To deploy a **Node.js application** to **Amazon EC2** using **AWS CodePipeline** and **CloudFormation**, we’ll build an end-to-end CI/CD pipeline that includes:

**✅ Overview of the Pipeline Stages**

1. **Source Stage** – Pull code from GitHub or CodeCommit.
2. **Build Stage** – Use CodeBuild to install dependencies, run tests, and package the app.
3. **Deploy Stage** – Use CodeDeploy to deploy to EC2.
4. **Infrastructure as Code** – Use CloudFormation to define all resources including EC2, CodePipeline, CodeBuild, CodeDeploy, IAM roles, etc.

**🏗️ CloudFormation Template Structure**

Here’s a high-level breakdown of what the CloudFormation template should include:

**1. EC2 Instance**

* Amazon Linux 2
* IAM Role with SSM and CodeDeploy permissions
* UserData script to install CodeDeploy agent

**2. CodePipeline**

* Source: GitHub or CodeCommit
* Build: CodeBuild project
* Deploy: CodeDeploy application and deployment group

**3. CodeBuild Project**

* Includes buildspec.yml for build instructions

**4. CodeDeploy Application**

* Deployment group targeting EC2 instance

**📄 buildspec.yml – CodeBuild Instructions**

This file goes in the root of your Node.js repo.

version: 0.2

phases:

install:

runtime-versions:

nodejs: 18

commands:

- echo Installing dependencies...

- npm install

build:

commands:

- echo Build started on `date`

- npm run build || echo No build step

post\_build:

commands:

- echo Build completed on `date`

- zip -r app.zip . -x node\_modules/\\*

- mkdir -p artifacts

- mv app.zip artifacts/

artifacts:

files:

- artifacts/app.zip

**📄 appspec.yml – CodeDeploy Instructions**

This file tells CodeDeploy how to deploy the app on EC2.

version: 0.0

os: linux

files:

- source: artifacts/app.zip

destination: /home/ec2-user/node-app

hooks:

AfterInstall:

- location: scripts/install\_dependencies.sh

timeout: 300

runas: ec2-user

ApplicationStart:

- location: scripts/start\_server.sh

timeout: 300

runas: ec2-user

You’ll need to include the scripts/ folder in your repo with the shell scripts.

**🧱 CloudFormation Template Snippet (CodePipeline Section)**

Here’s a simplified snippet for the CodePipeline resource:

Resources:

MyPipeline:

Type: AWS::CodePipeline::Pipeline

Properties:

RoleArn: !GetAtt PipelineRole.Arn

Stages:

- Name: Source

Actions:

- Name: SourceAction

ActionTypeId:

Category: Source

Owner: AWS

Provider: CodeCommit

Version: 1

OutputArtifacts:

- Name: SourceOutput

Configuration:

RepositoryName: my-node-repo

BranchName: main

- Name: Build

Actions:

- Name: BuildAction

ActionTypeId:

Category: Build

Owner: AWS

Provider: CodeBuild

Version: 1

InputArtifacts:

- Name: SourceOutput

OutputArtifacts:

- Name: BuildOutput

Configuration:

ProjectName: !Ref MyCodeBuildProject

- Name: Deploy

Actions:

- Name: DeployAction

ActionTypeId:

Category: Deploy

Owner: AWS

Provider: CodeDeploy

Version: 1

InputArtifacts:

- Name: BuildOutput

Configuration:

ApplicationName: !Ref MyCodeDeployApp

DeploymentGroupName: !Ref MyDeploymentGroup

**QUESTION**

Resources:

MyPipeline:

Type: AWS::CodePipeline::Pipeline

Properties:

RoleArn: !GetAtt PipelineRole.Arn

Stages:

- Name: Source

Actions:

- Name: SourceAction

ActionTypeId:

Category: Source

Owner: AWS

Provider: CodeCommit

Version: 1

OutputArtifacts:

- Name: SourceOutput

Configuration:

RepositoryName: my-node-repo

BranchName: main

what is the sourceouput here in OutputArtifacts

- Name: Build

Actions:

- Name: BuildAction

ActionTypeId:

Category: Build

Owner: AWS

Provider: CodeBuild

Version: 1

InputArtifacts:

- Name: SourceOutput

OutputArtifacts:

- Name: BuildOutput

Configuration:

ProjectName: !Ref MyCodeBuildProject

what is the sourceOutput again here in Inputartifacts and what is BuildOutput in outputArtifacts

Name: Deploy

Actions:

- Name: DeployAction

ActionTypeId:

Category: Deploy

Owner: AWS

Provider: CodeDeploy

Version: 1

InputArtifacts:

- Name: BuildOutput

Configuration:

ApplicationName: !Ref MyCodeDeployApp

DeploymentGroupName: !Ref MyDeploymentGroup

what is the BuildOuput here again in InputArtifacts ?

