**Capstone Project**

**​​​** **Deploy a Spring Boot Application on AWS Using CI/CD with ECR, ECS, and Fargate**

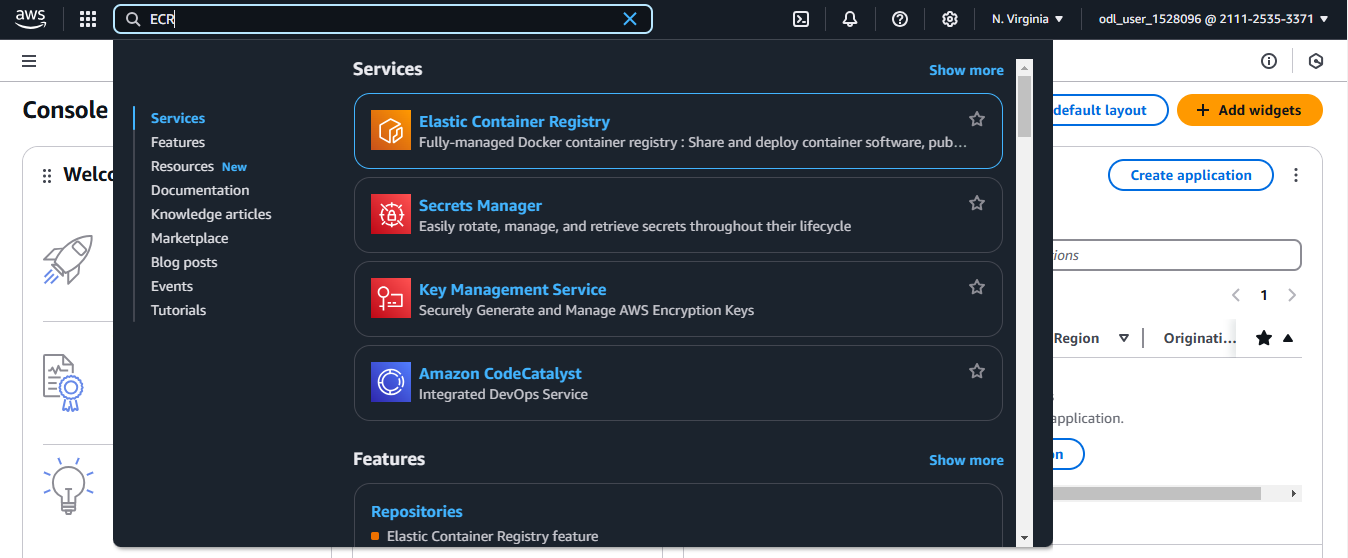
Steps to be followed:

1. Create an Amazon ECR repository
2. Configure the source repository
3. Create an IAM role and CodeBuild project
4. Configure IAM role permissions
5. Create an ECS cluster
6. Create and configure a task definition
7. Create a CodePipeline

**Step 1:**  **Create an Amazon ECR repository**

1. Log in to the AWS Management ConsoleA screenshot of a computer

   Description automatically generated
2. Navigate to **ECR** by searching for it in the services search bar and selecting the service



* 1. Click **Create**   
       
     A screenshot of a computer

     Description automatically generated
  2. Provide a name for your repository  
       
     A screenshot of a computer

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

     Description automatically generated
  4. Select the repository you created and click **View push commands**  
       
     A screenshot of a computer

     Description automatically generated
  5. Copy the command for pushing the Docker image to the repository  
       
     A screenshot of a computer

     Description automatically generated

**Step 2: Configure the source repository**

* 1. Fork the following repository:  
     **https://github.com/simplilearn10/aws-devops-springbootapp.git**  
     A screenshot of a computer

     Description automatically generated
  2. Open the **buildspec.yml** file in the repository  
       
     A screenshot of a computer

     Description automatically generated
  3. Insert the copied repository URI into line 10 of the **buildspec.yml** file  
     A screenshot of a computer

     Description automatically generated
  4. Click **Commit changes** to save the updates  
     A screenshot of a computer

     Description automatically generated
  5. Copy the repository URI  
     A screenshot of a computer

     Description automatically generated
  6. Paste the copied URI into line 12 of the **buildspec.yml** fileA screenshot of a computer

