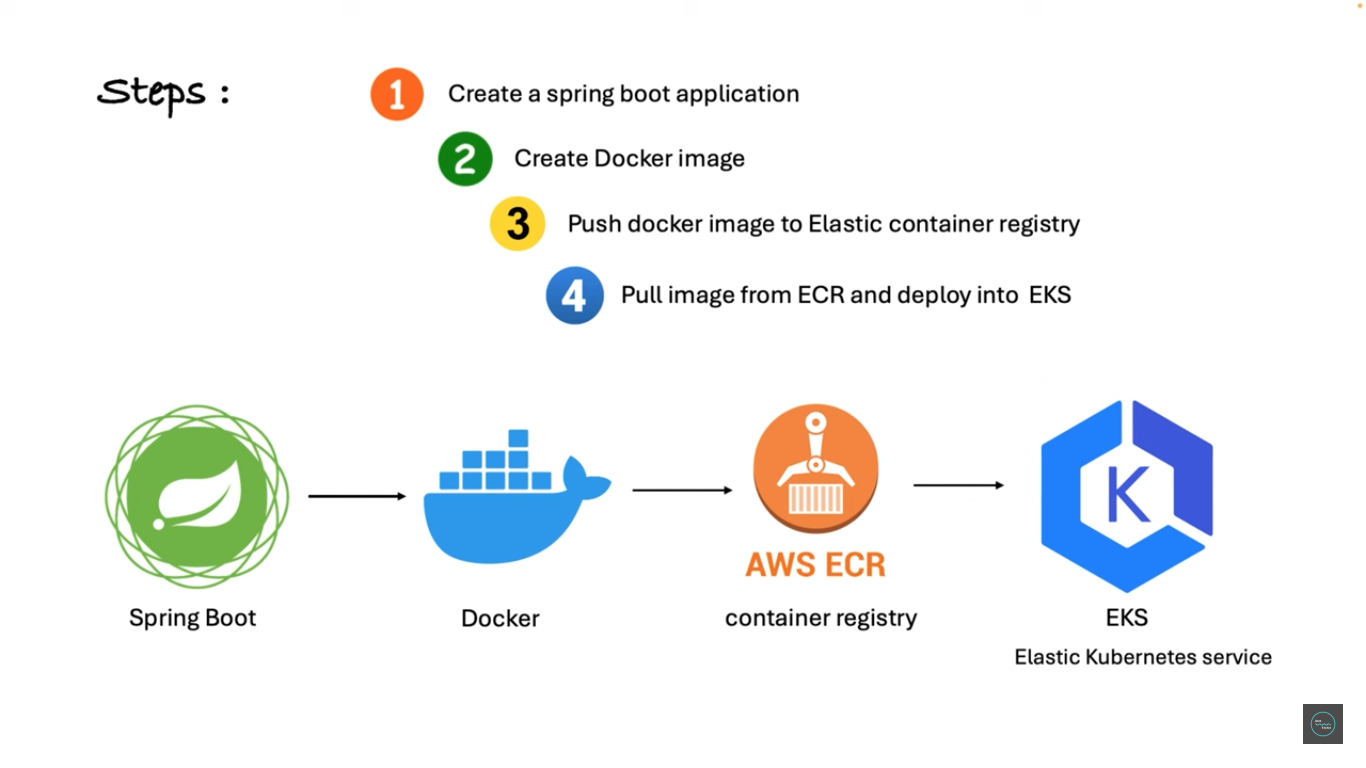
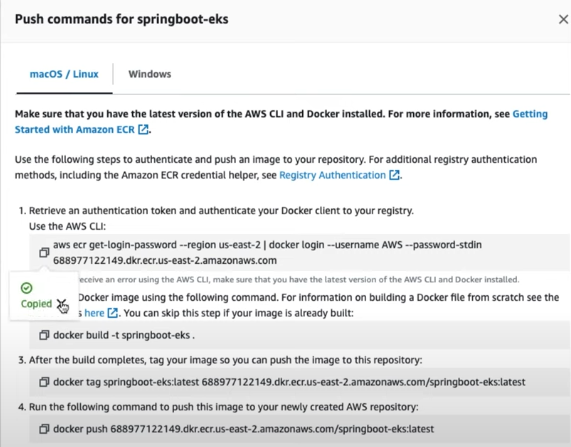
Here are the steps to deploy a Spring Boot application on AWS EKS:



1. Create a Spring Boot project and dockerize it: First, you need to create a simple Spring Boot project with a single endpoint. Then, create a Dockerfile to define the environment for your Spring Boot application and build a Docker image.
2. Push the Docker image to ECR: ECR stands for the Elastic Container Registry. It is a managed Docker registry service from AWS. Push your Docker image to ECR so it can be easily pulled and deployed to your EKS cluster.

