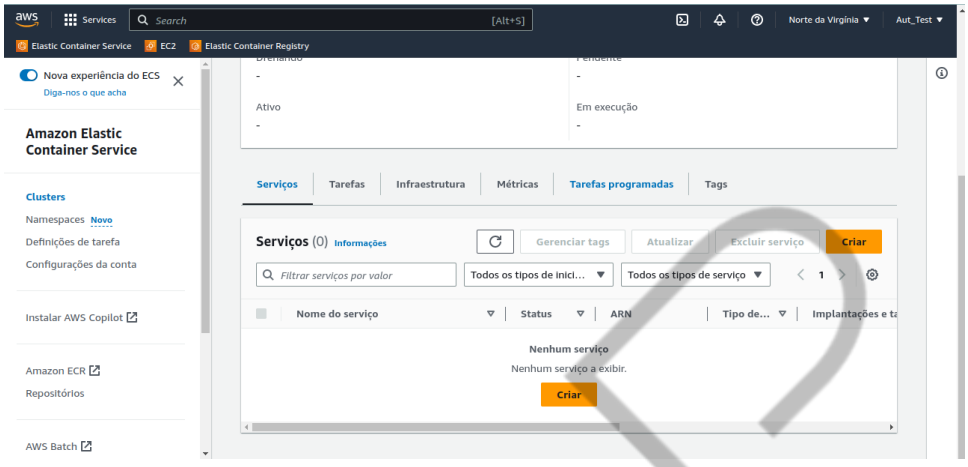

Figura 7 – Tasks
Fonte: Elaborado pelo autor (2023)

Depois disso, você verá a tela de criação:

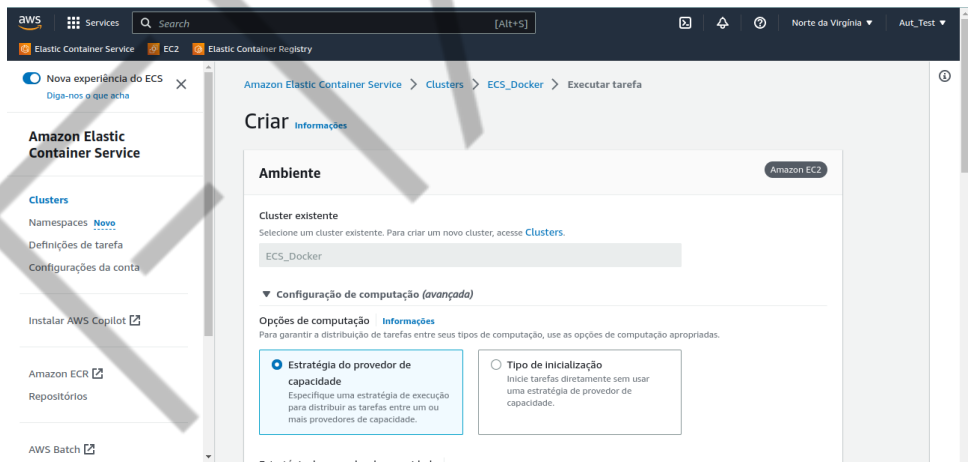

Imagem 8 – Criar task
Fonte: Elaborado pelo autor (2023)

Para continuar, um pouco mais abaixo, ele vai pedir para que seja adicionada uma família de definição de task (um link redirecionará para o passo a passo). Lembrando que, nesta etapa é importante pegar a URI da imagem no ECR para poder passar no processo.