     Description automatically generated
  7. Click **Commit changes** again to finalizeA screenshot of a computer

     Description automatically generated

**Step 3: Create an IAM role and CodeBuild project**

1. Search for **CodeBuild** in the AWS Management Console and select the service

A screenshot of a computer

Description automatically generated

1. Click **Create project**

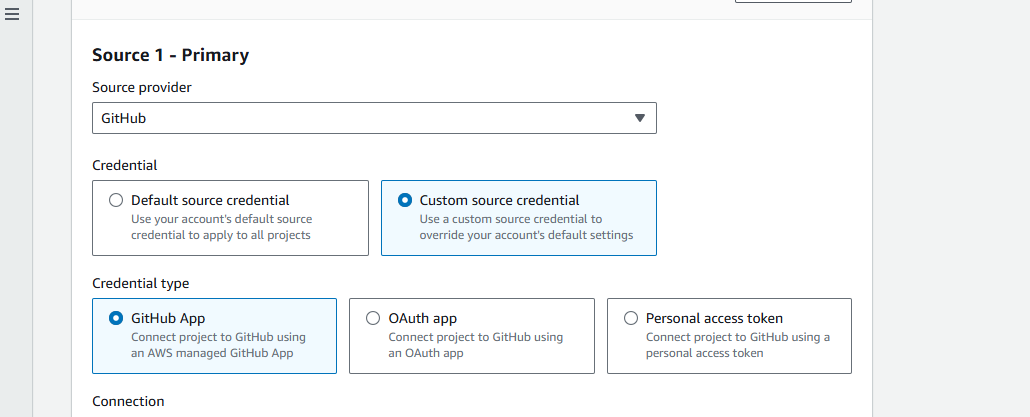
A screenshot of a computer

Description automatically generated

1. Provide a name for your project

A screenshot of a computer

Description automatically generated

1. Choose **GitHub** as the source provider  
   
2. Click **create a new GitHub connection**A screenshot of a computer

   Description automatically generated
3. Name the connection and click **Connect to GitHub**A screenshot of a computer

   Description automatically generated
4. Click **Install a new app**   
     
   A screenshot of a computer

   Description automatically generated
5. Provide your GitHub credentials and **Confirm** the connectionA screenshot of a computer

   Description automatically generated
6. Choose **Only select repositories**, select the forked repository, and save the selection  
   A screenshot of a computer screen

   Description automatically generated
7. Select the app installation and click **Connect**A screenshot of a computer

   Description automatically generated
8. Choose the repository in CodeBuildA screenshot of a computer

   Description automatically generated

1. Enter a name for the IAM role  
   A screenshot of a computer

   Description automatically generated
2. Specify **buildspec.yml** as the Buildspec file nameA screenshot of a computer

   Description automatically generated
3. Click **Create build project**A screenshot of a computer

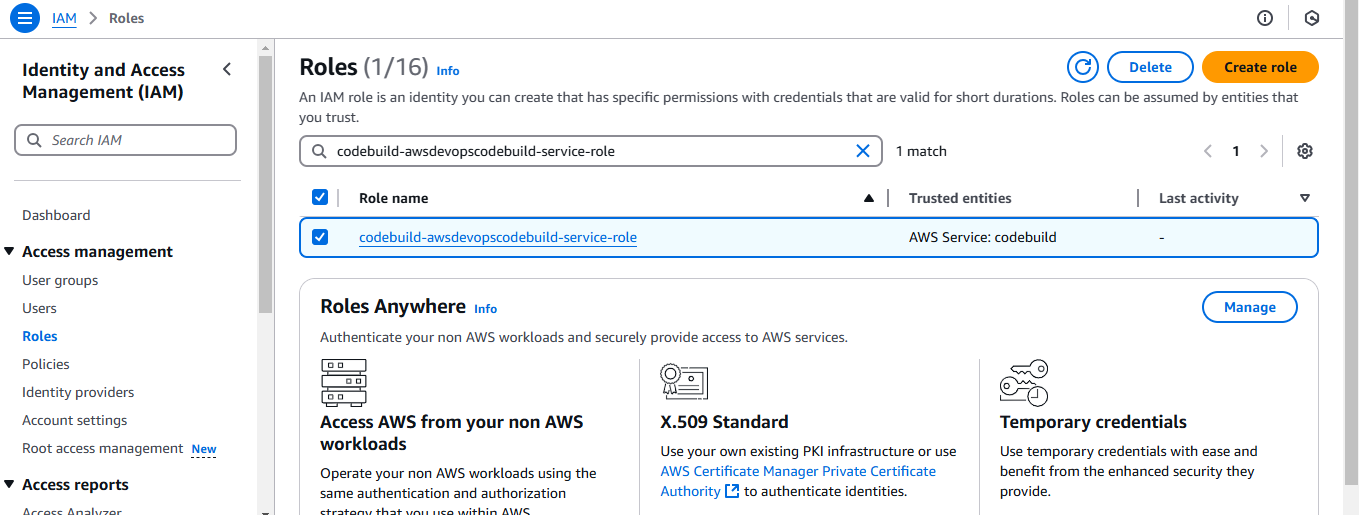
   Description automatically generated
4. Configure the IAM role if you encounter an error  
     
