**Lesson 02 Demo 03**

**Setting up a Java-Based Elastic Beanstalk Environment**

**Objective:** To automate Platform as a Service (PaaS) deployment without the need to create infrastructure components and host supported programming language artifacts using AWS Elastic Beanstalk

**Tools required:** AWS Identity and Access Management (IAM) and AWS Elastic Beanstalk

**Prerequisites:** None

Steps to be followed:

1. Create an IAM role for AWS Elastic Beanstalk
2. Create a Java-based Elastic Beanstalk application and environment

**Step 1: Create an IAM role for AWS Elastic Beanstalk**

1. Sign in to the AWS Console from AWS Labsin the LMS portal

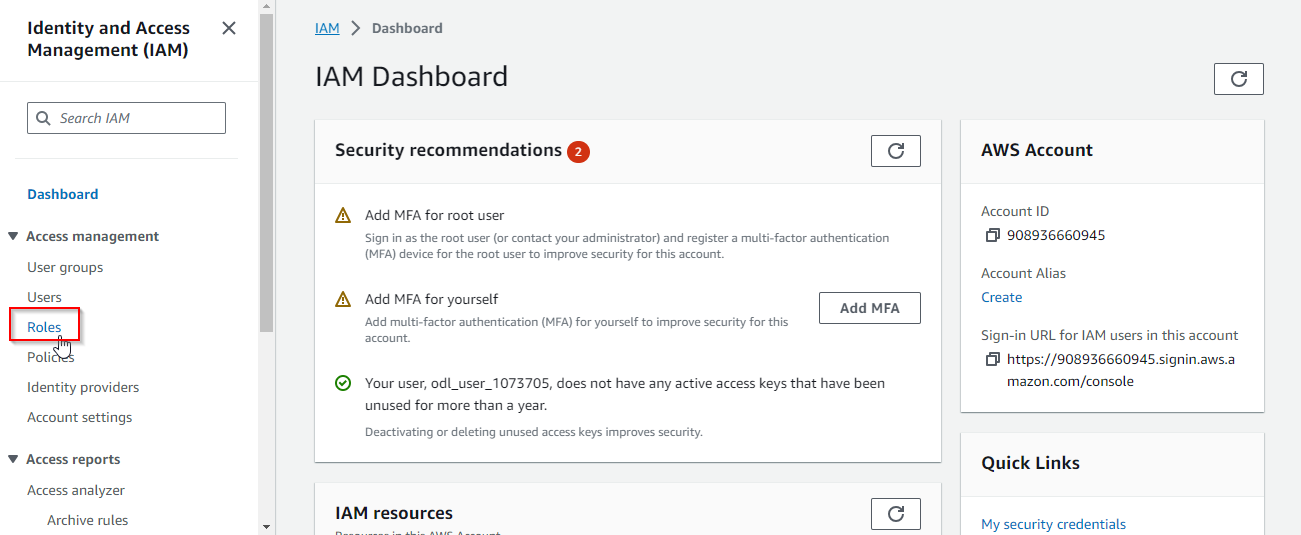
A screenshot of a computer

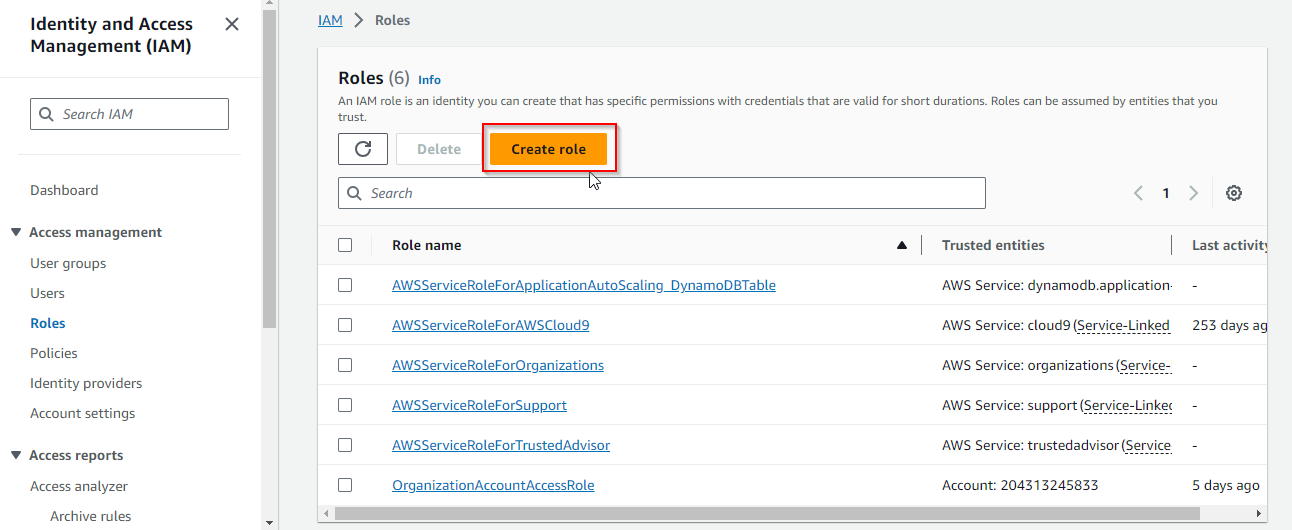
Description automatically generated

1. Once logged in to the AWS portal, searchfor and click on **IAM**

A screenshot of a computer

Description automatically generated

1. In the list on the left side, clickon **Roles**   
     
   

1. Clickon the **Create role** button  
     
   
2. In the **Use case** section, select **EC2** service from the drop-down list  
     
   A screenshot of a computer

   Description automatically generated

A screenshot of a computer

Description automatically generated

1. Click the **Next** button  
     
   A screenshot of a computer

   Description automatically generated

1. In the search bar, enter the policy name as **AdministratorAccess-AWSElasticBeanstalk**, and click on the checkboxto add the policy  
     
   A screenshot of a computer

   Description automatically generated
2. Click the **Next** button  
   A screenshot of a computer

   Description automatically generated

1. Enter the **Role name** as **elasticbeanstalk-instance-profile**  
     
   A screenshot of a computer

   Description automatically generated
2. Scroll to the bottom of the page and click the **Create role** button  
     
   A screenshot of a computer

   Description automatically generated

The role is created successfully as shown below:  
  
A screenshot of a computer

Description automatically generated

**Step 2: Create a Java-based Elastic Beanstalk application and environment**

1. Searchfor and click on **Elastic Beanstalk**  
     
   A screenshot of a computer

   Description automatically generated
2. On the top left corner, click on the list icon  
     
   A screenshot of a computer

   Description automatically generated
3. Click on **Applications**  
     
   A screenshot of a computer

   Description automatically generated
4. Now, click on the **Create application** button  
     
   A screenshot of a computer

   Description automatically generated
5. Enter the **Application name** as **CounterApplication**  
     
   A screenshot of a computer

   Description automatically generated
6. Click the **Create** button  
     
   A screenshot of a computer

   Description automatically generated
7. Now, click on the **Create new environment** button  
     
   A screenshot of a computer

   Description automatically generated

You will now be able to view the interface shown below:

A screenshot of a computer

Description automatically generated

* 1. In the **Environment** **information** section, enter the **Environment name**

A screenshot of a computer

Description automatically generated

* 1. In the **Platform** section, specify the platform for deployment. For a Java application, select the **Tomcat**-based platform as shown below:  
       
     A screenshot of a computer

     Description automatically generated
  2. Configure the infrastructure **Presets** based on your infrastructure requirements and

click on the **Next** button  
  
A screenshot of a computer code

Description automatically generated

* 1. Set up a service account and integrate it with your environment as shown below:

A screenshot of a computer

Description automatically generated

2.12 After setting up the service access, click on the **Next** button

A screenshot of a computer

Description automatically generated

* 1. Configure the networking settings as shown below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Note:** In case you have database configuration, you can enable the database and configure the same.

* 1. Scroll to the bottom of the page and click the **Next** button

A screenshot of a computer

Description automatically generated

* 1. Select optional settings, such as security group configurations, based on the

requirements

**Note**: If you are not aware of the requirements of the security group configuration, proceed further by enabling the **default** option as shown in the screenshot below:

A screenshot of a computer

Description automatically generated

* 1. Scroll to the bottom of the page and click on the **Next** button

A screenshot of a computer

Description automatically generated

* 1. Configure updates, monitoring, and logging details, and enable managed platform

updates for automatic platform updates according to the screenshots below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer program update

Description automatically generated

* 1. Scroll to the bottom of the page and click on the **Next** button

A screenshot of a computer

Description automatically generated

* 1. Review and validate all configurations

A screenshot of a computer

Description automatically generated

* 1. Once validated, click on the **Submit** button

A screenshot of a computer

Description automatically generated

You will then be able to see the interface shown below:  
  
A screenshot of a computer

Description automatically generated

* 1. Once the environment is launched successfully, click on the **Domain** **URL**

A screenshot of a computer

Description automatically generated

You will then be able to access the sample application deployed on Elastic Beanstalk as shown in the screenshot below:  
  
A screenshot of a computer

Description automatically generated

By following these steps, you have successfully set up a Java-based Elastic Beanstalk environment on AWS, which allows you to deploy and manage Java applications easily without the underlying infrastructure complexities.