



CAPACITAÇÕES  
**JAVA PARA  
A PDPJ-Br**

# JAVA AVANÇADO

CADERNO DE ATIVIDADES - TRILHA 4

*Ronaldo Pinheiro Gonçalves Junior*

## ATIVIDADES DO CADERNO

### Atividade 1: Implantação do projeto Spring Boot em containers no Docker

**Objetivo:**

Utilizar o Docker Compose, para implantar o projeto Spring Boot em containers.

**Requisitos:**

- Atividade 5 (Caderno 2)
- Atividade 7 (Caderno 3)
- Atividade 8 (Caderno 3)
- Atividade 10 (Caderno 3)

**Descrição da atividade 1:**

Nesta primeira atividade da quarta trilha de aprendizagem, vamos configurar e implantar o projeto Spring Boot desenvolvido anteriormente, utilizando o serviço Docker Compose. Você pode acessar o DockerHub (<https://hub.docker.com/>) e utilizar imagens com suporte oficial das tecnologias necessárias no projeto:

- Keycloak: "bitnami/keycloak"
- PostgreSQL: "postgres"
- FlyWay: "redgate/flyway"
- RabbitMQ: "rabbitmq"
- WildFly: "jboss/wildfly"
  - Esta versão não está atualizada, pois foi movida do repositório DockerHub para o repositório Quay da Red Hat.
  - Para uma versão mais recente, use o "wildfly / wildfly".
    - <https://quay.io/repository/wildfly/wildfly>

Caso deseje um maior controle nos servidores e suas instalações ou por motivos de segurança, você pode criar cada imagem, através de um Dockerfile distinto, e usar suas próprias imagens. Durante essa atividade, recomendo o uso de um container por vez. Após testado e verificado que está funcional, passe para o próximo container. Alternativamente, você pode tentar ajustar o arquivo "docker-compose.yml" a seguir, para o seu sistema, e executar todos os serviços de uma única vez.

version: '2'

services:

wildfly:

image: wildfly

ports:

- "8080:8080"

volumes:

- ./<caminho-para-sua-pasta-de-deployments>:/opt/jboss/wildfly/standalone/deployments

keycloak:

image: bitnami/keycloak

environment:

- KEYCLOAK\_USER=<seu-usuário-keycloak>

- KEYCLOAK\_PASSWORD=<sua-senha-keycloak >

ports:

- "8081:8080"

postgresql:

image: postgres

environment:

- POSTGRES\_DB=<seu-banco-de-dados>
- POSTGRES\_USER=<seu-usuário-postgresql>
- POSTGRES\_PASSWORD=<sua-senha-postgresql>

volumes:

- ./<caminho-para-seu-postgresql>/data:/var/lib/postgresql/data

flyway:

image: redgate/flyway

command: -url=jdbc:postgresql://postgresql:5432/ <seu-banco-de-dados> -user= <seu-usuário-postgresql> -password= <sua-senha-postgresql> migrate

rabbitmq:

image: rabbitmq

ports:

- "5672:5672"
- "15672:15672"

environment:

- RABBITMQ\_DEFAULT\_USER=user
- RABBITMQ\_DEFAULT\_PASS=password

log4j2:

image: alpine

command: tail -f /dev/null

volumes:

- ./ <caminho-para-seus-logs>:/logs

## Atividade 2: Documentação automática através do Swagger (Open UI)

### Objetivo:

Realizar a documentação do projeto Spring Boot e seus serviços.

### Requisito:

- Atividade 1

### Descrição da atividade 2:

Utilize as anotações do SpringDoc Swagger Open UI, para realizar a documentação automática do seu projeto Spring Boot. Inclua todas as informações abaixo em seu projeto (para projetos longos, comece por um controlador, um serviço e assim por diante):