   A screenshot of a computer

   Description automatically generated

**Step 4: Configure IAM role permissions**

1. Search for **IAM** in the AWS Management Console and click on the service  
     
   A screenshot of a computer

   Description automatically generated
2. Click **Roles** in the left navigation panel   
     
   A screenshot of a computer

   Description automatically generated
3. Select the role you created for CodeBuild   
   
4. Click **Add permissions** and choose **Attach policies**A screenshot of a computer

   Description automatically generated
5. Click **AdministratorAccess** policy  
   A screenshot of a computer

   Description automatically generated
6. Search for and attach the **AmazonEC2ContainerRegistryPowerUser** policy  
   A screenshot of a computer

   Description automatically generated
7. Click **Add permissions**A screenshot of a computer

   Description automatically generated

The policy is attached successfully.A screenshot of a computer

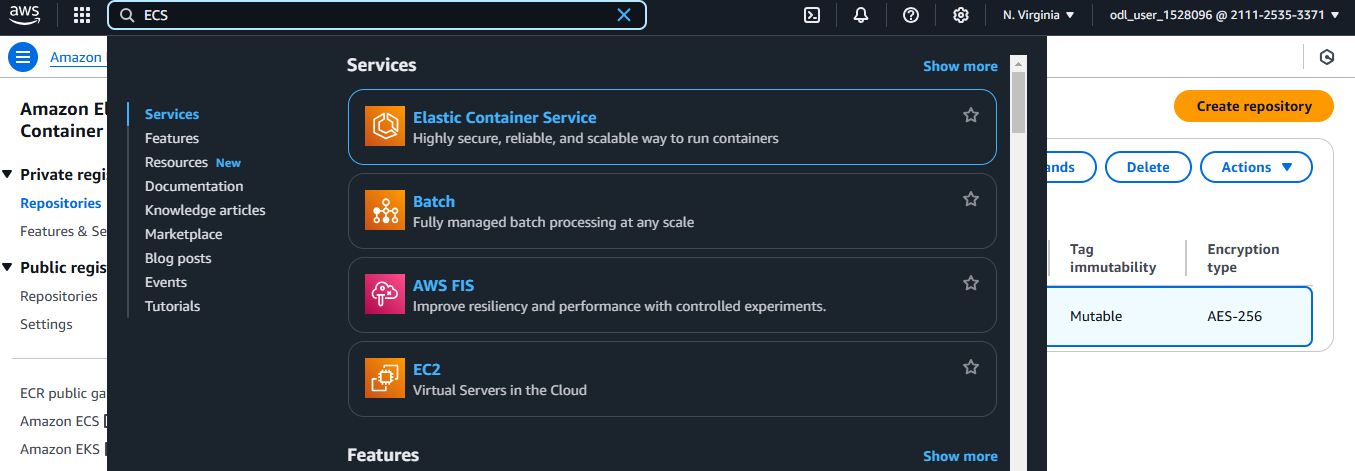
Description automatically generated

1. Navigate back to CodeBuild and attempt to create the build project againA screenshot of a computer

   Description automatically generated  
     
     
     
     
     
     
     
     
     
     
     
   The build is successful.  
     