Excellent question! Let's walk through what SourceOutput and BuildOutput mean in the context of **AWS CodePipeline**, and how they are passed between stages.

**🔄 Artifact Flow in CodePipeline**

**📦 1. Source Stage**

OutputArtifacts:

- Name: SourceOutput

* **What is SourceOutput?**
  + This is the name of the artifact that contains the source code pulled from the repository (e.g., CodeCommit).
  + It includes all the files from your repo, such as buildspec.yml, appspec.yml, your Node.js code, etc.
  + This artifact is passed to the next stage (Build) as an **input**.

**🛠️ 2. Build Stage**

InputArtifacts:

- Name: SourceOutput

OutputArtifacts:

- Name: BuildOutput

* **What is SourceOutput here?**
  + It refers to the artifact produced by the Source stage.
  + CodeBuild uses this artifact to run the build process defined in buildspec.yml.
* **What is BuildOutput?**
  + This is the artifact produced by CodeBuild after executing the build steps.
  + It typically includes:
    - The zipped deployment package (app.zip)
    - appspec.yml
    - Any other files needed for deployment
  + This artifact is passed to the Deploy stage.

**🚀 3. Deploy Stage**

InputArtifacts:

- Name: BuildOutput

* **What is BuildOutput here?**
  + It refers to the artifact produced by the Build stage.
  + CodeDeploy uses this artifact to deploy the application to EC2.
  + It expects this artifact to contain:
    - appspec.yml at the root
    - Any referenced scripts and application files

**🔁 Summary of Artifact Flow**

| **Stage** | **Input Artifact** | **Output Artifact** | **Purpose** |
| --- | --- | --- | --- |
| **Source** | **—** | **SourceOutput** | **Pulls code from repo** |
| **Build** | **SourceOutput** | **BuildOutput** | **Builds and packages app** |
| **Deploy** | **BuildOutput** | **—** | **Deploys app to EC2** |

**About Buildspec VS Appspec file**

Great follow-up! Let's clarify **where the Node.js application is actually run** in the CI/CD pipeline:

**🧱 Breakdown: buildspec.yml vs appspec.yml**

**🔧 buildspec.yml (Used in CodeBuild)**

* Purpose: **Build and prepare** the application.
* Typical tasks:
  + Install dependencies (npm install)
  + Run tests (npm test)
  + Compile code (npm run build)
  + Package the app (zip, etc.)

✅ **It does NOT run the application**. It only prepares the app for deployment.

**🚀 appspec.yml (Used in CodeDeploy)**

* Purpose: **Deploy and run** the application on EC2.
* This is where the app is actually started.

Here’s how it works:

hooks:

AfterInstall:

- location: scripts/install\_dependencies.sh

timeout: 300

runas: ec2-user

ApplicationStart:

- location: scripts/start\_server.sh

timeout: 300

runas: ec2-user

**✅ start\_server.sh (inside your repo)**

This script is where you run the Node.js app:

#!/bin/bash

cd /home/ec2-user/node-app

npm install

npm start

Or if you use a specific entry point:

node index.js

**🔁 Summary**

| **File** | **Tool** | **Purpose** | **Runs App?** |
| --- | --- | --- | --- |
| buildspec.yml | CodeBuild | Build, test, package | ❌ No |
| appspec.yml | CodeDeploy | Deploy and start app | ✅ Yes |

Great question! Here's why **buildspec.yml** and **appspec.yml** are placed in the **root** of your repository:

**📄 Why buildspec.yml Must Be in the Root**

**✅ Used by AWS CodeBuild**

* CodeBuild looks for buildspec.yml in the **root directory** of the source artifact by default.
* If it's not there, you must explicitly specify the path in the CodeBuild project configuration (which adds complexity).

🔧 You *can* place it elsewhere (e.g., pipeline/dev/buildspec.yml), but then you must configure CodeBuild to use that path:

Source:

BuildSpec: pipeline/dev/buildspec.yml

**📄 Why appspec.yml Must Be in the Root**

**✅ Used by AWS CodeDeploy**

* CodeDeploy expects appspec.yml to be at the **root of the deployment artifact** (e.g., the ZIP file created in buildspec.yml).
* If it's not at the root, CodeDeploy will fail with an error like: > AppSpec file does not exist within the deployment archive

🔧 You *cannot* configure CodeDeploy to look elsewhere for appspec.yml.

**🧠 Summary**

| **File** | **Required Location** | **Why?** |
| --- | --- | --- |
| buildspec.yml | Root (or specify path in CodeBuild) | CodeBuild default behavior |
| appspec.yml | Root of deployment artifact | CodeDeploy requirement |