- Descrição Geral da API: texto geral, com propósito, recursos fornecidos e instruções para clientes;
- Recursos e Endpoints: liste os recursos disponíveis e mencione os respectivos endpoints, fornecendo detalhes de todos os métodos HTTP presentes (GET, POST, PUT, DELETE), URLs, parâmetros de consulta, cabeçalhos etc.;
- Exemplos de Uso: crie exemplos concretos de uso para que clientes possam se basear, durante o acesso aos endpoints. Inclua exemplos de requisições e respostas HTTP, com cabeçalho e corpo no formato esperado;
- Esquemas de Dados: defina todos os tipos de dados esperados para entradas e defina os tipos de dados enviados para saídas, mostrando quais formatos são suportados (JSON, XML etc.);
- Códigos de Status HTTP: documente o que significa cada código de *status* HTTP;
- Autenticação e Autorização: onde houver acesso que requer autenticação e autorização, descreva os métodos suportados pela API, incluindo os tipos de token e esquemas de autenticação e permissões necessárias para acessar recursos específicos;
- Limites de Taxa: se usar limite de taxa, documente os detalhes de limite, como o número máximo de solicitações permitidas, para um determinado intervalo de tempo;



- Versão da API: se usar versionamento, documente a versão da API e opte por usar um versionamento no endereço (exemplo, “/api/arquivos/v1”).

### Atividade 3: Criação de conta na Amazon Web Services (AWS)

**Objetivo:**

Criar uma conta na plataforma AWS, para utilizar o serviço EKS e demais serviços (IAM, VPC, EC2, EBS etc.).

**Requisitos:**

- O uso do serviço Amazon EKS presente neste material exige uma conta na plataforma AWS;
- Para criar uma conta na AWS, será necessário informar um cartão de crédito.

**Descrição da atividade 3:**

Esta atividade é destinada àqueles que não possuem uma conta da AWS. Atualmente, a criação de uma nova conta irá lhe conferir os recursos gratuitos, contidos no período de teste de um ano do Free Tier. Utilize o guia oficial da Amazon Web Services para criar sua conta: [https://docs.aws.amazon.com/pt\\_br/accounts/latest/reference/welcome-first-time-user.html](https://docs.aws.amazon.com/pt_br/accounts/latest/reference/welcome-first-time-user.html)

Contas recém-criadas na AWS ganham um “Free Tier”. Podemos usar opções gratuitas durante esse período, para realizar as próximas atividades do caderno.

**Aviso!** Ao final do período de testes, custos podem ser gerados se recursos ainda estiverem em uso. Lembre-se de liberar todos os recursos da nuvem, logo após a realização das atividades.

### Atividade 4: Configuração do Identity and Access Management (IAM)

**Objetivo:**

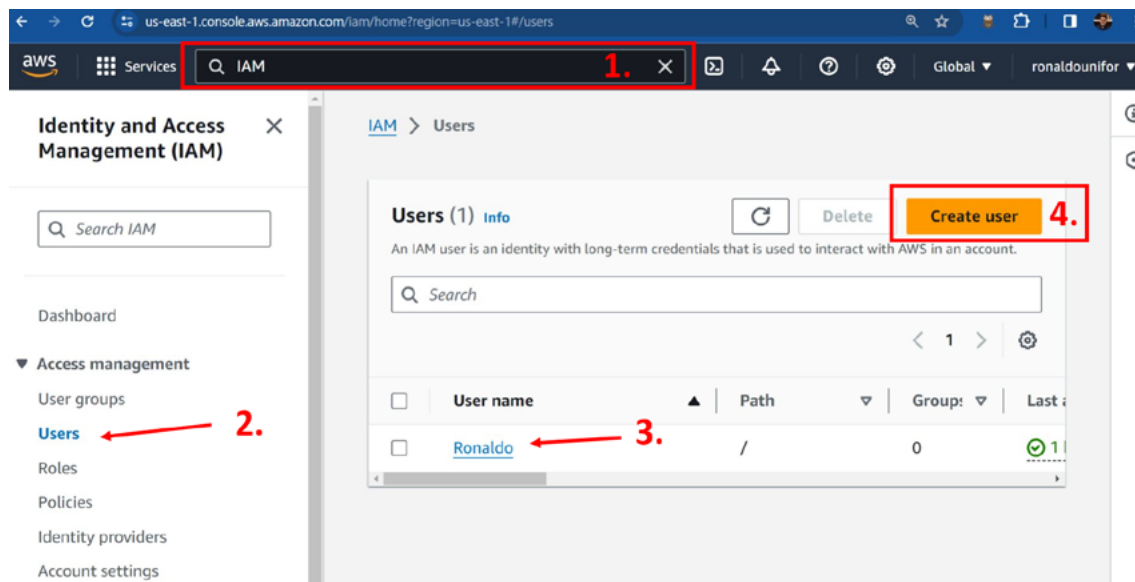
Configurar as credenciais e permissões do usuário da conta AWS.