Após isso, você poderá voltar para a aba de configurar uma nova task e passar a nova definição da tarefa, e você terá o seguinte resultado:

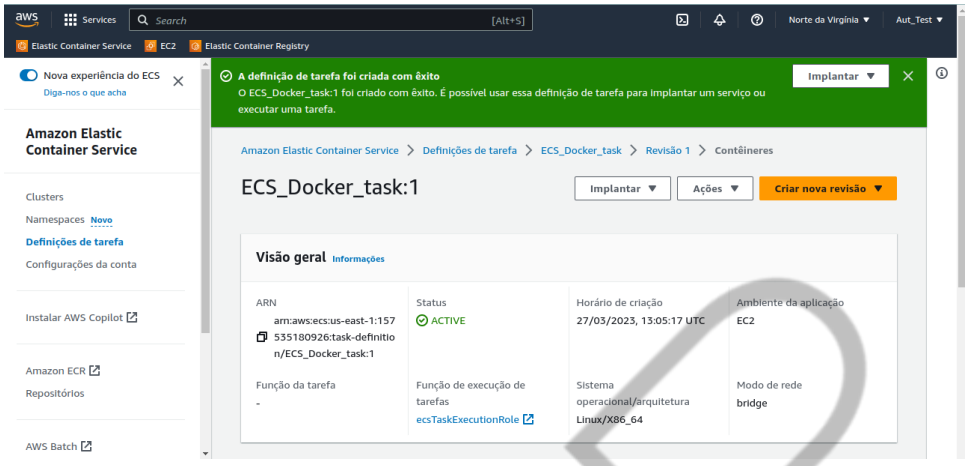

Figura 9 - Task criada com container 1
Fonte: Elaborado pelo autor (2023)

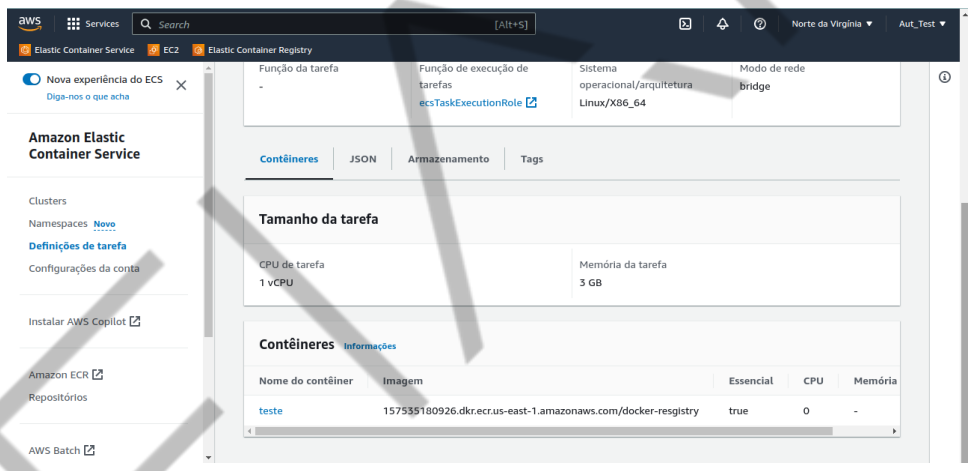

Figura 10 – Task criada com container 2
Fonte: Elaborado pelo autor (2023)

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, vimos como trazer uma imagem e um contêiner para um ambiente produtivo e controlado, utilizando recursos cloud da AWS!

Daqui para a frente, é importante que você replique os conhecimentos adquiridos para fortalecer mais suas bases e conhecimentos, já que um bom ou uma boa cientista de dados não é somente aquele(a) que é uma enciclopédia humana, mas sim aquele(a) que sabe ler um problema e atuar com eficácia.

IMPORTANTE: não esqueça de praticar com o desafio da disciplina, para que assim você possa aprimorar os seus conhecimentos!

Você não está só nesta jornada! Te esperamos no Discord e nas lives com os nossos especialistas, onde você poderá tirar dúvidas, compartilhar conhecimentos e estabelecer conexões! Aproveite!

REFERÊNCIAS

POULTON, N. **Docker Deep Dive**. [s.l.]: Independently published, 2017.

VITALINO, J. F. N. **Descomplicando o Docker**. [s.l.]: Brasport. 2018.

PALAVRAS-CHAVE

Docker. Elasticsearch. Network.