Project: AWS DevOps – CI/CD Pipeline for Sample Python App

🔧 Overview

This project demonstrates a complete CI/CD pipeline on AWS to automate the build and deployment of a sample Python application using the following AWS services:

- **CodePipeline**: Automates the end-to-end CI/CD workflow.
- **CodeCommit/GitHub**: Serves as the source code repository (GitHub in this case).
- **CodeBuild**: Compiles and builds the application.
- **CodeDeploy**: Automates deployment to EC2 instances.
- **EC2 Instances**: Host the deployed application.

🗂 Repository Structure

Your GitHub repository: aws-devops-zero-to-hero

```
bash
CopyEdit
aws-devops-zero-to-hero/
│
├── appspec.yml               # Deployment specification for
CodeDeploy
├── buildspec.yml             # Build instructions for CodeBuild
├── Dockerfile                # (If Dockerized, optional)
├── scripts/                  # Any startup or config scripts
├── sample-python-app/       # Application source code
└── README.md                 # Project description
```

🔗 Pipeline Flow

1. **Source Stage**
    a. Triggered when a new commit is pushed to GitHub.
    b. Uses GitHub as the source provider.
2. **Build Stage**
    a. Uses AWS CodeBuild to run `buildspec.yml` and package the app.
    b. Produces an artifact for CodeDeploy.
3. **Deploy Stage**
    a. Uses AWS CodeDeploy.
    b. Deploys the built artifact onto a target EC2 instance.

🐛 Current Issue

The pipeline is successfully fetching code from **GitHub** and building it via **CodeBuild**, but **CodeDeploy is fetching from S3** instead of directly from GitHub.

☑ Fix

You **cannot directly deploy from GitHub using CodeDeploy**. AWS **requires an artifact (zip or tar)** which must be:

- Stored in **S3**, or
- Generated from the **CodeBuild stage**.

So this is working **as expected**:

- GitHub → CodeBuild → Output ZIP artifact → CodeDeploy (reads from S3 location of this artifact).

Abhishek's video likely uses the **same mechanism**, where CodeBuild outputs the artifact to S3 and CodeDeploy reads from there. It only appears as if it comes "from GitHub."

☑ Prerequisites

- IAM roles:
  - `CodePipelineRole`
  - `CodeBuildRole`
  - `CodeDeployRole`
- EC2 Instance with:
  - CodeDeploy Agent installed
  - IAM Role attached with `AmazonEC2RoleforAWSCodeDeploy`
- appspec.yml in the root of the build artifact
- CodeDeploy Application and Deployment Group created

📷 Screenshots

Include the screenshots you shared in your documentation:

- CodePipeline structure (Source → Build → Deploy)
- GitHub repo and file structure
- EC2 instance with CodeDeploy agent setup

📝 Sample `buildspec.yml`

```
yaml
CopyEdit
version: 0.2

phases:
  install:
    runtime-versions:
      python: 3.9
    commands:
      - echo Installing dependencies...
      - pip install -r requirements.txt
```

```
artifacts:
  files:
    - appspec.yml
    - sample-python-app/**/*
    - scripts/**/*
```

📝 Sample `appspec.yml`

```
yaml
CopyEdit
version: 0.0
os: linux
files:
  - source: /
    destination: /home/ec2-user/sample-python-app
hooks:
  AfterInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: ec2-user
  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: ec2-user
```

📌 Deployment Notes

- Always make sure CodeDeploy agent is running on EC2 (`sudo service codedeploy-agent status`)
- If pipeline fails at the Deploy stage, check:
  - `appspec.yml` syntax
  - IAM role permissions
  - EC2 instance health and agent status

- CodeBuild artifacts uploaded correctly

🌐 GitHub Repository

➡️ GitHub – aws-devops-zero-to-hero

☑️ Future Enhancements

- Add automated tests in CodeBuild stage.
- Add SNS notifications on pipeline failure.
- Add Docker & ECS deployment options.
- Integrate with CloudWatch logs and alarms.