**Requisito:**

- Atividade 3

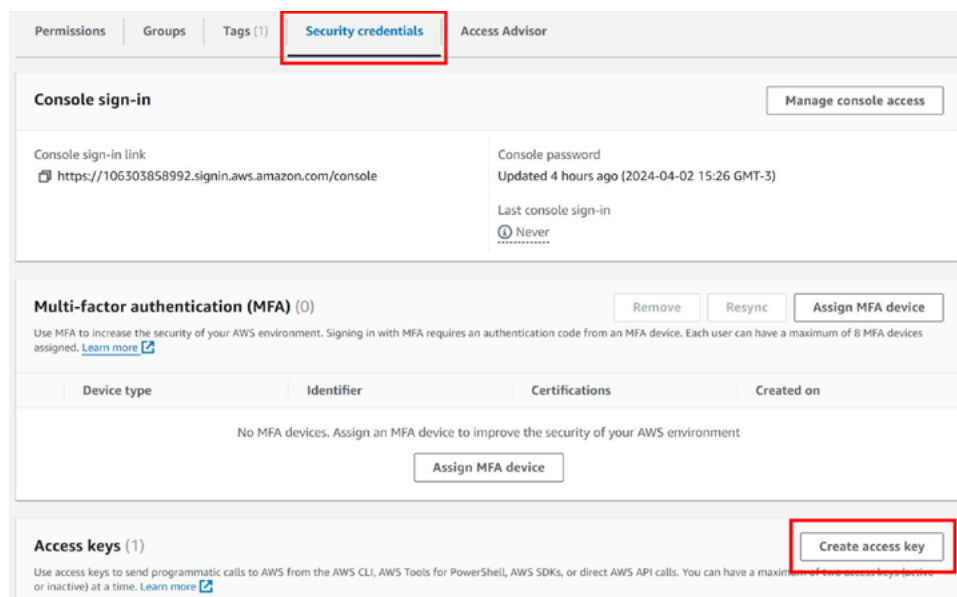
**Descrição da atividade 4:**

Se você seguiu todos os passos da atividade 3, você não só terá uma conta AWS, mas também um usuário configurado no serviço de gerenciamento de identidade e acesso na sua conta da Amazon. Uma vez logado na plataforma da AWS, siga os passos a seguir para configurar as credenciais e permissões de seu usuário.



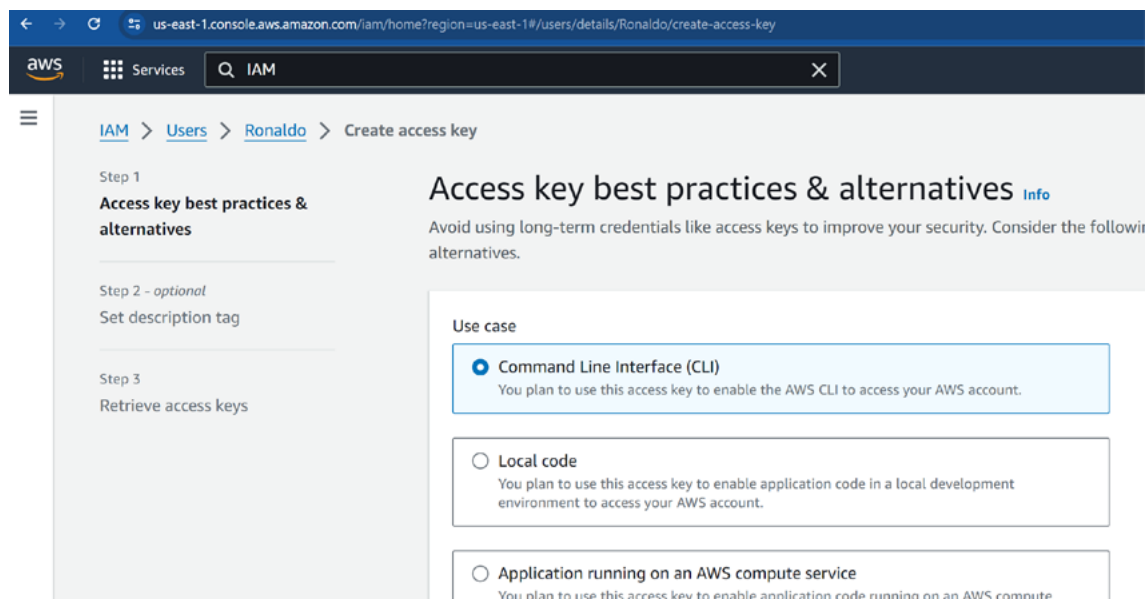
Fonte: Elaborada pelo autor, 2024.

