**Prerequisites**

1. **GitHub Repo** with:
   * pom.xml
   * buildspec.yml
   * appspec.yml
   * scripts/
   * WAR output from Maven build
2. **EC2 Instance**:
   * Amazon Linux 2
   * Java + Tomcat installed
   * CodeDeploy agent installed and running
   * Proper **IAM role** with AmazonEC2RoleforAWSCodeDeploy
   * Instance tagged: e.g., Key: Name, Value: CodeDeployEC2
3. **IAM Roles**:
   * CodePipeline service role (AWSCodePipelineFullAccess)
   * CodeBuild service role (AWSCodeBuildDeveloperAccess)
   * CodeDeploy role (AWSCodeDeployRole)

**Step-by-Step Setup**

**Step 1: Create CodeDeploy Application and Deployment Group**

1. Go to **AWS Console → CodeDeploy**
2. Choose **Applications → Create application**
   * **Name**: JSPDeployApp
   * **Compute Platform**: EC2/On-premises
3. After creation, choose **Create Deployment Group**
   * **Name**: JSPDeployGroup
   * **Service Role**: Select CodeDeployRole (with AWSCodeDeployRole)
   * **Deployment Type**: In-place
   * **Environment Configuration**:
     + **Tag Key**: Name
     + **Value**: CodeDeployEC2 (or whatever your EC2 tag is)
   * **Deployment Settings**: Choose defaults

**Step 2: Create CodeBuild Project**

1. Go to **AWS Console → CodeBuild**
2. Click **Create build project**
   * **Name**: JSPBuildProject
   * **Source Provider**: GitHub → Connect your GitHub and select the repo
   * **Environment**:
     + Managed image
     + OS: Amazon Linux 5
     + Runtime: Standard
     + Image: aws/codebuild/standard:5.0
     + Service Role: Create or select an IAM role (e.g., CodeBuildRole)
   * **Buildspec**: Use buildspec.yml from the source
   * **Artifacts**:
     + Type: Amazon S3
     + Bucket: Choose/create bucket (uncheck "enable cache")

**Step 3: Create CodePipeline**

1. Go to **AWS Console → CodePipeline**
2. Click **Create pipeline**
   * **Name**: JSPPipeline
   * **Service Role**: Create a new role or use existing (CodePipelineRole)

**Source Stage:**

* **Source Provider**: GitHub
* **Repository**: Select your repo and branch
* **Change detection**: GitHub webhooks

**Build Stage:**

* **Build Provider**: CodeBuild
* **Project**: JSPBuildProject

**Deploy Stage:**

* **Deploy Provider**: AWS CodeDeploy
* **Application Name**: JSPDeployApp
* **Deployment Group**: JSPDeployGroup

1. Click **Create pipeline**

**Deployment Flow**

* GitHub commit → triggers CodePipeline
* CodeBuild packages app into WAR
* Artifacts passed to CodeDeploy
* CodeDeploy installs WAR to /opt/tomcat/webapps/ on EC2

**Verify**

* SSH into your EC2 instance and confirm WAR is under /opt/tomcat/webapps/
* Access http://<EC2-Public-IP>:8080/hello-jsp/
* Output: **"Hello World from JSP"**