   A screenshot of a computer

   Description automatically generated

**Step 5: Create an ECS cluster**

1. Search for **ECS (Elastic Container Service)** in the AWS Management Console  
     
   
2. Click **Get started**A black box with white text

   Description automatically generated
3. Enter a name for your cluster  
     
   A screenshot of a computer

   Description automatically generated
4. Choose **AWS Fargate (serverless)** as the launch typeA screenshot of a computer

   Description automatically generated
5. Click **Create**A screenshot of a computer

   Description automatically generatedThe cluster is created successfully.  
   A screenshot of a computer

   Description automatically generated

**Step 6: Create and configure a task definition**

1. Select the created ECS cluster  
     
   A screenshot of a computer

   Description automatically generated
2. Click **Create new task definition**  
   A screenshot of a chat

   Description automatically generated
3. Provide a name for the task definition  
   A screenshot of a computer

   Description automatically generated
4. Provide a name for the container, paste the Docker image URI, and set the port number to **8080**  
   A screenshot of a computer

   Description automatically generated
5. Click **Create** to save the task definition  
     
   A screenshot of a computer

   Description automatically generated  
     
   The task definition is created successfully.  
     
   A screenshot of a computer

   Description automatically generated
6. Navigate back to the cluster  
   A green and white box with text

   Description automatically generated
7. Scroll down to **Services** and click **Create**  
     
   A screenshot of a computer

   Description automatically generated
8. Choose **FARGATE** as the launch type  
     
   A screenshot of a computer

   Description automatically generated
9. Provide a family name for the service  
     
   A screenshot of a computer

   Description automatically generated
10. Enter a service name  
      
    A screenshot of a computer

    Description automatically generated
11. Configure the networking settings by creating a new security group  
    A screenshot of a computer

    Description automatically generated
12. Click **Create a new security group**A screenshot of a computer

    Description automatically generated
13. Set inbound rules to allow **All traffic** from **Anywhere**A screenshot of a computer

    Description automatically generated
14. Save the settings and click **Create**  
      
    A screenshot of a computer

    Description automatically generated

The task definition is created successfully.  
  
A screenshot of a computer

Description automatically generated

1. Click on the task   
     
   A screenshot of a computer

   Description automatically generated
2. Copy the public IP address for testing the deployed application  
     
   A screenshot of a computer

   Description automatically generated  
     
   The following landing page is displayed:  
     
   A screenshot of a computer

   Description automatically generated

**Step 7: Create a CodePipeline**

1. Search for **CodePipeline** in the AWS Management Console and select the service  
     
   A screenshot of a computer

   Description automatically generated
2. Click on **Create pipeline**A screenshot of a computer

   Description automatically generated
3. Choose **Build custom pipeline** as the creation option and click **Next**A screenshot of a computer

   Description automatically generated
4. Enter a name to the pipeline  
     
   A screenshot of a computer

   Description automatically generated
5. Click **Next** to continue  
     
   A screenshot of a computer

   Description automatically generated
6. Choose **GitHub (via GitHub App)** as the source provider and connect to GitHub  
     
   A screenshot of a computer

   Description automatically generated

1. Name the connection and connect to the GitHub repository  
     
   A screenshot of a computer

   Description automatically generated
2. Select the app installation and click **Connect**  
   A screenshot of a computer

   Description automatically generated
3. Select the GitHub connection and repository  
     
   A screenshot of a computer

   Description automatically generated
4. Keep other settings at their default values  
     
   A screenshot of a computer

   Description automatically generated
5. Click **Next**  
     
   A screenshot of a computer

   Description automatically generated
6. Choose **AWS CodeBuild** as the build provider and select the project you created  
     
   A screenshot of a computer

   Description automatically generated
7. Click **Next**  
     
   A screenshot of a computer

   Description automatically generated
8. Choose **Amazon ECS** as the deploy provider  
     
   A screenshot of a computer

   Description automatically generated
9. Select the ECS cluster and service created earlier  
     
   A screenshot of a computer

   Description automatically generated
10. Click **Next**  
      
    A screenshot of a computer

    Description automatically generated
11. Click **Create pipeline** to finalize  
      
    A screenshot of a computer program

    Description automatically generated

The pipeline is successfully created.  
  
A screenshot of a computer

Description automatically generated  
  
By following these steps, you have successfully deployed a Spring Boot application using a CI/CD pipeline with AWS ECR, CodeBuild, and ECS.