1. Procure pelo serviço IAM e selecione-o na lista de serviços;
2. Na página do IAM, selecione o menu de usuários;
3. Acesse os detalhes de seu usuário, para configurar suas credenciais e permissões;
4. Caso não tenha um usuário, clique no botão de criação de usuário e informe um nome.



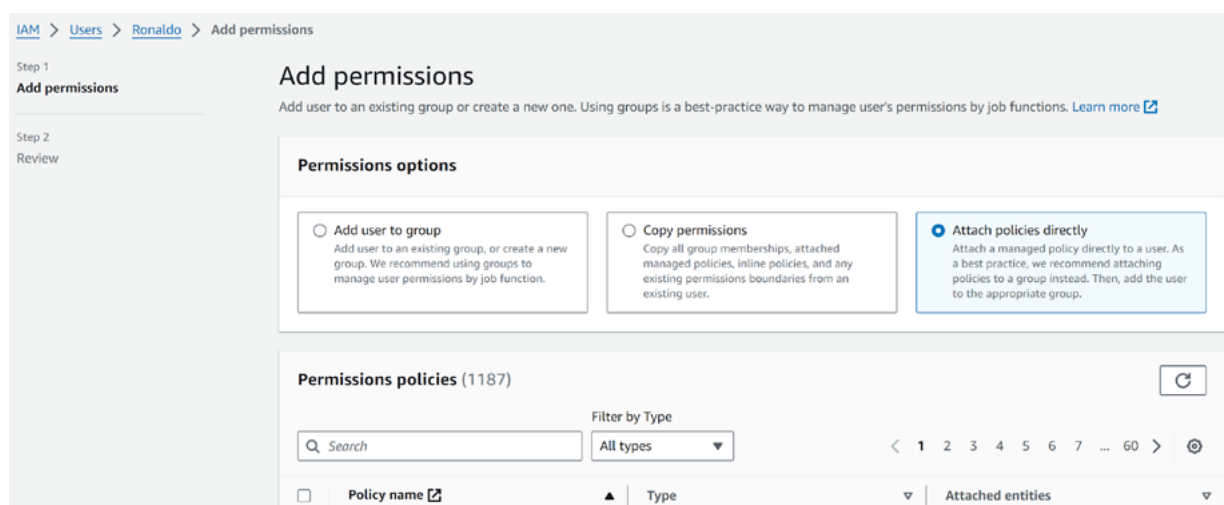
Fonte: Elaborada pelo autor, 2024.

Na tela de detalhes do usuário, você precisará acessar a aba “Security credentials” e escolher a opção de criar uma chave de acesso. Habilite a opção da interface de linha de comando, conforme mostra a imagem abaixo e faça o *download* do seu arquivo de credenciais. Lembre-se do diretório onde o arquivo ficará salvo, pois precisaremos, nas informações de AWS Access Key ID e AWS Secret Access Key, para configurar a ferramenta de linha de comando da Amazon.



Fonte: Elaborada pelo autor, 2024.

Finalmente, na aba de permissões, escolha a opção de adicionar permissões, para realizar os últimos passos de configuração de acesso. Escolha a opção anexar as políticas de acesso diretamente, “Attach policies directly”, conforme mostra a imagem a seguir.