Here are the AWS ECR commands used in the video:

* Login to ECR: aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 084889061199.dkr.ecr.us-east-2.amazonaws.com
* Build Docker image: docker build -t 084889061199.dkr.ecr.us-east-2.amazonaws.com/spring-boot-eks:latest .
* Tag Docker image: docker tag 084889061199.dkr.ecr.us-east-2.amazonaws.com/spring-boot-eks:latest 084889061199.dkr.ecr.us-east-2.amazonaws.com/spring-boot-eks:latest
* Push Docker image to ECR: docker push 084889061199.dkr.ecr.us-east-2.amazonaws.com/spring-boot-eks:latest



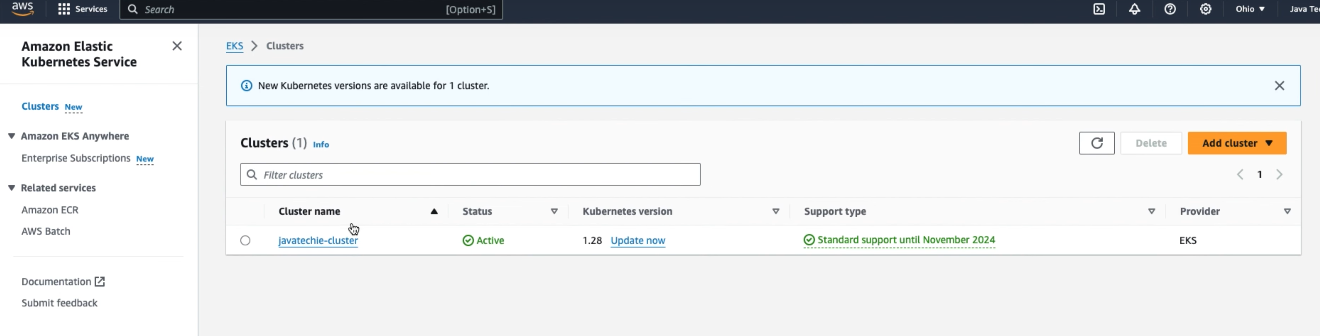
1. Create an EKS cluster: An EKS cluster is a managed Kubernetes service provided by AWS. Create an EKS cluster using the AWS Management Console or the eksctl command-line tool.

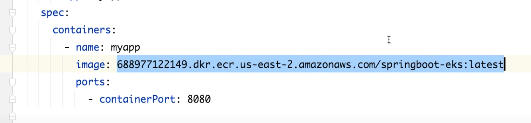
# Create EKS Cluster

```eksctl create cluster --name jt-cluster --version 1.28 --nodes=1 --node-type=t2.small --region us-east-2```

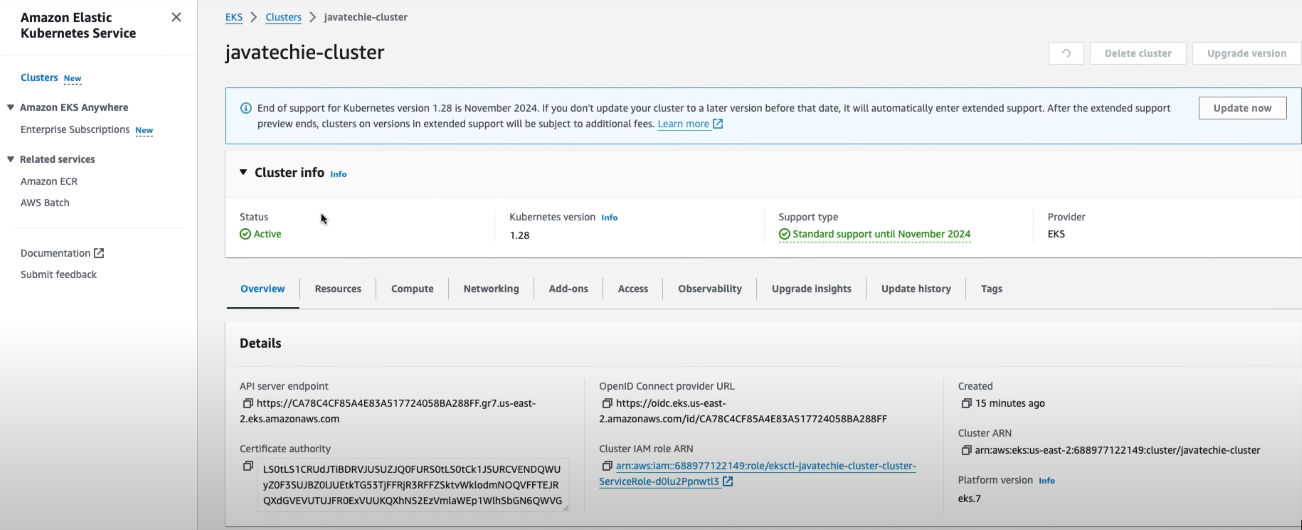
# Configure kubectl to Use the EKS Cluster

