Tutorial: Criar um pipeline simples (repositório CodeCommit)

Digite IAM para encontrar a opção

**1º Passo**  
Escolher a opção IAM

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**2º Passo**

Clique em Usuários:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Clique em usuários

**3º Passo**

Procure seu nome e clique sobre ele

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Clique sobre ele

Procure seu nome

**4º passo**

Clique em credenciais de segurança

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Clique aqui

**5º passo**

Desça a pagina até a opção de Credenciais HTTPS do Git para o AWS CodeCommit e Clique sobre credenciais de segurança

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Baixe a sua credencial e não perca!

**6º Passo**

Precisamos de um arquivo de configuração que tem que estar na raiz do seu projeto, para isso precisamos criar ele:

- abra o projeto no intelijj > Crie um arquivo com o nome **buildspec.yml**

Tela de celular

Descrição gerada automaticamente

Dentro desse arquivo cole esse código – lembre da endentação

Version: 0.2

Phases:

Install:

Runtime-versions:

Java: corretto11

Build:

Commands:

- echo “Fase de build começando...”

- mvc clean install – Dskiptests

post\_build:

commands:

- echo “Pipeline sucesso”

Texto

Descrição gerada automaticamente

**7º passo**

Abre seu GitHub > escolha o projeto que vai migrar (ele precisa estar atualizado com o arquivo buildspec.yml

**8º passo**

Crie a pasta Target com o seguinte comando no terminal

.\mvnw clean install -DskipTests

Texto

Descrição gerada automaticamente

Tela de computador com letras brancas

Descrição gerada automaticamente

Pasta criada

**9º passo**

Abra o CloudShell da WAS

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Site

Descrição gerada automaticamente

**10º Passo**

Cole esse comando no terminal do Shell

migration\_dir="./\_trash"git\_url="URL\_GITHUB"aws\_region="REGIÃO"codecommit\_repo\_name="NOME\_DO\_REPOSITÓRIO\_NA\_AWS"

Substituindo nesse comando

URL\_Github – pela URL do seu repositório que desejar

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Região – pela região que você esta usando

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

NOME\_DO\_REPOSITÓRIO\_NA\_AWS – substituir pelo nome do seu repositorio

**11º passo**

próximo comando

aws codecommit create-repository \ --repository-name ${codecommit\_repo\_name} \ --repository-description "Meu repositorio"

"Meu repositorio" – substituir pela sua descrição

Você vera essa tela:

Texto

Descrição gerada automaticamente

Caso você erre o comando, essa tela aparecerá

Texto

Descrição gerada automaticamente

Aparecerá essa tela e você conseguirá editar o comando novamente.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**12º Passo**

Cole o seguinte comando no terminal sendo que (migretion\_dir é o nome da pasta que você criou, se você mudar esse nome o próximo comando tb muda)

git clone --mirror ${git\_url} ${migration\_dir}

Texto

Descrição gerada automaticamente

**13º Passo**

Cole o seguinte comando para entrar dentro da pasta que voce criou:

cd ${migration\_dir}/



**14º passo**

Cole o seguinte comando

git push [https://git-codecommit](https://git-codecommit/).${aws\_region}.[amazonaws.com/v1/repos/${codecommit\_repo\_name}](http://amazonaws.com/v1/repos/$%7Bcodecommit_repo_name%7D) –all

Ele pedira a sua senha que geramos no passo 5 é agora que você vai informar.

Texto

Descrição gerada automaticamente

**15º passo**

Digite Codecommit para encontrar essa pagina

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Você vera o repositório que você acabou de criar:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**16º passo**

Pesquise por Codepipeline

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**17º passo**

Clique em criar pipeline

Interface gráfica do usuário, Texto, Aplicativo, Email, Site

Descrição gerada automaticamente

**18º passo**

Preencha as informações como abaixo

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Nessa próxima janela clique em Criar projeto

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Essa nova janela se abrirá

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

A pagina vai fechar

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente



Próxima etapa a gente ignora

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Vai aparecer uma revisão de tudo que a gente fez

Interface gráfica do usuário, Texto, Aplicativo, Email, Teams

Descrição gerada automaticamente

Clique em criar papeline

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Fim

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**Instruções colocadas no Slack pela Mari**

(Será visto em aula) Passos para PIPELINE no AWS BUILD:

1. Primeiro ir no IAM -> Usuários -> Credenciais HTTPS -> Gerar Credenciais HTTPS;
2. Depois fazer migrar o repositório Git para o CodeCommit da AWS (São os passos acima) (Só consegue migrar se as credenciais foram geradas no passo 1);
3. Depois para simplificação, clone o repositório da AWS no seu computador e mexa no código a partir daí.
4. Depois criar o arquivo buildspec.yml na raiz do projeto e enviar esse arquivo pro repositório da AWS (git push normal);
5. Depois ir no CodePipeline da AWS e criar um Pipeline.
6. Conferir se deu tudo certo na esteira de produção (o Pipeline).

Pessoal segue abaixo script para passar o seu repositório do GitHub PARA a AWS. Precisaremos disso para o requisito do desafio final de ter um pipeline em AWS Build.

1. Primeiro configurar senha HTTPS no IAM (Pode escolher SSH também).

 2.  Depois abra um terminal shell na AWS. 3.  Após isso rode os comandos:migration\_dir="./\_trash"  
git\_url="URL\_GITHUB"  
aws\_region="REGIÃO"  
codecommit\_repo\_name="NOME\_DO\_REPOSITÓRIO\_NA\_AWS"

Exemplo: (Esse é o meu exemplo, vc deve mudar de acordo com o seu projeto)  
migration\_dir="./\_trash"  
git\_url="<https://github.com/mariannesalomao/api-spring-docker.git>"  
aws\_region="us-east-1"  
codecommit\_repo\_name="api-spring-docker-aws"4. Depois rode esses comandos:aws codecommit create-repository \  
--repository-name ${codecommit\_repo\_name} \  
--repository-description "Meu repositorio"5. Depois:git clone --mirror ${git\_url} ${migration\_dir}6. Depois entre na pasta da migração, com o comando:cd ${migration\_dir}/7. E finalmente:git push [https://git-codecommit](https://git-codecommit/).${aws\_region}.[amazonaws.com/v1/repos/${codecommit\_repo\_name}](http://amazonaws.com/v1/repos/$%7Bcodecommit_repo_name%7D) --all