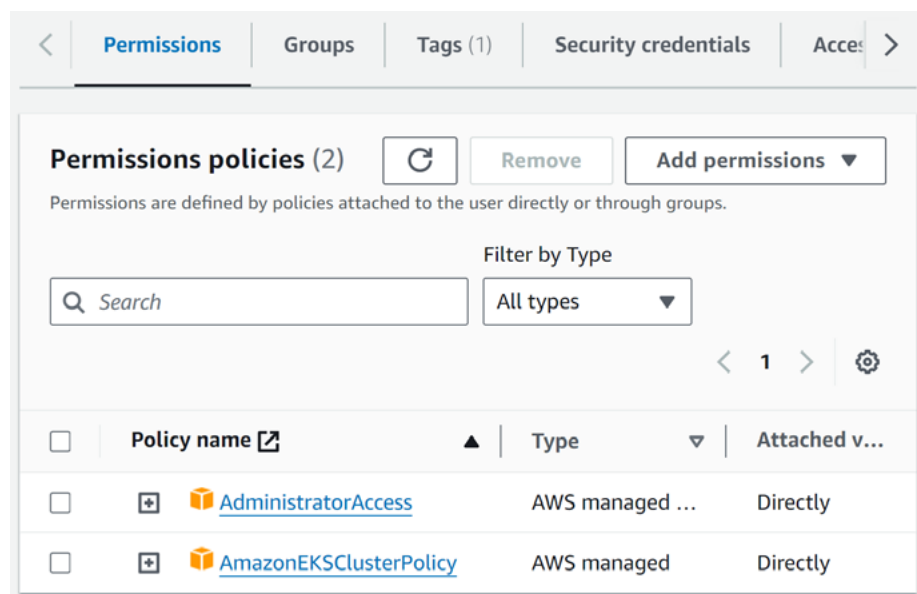


Fonte: Elaborada pelo autor, 2024.





Procure pelas políticas “AdministratorAccess” e “AmazonEKSClusterPolicy” e adicione-as às permissões do usuário. O resultado deve ser semelhante à figura abaixo.



Fonte: Elaborada pelo autor, 2024.

## Atividade 5: Instalação e configuração da Amazon Command Line Tool

### Objetivos:

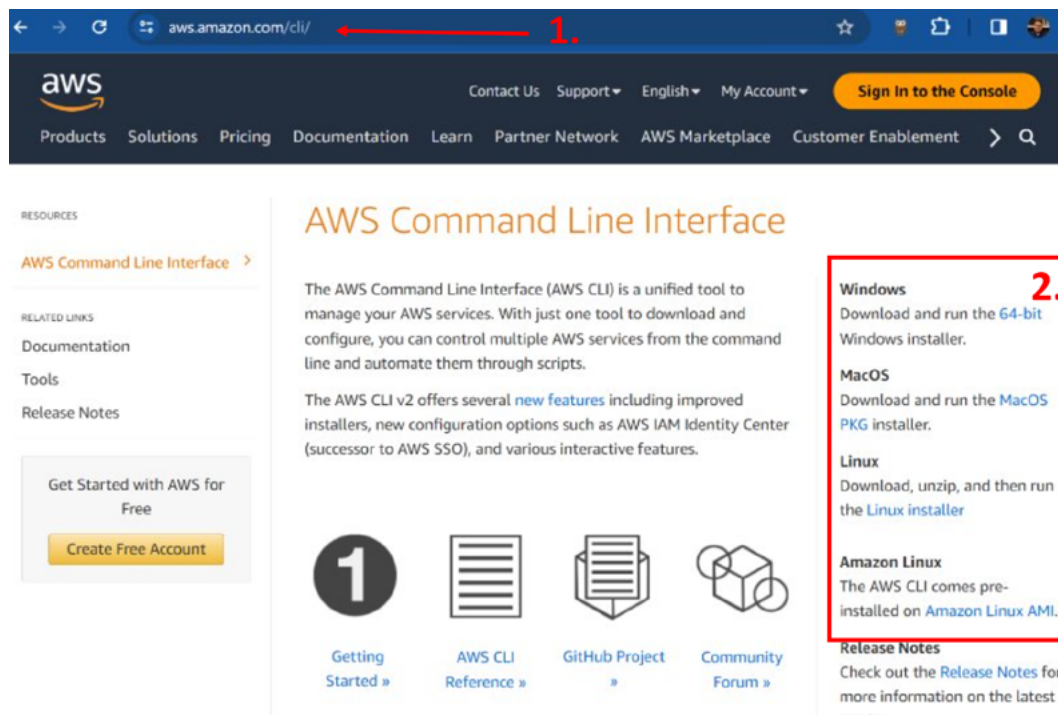
Instalar e configurar a ferramenta de linha de comando da Amazon (AWS Command Line Interface).

### Requisito:

- Atividade 4

### Descrição da atividade 5:

Para instalar e configurar a ferramenta de linha de comando da Amazon, a AWS Command Line Interface, siga os passos a seguir:



Fonte: Elaborada pelo autor, 2024.

