

# Preparação do ambiente de desenvolvimento

---

Para participar do workshop e realizar as atividades práticas, será necessário preparar seu ambiente/máquina de desenvolvimento, instalando as seguintes ferramentas/bibliotecas:

- NodeJS e npm
- AWS CLI
- JDK 8 ou superior
- Maven 3.5 ou superior
- IntelliJ IDEA Community ou outra IDE para se trabalhar com Java
- Postman
- AWS CDK

Os passos a seguir descrevem o que deve ser feito para cada item.

## 1) Instalação do NodeJS e npm:

---

Vá até esse [link](#) e baixe o instalador do NodeJS LTS, de acordo com o seu sistema operacional. O processo será guiado pelo instalador. Ao final, abra um terminal para conferir as versões do NodeJS:

```
1 | node -v
```

A resposta esperada deverá conter a versão instalada do NodeJS, como no exemplo a seguir:

```
1 | v14.16.1
```

e do npm:

```
1 | npm -v
```

Da mesma forma para o `npm`:

```
1 | 7.9.0
```

Cada comando deverá exibir a versão correspondente do pacote que foi instalado.

## 2) AWS CLI:

---

Vá até esse [link](#) e baixe o instalador do AWS CLI, que deverá estar no canto superior direito da página, de acordo com o tipo do sistema operacional.

Ao final do processo de instalação, abra um terminal e verifique a versão do AWS CLI através do seguinte comando:

```
1 | aws --v
```

O resultado deverá ser algo semelhante ao trecho a seguir:

```
1 | aws-cli/2.0.58 Python/3.7.4 Darwin/20.4.0 exe/x86_64
```

### 3) Java Development Kit:

---

O projeto criado com o AWS CDK para a construção da infraestrutura será feito utilizando a linguagem Java. Para isso será necessário instalar o JDK 8 ou superior, que pode ser baixado nesse [link](#), de acordo com o seu sistema operacional.

### 4) Maven:

---

O projeto criado pelo AWS CDK, quando se utiliza a linguagem Java, é baseado no Maven. Então para isso, acesse esse [link](#), baixe-o e instale-o seguindo as orientações segundo o seu sistema operacional.

### 5) IntelliJ IDEA Community:

---

O IntelliJ IDEA Community é uma IDE gratuita para se trabalhar com projetos em Java. Caso você já tenha familiaridade com outra IDE para Java, fique à vontade para utilizá-la, lembrando que o projeto será baseado em **Maven**, logo sua IDE deverá estar preparada para isso.

Caso você queira utilizar a mesma IDE que será usada no workshop, acesse esse [link](#) e baixe a versão **Community**, de acordo com o seu sistema operacional.

### 6) Postman:

---

Durante os testes da aplicação, depois que for hospedada na AWS, será utilizado o Postman, que é um cliente REST. Ele pode ser baixado através desse [link](#).

Qualquer cliente REST poderá ser utilizado para os testes durante o workshop.

## 7) AWS CDK:

Tendo instalado o `npm`, agora é possível instalar o AWS CDK. Para isso, abra um terminal e execute o seguinte comando:

```
1 | npm install -g aws-cdk
```

Para verificar se ele foi instalado corretamente, execute o comando a seguir para verificar sua versão:

```
1 | cdk --version
```

O resultado deverá ser como o trecho a seguir:


```
1 | 1.93.0 (build c506d3b)
```

## 8) Criando um projeto de teste:

Para testar todos os passos de instalação aqui apresentados, abra um terminal em uma pasta de sua preferência e crie um projeto com o AWS CDK, executando os seguintes comandos:

```
1 | mkdir aws_ecs_fargate_cdk
2 | cd aws_ecs_fargate_cdk/
3 | cdk init app --language java
```

Os dois primeiros comandos são para criar uma pasta e navegar para dentro dela, respectivamente. O terceiro comando cria efetivamente o projeto com o AWS CDK. O resultado esperado deve ser semelhante ao trecho a seguir:

```
1 | ## Useful commands
2 |
3 | * `mvn package`      compile and run tests
4 | * `cdk ls`          list all stacks in the app
5 | * `cdk synth`       emits the synthesized CloudFormation template
6 | * `cdk deploy`      deploy this stack to your default AWS account/region
7 | * `cdk diff`        compare deployed stack with current state
8 | * `cdk docs`        open CDK documentation
9 |
10 | Enjoy!
11 |
12 | Initializing a new git repository...
13 | Executing 'mvn package'
14 |  All done!
```

Agora o projeto Maven que foi criado com esse comando pode ser aberto no IntelliJ IDEA.

Dentro do IntelliJ IDEA, basta abrir o projeto, no mesmo local onde está o arquivo `pom.xml`.