**Lab 2: CI/CD with AWS CodePipeline & CodeBuild (CloudShell-Based Lab Manual)**

---

## Lab Title:

CI/CD Pipeline for a Web App using AWS CodePipeline and CodeBuild

## Objective:

Set up a complete Continuous Integration and Continuous Deployment (CI/CD) pipeline using AWS CodePipeline, CodeBuild, and GitHub as the source repository.

## Duration:

2 hours

## Pre-requisites:

- AWS Free Tier account with access to CodePipeline, CodeBuild, S3, IAM
- GitHub account with a sample web app repo (HTML or static site preferred)
- GitHub personal access token (PAT)
- Basic knowledge of Git and terminal operations

---

# Part A: Preparation (30 mins)

## 1. Launch AWS CloudShell

- Open: https://console.aws.amazon.com/cloudshell
- Region: Use `us-east-1` or another CodePipeline-supported region

## 2. Clone a Sample Web App

In CloudShell:

```
cd ~/
git clone https://github.com/YOUR_USERNAME/sample-web-app.git
cd sample-web-app
```

Make sure this repo has an `index.html` or static website files.

## 3. Push Your App to GitHub (if not already hosted)

```
git remote set-url origin https://github.com/YOUR_USERNAME/sample-web-app.git
git push origin main
```

## 4. Create an S3 Bucket (for CodePipeline artifacts)

```
aws s3 mb s3://your-cicd-artifacts-<unique-id>
```

Replace `<unique-id>` with something unique (e.g., your initials + date).

---

# Part B: IAM Role Setup (10 mins)

## 1. Create IAM Role for CodeBuild

Go to IAM Console:

- Create role > AWS service > CodeBuild
- Attach policy: **`AmazonS3FullAccess`**, **`CloudWatchLogsFullAccess`**, **`AmazonEC2ContainerRegistryReadOnly`**
- Name: `CodeBuildServiceRole`

## 2. Create IAM Role for CodePipeline

- Create role > AWS service > CodePipeline
- Attach policy: **`AWSCodePipelineFullAccess`**, **`AmazonS3FullAccess`**
- Name: `CodePipelineServiceRole`

---

# Part C: CodeBuild Project Setup (20 mins)

## 1. Create a `buildspec.yml` file in project root:

```
echo "version: 0.2
phases:
  build:
    commands:
      - echo Build started
      - echo Build complete
artifacts:
  files:
    - '**/*'" > buildspec.yml
```

Commit and push:

```
git add buildspec.yml
git commit -m "Add buildspec for CodeBuild"
git push
```

## 2. Create CodeBuild Project

Go to AWS Console > CodeBuild:

- Project name: `SampleWebAppBuild`
- Source: GitHub (connect using OAuth or PAT)
- Environment: Managed image (Amazon Linux, standard)

- Service Role: Select `CodeBuildServiceRole`
- Buildspec: Use buildspec.yml from source repo

---

# Part D: Create CodePipeline (30 mins)

## 1. Go to CodePipeline Console

- Create pipeline > Name: `SampleWebAppPipeline`
- Service Role: `CodePipelineServiceRole`
- Artifact Store: Use previously created S3 bucket

## 2. Add Source Stage

- Source Provider: GitHub (use PAT if OAuth fails)
- Repository: Select your `sample-web-app`
- Branch: `main`

## 3. Add Build Stage

- Provider: AWS CodeBuild
- Project: `SampleWebAppBuild`

## 4. Deploy Stage (optional)

For static website deployment, you can:

- Create another S3 bucket: `aws s3 mb s3://my-deploy-bucket`
- Enable static website hosting
- Use a Lambda or manual step to move artifacts

---

# Part E: Test Your CI/CD Pipeline (15 mins)

## 1. Make a change in your code:

```
echo "<h2>Updated on $(date)</h2>" >> index.html
git add index.html
git commit -m "Update index with timestamp"
git push
```

## 2. Observe Pipeline Execution

- Go to CodePipeline > View execution
- Each stage should show green if successful