1 Acesse o *site* oficial:

a. <https://aws.amazon.com/cli>

2. Identifique a opção de instalação correspondente ao seu sistema operacional e execute o instalador.

```
C:\Users\Ronaldo\Documents\Docker-workspace\eksctl_Windows_amd64>aws cli

usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

    aws help
    aws <command> help
    aws <command> <subcommand> help

aws: error: argument command: Invalid choice, valid choices are:

accessanalyzer          | account
acm                     | acm-pca
alexaforbusiness        | amp
amplify                 | amplifybackend
amplifyuibuilder        | apigateway
apigatewaymanagementapi | apigatewayv2
appconfig               | appconfigdata
appfabric               | appflow
appintegrations         | application-autoscaling
application-insights    | applicationcostprofiler
appmesh                 | apprunner
appstream               | appsync
```

Fonte: Elaborada pelo autor, 2024.

Após o processo de instalação, abra um terminal e escreva o comando “aws cli”, para ver uma lista de comandos disponíveis. Para configurar a conta Amazon na ferramenta de linha de comando, execute “aws configure” e informe o seu AWS Access Key ID, sua AWS Secret Access Key e o nome da região que deseja usar. O formato de output pode ficar vazio para usar o padrão, conforme a seguinte imagem:

```
C:\Users\Ronaldo\Documents\Docker-workspace\eksctl_Windows_amd64>aws configure
AWS Access Key ID [None]: AKIA[REDACTED]6J
AWS Secret Access Key [None]: mL[REDACTED]EW
Default region name [None]: us-east-1
Default output format [None]:

C:\Users\Ronaldo\Documents\Docker-workspace\eksctl_Windows_amd64>
```

Fonte: Elaborada pelo autor, 2024.

## Atividade 6: Instalação e configuração da ferramenta eksctl

### Objetivo:

Instalar a ferramenta eksctl.

### Requisito:

- Atividade 5

### Descrição da atividade 6:

Para instalar a ferramenta eksctl, siga os passos abaixo:

- 1 Acesse o *site* oficial:
  - a. <https://eksctl.io/installation/>
- 2 Identifique a opção de instalação correspondente ao seu sistema operacional;
- 3 Siga o processo de instalação.

Vale mencionar que, apesar de não ser necessário, os exemplos contidos neste caderno de atividades foram todos instalados em um mesmo diretório, a pasta onde as credenciais da Amazon foram salvas. Para verificar que a ferramenta foi instalada com sucesso, execute o comando “eksctl version”, para imprimir a versão da ferramenta.

```
C:\Users\Ronaldo\Documents\Docker-workspace\eksctl_Windows_amd64>eksctl version
0.175.0

C:\Users\Ronaldo\Documents\Docker-workspace\eksctl_Windows_amd64>
```

Fonte: Elaborada pelo autor, 2024.



## Atividade 7: Criação de um cluster na Amazon EKS, usando a ferramenta eksctl

### Objetivo:

Criar um *cluster* básico inicial.