```aws eks --region us-east-2 update-kubeconfig --name javatechie-cluster```



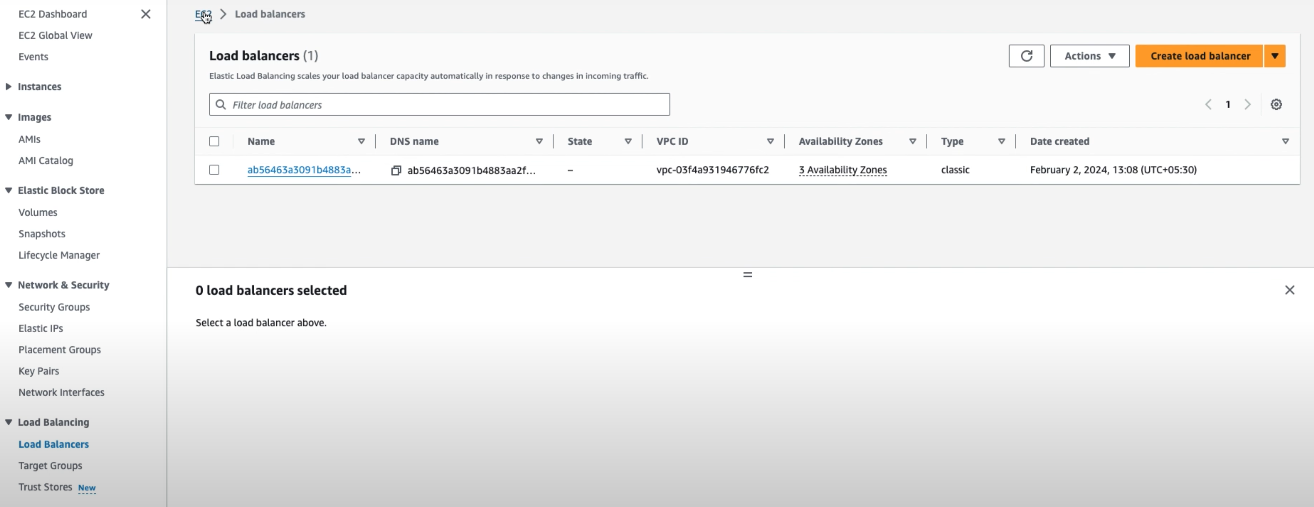


Above is image id in ecr for the application



1. Deploy the application to EKS: Define the Kubernetes manifests (deployment and service) to specify how your application should be deployed to the EKS cluster. Use the kubectl apply command to apply these manifests and deploy your application.

kubectl apply -f deployment.yaml



1. Access your application: Once deployed, you can access your Spring Boot application using the public IP address of the load balancer created by EKS.
2. (Optional) Delete the EKS cluster: After you have finished testing your application, you can delete the EKS cluster to avoid incurring charges. Use the eksctl delete cluster command to delete the cluster.

Additional notes:

* Make sure you have an AWS account and the necessary tools (AWS CLI, kubectl, Docker) installed on your machine.
* You can find more detailed information about each step in the video description and the JavaTechie's Medium blog.

