Lab 4: CI/CD for Node.js Application using AWS CodePipeline & Elastic Beanstalk

---

Lab Title:

Deploy a Node.js Web App to Elastic Beanstalk using CodePipeline

Objective:

Create a simple Node.js application, push it to GitHub, and set up an end-to-end CI/CD pipeline using AWS CodePipeline and Elastic Beanstalk.

Duration:

2 hours

Pre-requisites:

- AWS Free Tier account
- GitHub account
- Basic knowledge of Node.js

Part A: Create Sample Node.js App (15 mins)

1. Launch CloudShell and Create Project Folder

```
mkdir ~/node-lab && cd ~/node-lab
```

2. Initialize Node.js App

```
npm init -y
```

3. Create Basic App Server

Create a file called `app.js`:

```
echo "
const express = require('express');
const app = express();
const PORT = process.env.PORT || 3000;

app.get('/', (req, res) => {
  res.send('<h1>Node.js App Deployed via Elastic Beanstalk</h1>');
});

app.listen(PORT, () => console.log(`App running on port ${PORT}`));
" > app.js
```

4. Install Express

```
npm install express
```

5. Create `package.json` start script

Edit `package.json` and set:

```
"scripts": {
  "start": "node app.js"
}
```

6. Create `.ebextensions` & Config Files

Create a file `.ebextensions/nodecommand.config`:

```
mkdir .ebextensions
echo "option_settings:
  aws:elasticbeanstalk:container:nodejs:
    NodeCommand: 'npm start'" > .ebextensions/nodecommand.config
```

Create a `Procfile`:

```
echo "web: node app.js" > Procfile
```

Part B: Push to GitHub (15 mins)

1. Initialize Git and Push

```
git init
git remote add origin https://github.com/YOUR_USERNAME/node-lab.git
git add .
git commit -m "Initial commit"
git push -u origin main
```

Part C: Create Elastic Beanstalk Environment (15 mins)

1. Create Elastic Beanstalk App and Environment

- Go to AWS Console → Elastic Beanstalk
- Create Application: `node-lab`
- Platform: Node.js
- Environment type: Web server
- Upload a ZIP if required for first deploy (you can use a dummy `index.js` temporarily)

2. Note the Beanstalk Environment Name

(e.g., `node-lab-env`)

Part D: Create CodePipeline (30 mins)

1. Start New Pipeline

- Source: GitHub → your repo → branch: `main`
- Build: Skip/No Build (CodeBuild is optional for Node.js static deploy)
- Deploy provider: **Elastic Beanstalk**
- Application name: `node-lab`
- Environment name: `node-lab-env`

2. IAM Role Permissions

Ensure the pipeline role can call `elasticbeanstalk:*` and `s3:*`.

Part E: Test Pipeline Execution (10 mins)

1. Update a File in Your App

```
echo "<p>Updated on $(date)</p>" >> README.md
git add README.md
git commit -m "Update timestamp"
git push
```

2. Watch CodePipeline Execute

- Navigate to CodePipeline → your pipeline
- All stages (Source → Deploy) should turn green
- Go to Elastic Beanstalk → App URL should now reflect the update

---

Summary

Key Learnings:

- Created and containerized a Node.js web app
- Integrated GitHub with CodePipeline
- Automated deployment to Elastic Beanstalk
- Understood how `Procfile`, `.ebextensions`, and start scripts control app behavior

---