Ecomm Project – End-to-End Deployment Guide

1.Overview

This document explains step-by-step how to build and deploy the Ecomm web application on an EC2 instance with Tomcat 9 using AWS CodePipeline, CodeBuild, and CodeDeploy.

2. Prerequisites

- AWS Account with admin access
- EC2 instance (Amazon Linux 2 preferred)
- GitHub repository containing:
    - Ecomm.war
    - appspec.yml
    - buildspec.yml
    - scripts/
    - source/
    - pom.xml
    - tomcat-users.xml
- **github link**: https://github.com/venkataganesh48/java--deploy-tomcat.git

3. IAM Roles Setup

- **Role for EC2:**

    1.Go to IAM - Roles - Create role

    2.Select **EC2** as trusted entity.

    3.Attach policies:

    - AmazonEC2RoleforAWSCodeDeploy
    - AmazonS3FullAccess

    4.Name: EC2-CodeDeploy-Role

5.Attach this role to the EC2 instance.

- **Role for CodeDeploy :**

  1. Create new IAM role.
  2. Trusted entity: **CodeDeploy**
  3. Attach policy: AWSCodeDeployRole
  4. Name: CodeDeployServiceRole

- **Role for CodeBuild :**

  1. Create new IAM role.
  2. Trusted entity: **CodeBuild**
  3. Attach policies:
     - AmazonS3FullAccess
     - AWSCodeCommitReadOnly
     - AmazonEC2ContainerRegistryReadOnly

  4.Name: CodeBuildServiceRole

- **Role for CodePipeline:**

  1.Create IAM role.

  2.Click Custom Trusted entity:

  3.Attach:

```
{
  "Version": "2012-10-17"
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "codepipeline.amazonaws.com"
      },
        "Action": "sts:AssumeRole"
      }
    ]
  }
```

4.Attach policies:

- AmazonS3FullAccess
- AWSCodeDeployFullAccess
- AWSCodeBuildDeveloperAccess
- AWScodepipelinefullaccess

5.Name: CodePipelineServiceRole

4. EC2 Setup:

1. Launch Amazon Linux 2 EC2 instance in your region.
2. Attach EC2-CodeDeploy-Role IAM role.
3. Opening a Port in EC2 for 8080
4. Install CodeDeploy agent:

```
sudo yum update -y
sudo yum install -y ruby wget
cd /home/ec2-user
wget https://aws-codedeploy-ap-south-1.s3.ap-south1.amazonaws.com/latest/install
chmod +x ./install
sudo ./install auto
sudo systemctl enable codedeploy-agent
sudo systemctl start codedeploy-agent
```

6. Creating AWS Services

➢ **Create S3 Bucket:**

Create an S3 bucket for storing build artifacts.

- **Create CodeBuild Project**

Go to CodeBuild **:**

1**.** Create project: Ecomm project

2.Source: GitHub

3.Buildspec: buildspec.yml

4.select - existing Service role: CodeBuildServiceRole

- **Create CodeDeploy Application**

1.Go to CodeDeploy - Applications - Create application

2.Compute platform: EC2/On-premises

3.Name: EcommApp

- **Create CodeDeploy Deployment Group**

1.Inside the application - create a Deployment Group

2.Name: EcommDG

3.Service role:select existingrole – CodeDeployServiceRole

4.Deployment type: In-place

5.Environment: add tags

- **Create CodePipeline**

1.Go to CodePipeline - Create pipeline-build-custom pipeline

2.Pipeline name: EcommPipeline

3.Service role: CodePipelineServiceRole

4.Source stage: GitHub oauth app

5.Build stage: Select EcommBuild

6.Deploy stage: Select EcommApp and EcommDG

7.create pipeline

7.Access:

- **Tomcat Home** → http://<EC2-Public-IP>:8080
- **Manager App** → http://<EC2-Public-IP>:8080/manager/html
- **Ecomm App** → http://<EC2-Public-IP>:8080/Ecomm