### Requisito:

- Atividade 6

### Descrição da atividade 7:

Para criar um *cluster* básico inicial, utilizando o serviço EKS, execute o comando:

```
eksctl create cluster --name projeto --version 1.29 --region us-east-1 --nodegroup-name linux-nodes --node-type t2.micro --nodes 2
```

```
C:\Windows\System32\cmd.exe - eksctl create cluster --name projeto --region us-east-1

C:\Users\Ronaldo\Documents\Docker-workspace\eksctl_Windows_amd64>eksctl create cluster --name projeto

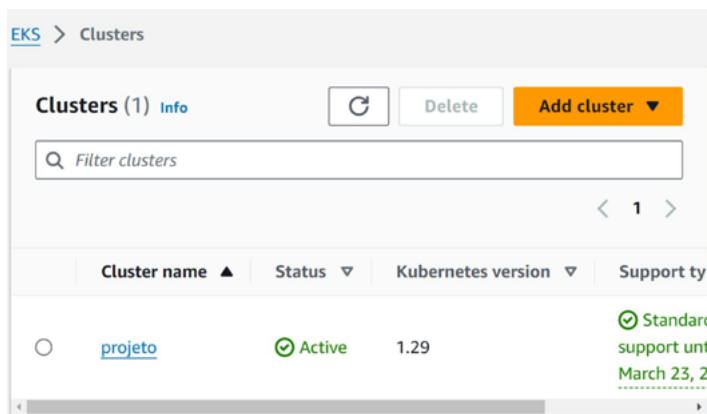
2024-04-02 16:45:57 [0] eksctl version 0.175.0
2024-04-02 16:45:57 [0] using region us-east-1
2024-04-02 16:45:58 [0] setting availability zones to [us-east-1b us-east-1a]
2024-04-02 16:45:58 [0] subnets for us-east-1b - public:192.168.0.0/19 private:192.168.64.0/19
2024-04-02 16:45:58 [0] subnets for us-east-1a - public:192.168.32.0/19 private:192.168.96.0/19
2024-04-02 16:45:58 [0] nodegroup "ng-14a64079" will use "" [AmazonLinux2/1.29]
2024-04-02 16:45:58 [0] using Kubernetes version 1.29
2024-04-02 16:45:58 [0] creating EKS cluster "projeto" in "us-east-1" region with managed nodes
2024-04-02 16:45:58 [0] will create 2 separate CloudFormation stacks for cluster itself and the initial
up
2024-04-02 16:45:58 [0] if you encounter any issues, check CloudFormation console or try 'eksctl util
--region=us-east-1 --cluster=projeto'
2024-04-02 16:45:58 [0] Kubernetes API endpoint access will use default of {publicAccess=true, privat
cluster "projeto" in "us-east-1"
2024-04-02 16:45:58 [0] CloudWatch logging will not be enabled for cluster "projeto" in "us-east-1"
2024-04-02 16:45:58 [0] you can enable it with 'eksctl utils update-cluster-logging --enable-types={S
PES-HERE (e.g. all)} --region=us-east-1 --cluster=projeto'
2024-04-02 16:45:58 [0]
2 sequential tasks: { create cluster control plane "projeto",
  2 sequential sub-tasks: {
    wait for control plane to become ready,
    create managed nodegroup "ng-14a64079",
  }
}
2024-04-02 16:45:58 [0] building cluster stack "eksctl-projeto-cluster"
2024-04-02 16:45:59 [0] deploying stack "eksctl-projeto-cluster"
2024-04-02 16:46:29 [0] waiting for CloudFormation stack "eksctl-projeto-cluster"
```

Fonte: Elaborada pelo autor, 2024.

Conforme mostra a figura apresentada, os diversos passos que teríamos que fazer são, na verdade, executados automaticamente, devido ao uso da eksctl. Uma vez que o comando termine de executar, você pode visualizar os nodes criados, através do comando:

```
eksctl get nodegroups --cluster projeto
```

Alternativamente, você pode visitar a plataforma da AWS, para utilizar a UI e verificar se o *cluster* foi criado com sucesso:



Fonte: Elaborada pelo autor, 2024.

Acesse a página do serviço EC2 e verá os nodes em execução. Uma vez que você termine de acessar os recursos, para evitar gastos imprevistos, execute o comando “*eksctl delete cluster --name projeto*” e confirme que os recursos, agora, estão sendo liberados:

```
C:\Users\Ronaldo\Documents\Docker-workspace\eksctl_Windows_amd64>eksctl delete cluster --name projeto
2024-04-02 16:56:32 [i] deleting EKS cluster "projeto"
2024-04-02 16:56:33 [i] deleted 0 Fargate profile(s)
2024-04-02 16:56:34 [i] kubeconfig has been updated
2024-04-02 16:56:34 [i] cleaning up AWS load balancers created by Kubernetes objects of Kind Service or Ingress
2024-04-02 16:56:37 [i] 1 task: { delete cluster control plane "projeto" [async] }
2024-04-02 16:56:37 [i] will delete stack "eksctl-projeto-cluster"
2024-04-02 16:56:37 [i] all cluster resources were deleted
C:\Users\Ronaldo\Documents\Docker-workspace\eksctl_Windows_amd64>
```

Fonte: Elaborada pelo autor, 2024.

## Atividade 8: (Desafio - Opcional) Implantação do projeto Spring Boot, utilizando o serviço Amazon EKS

### Objetivo:

Implantação do projeto Spring Boot.

### Requisitos:

- Atividades 2 e 7

### Descrição da atividade 8:

Esta atividade pode gerar gastos, pois nossa aplicação possui diversos serviços. Entretanto, tente utilizar os mesmos passos de criação de um *cluster* da atividade anterior, só que, desta vez, realize a implantação do projeto Spring Boot.