AWS CodePipeline with CodeDeploy on EC2 – Setup & IAM Role Guide

This document explains the complete setup of AWS CodePipeline with CodeDeploy on an EC2 instance. It includes the IAM role requirements, EC2 permissions, and installation of the CodeDeploy and SSM agents.

---

1. IAM Roles Required

a) CodePipeline Service Role

- **Policy Attachments:**
- `AWSCodePipeline_FullAccess`
- `AmazonS3FullAccess` (if artifacts stored in S3)
- `CloudWatchLogsFullAccess` (for pipeline logging)
- `AmazonEC2ReadOnlyAccess` (to describe EC2 instances)
- `AmazonSSMFullAccess` (to send commands to instances)

b) CodeDeploy Service Role

- **Trust Relationship:** Service → `codedeploy.amazonaws.com`
- **Policy Attachments:**
- `AWSCodeDeployRole`
- `AmazonEC2ReadOnlyAccess`
- `AmazonSSMFullAccess`

c) EC2 Instance Role

- **Trust Relationship:** Service → `ec2.amazonaws.com`
- **Policy Attachments:**
- `AmazonSSMManagedInstanceCore` (mandatory for Systems Manager)
- `AWSCodeDeployFullAccess`
- (Optional) `AmazonS3ReadOnlyAccess` if artifacts pulled directly from S3

---

2. EC2 Instance Preparation

a) Install CodeDeploy Agent (Ubuntu 24)

```
#!/bin/bash
sudo apt-get update -y
sudo apt-get install -y ruby-full wget
```

```
cd /tmp
wget https://aws-codedeploy-us-east-2.s3.us-east-2.amazonaws.com/latest/install
chmod +x ./install
sudo ./install auto

# Enable and start service
sudo systemctl enable codedeploy-agent
sudo systemctl start codedeploy-agent
sudo systemctl status codedeploy-agent --no-pager
```

👉Replace `us-east-2` with your AWS region.

---

b) Install & Enable SSM Agent (Ubuntu 24)

```
# Check if SSM Agent is installed
sudo systemctl status amazon-ssm-agent

# If not installed
sudo snap install amazon-ssm-agent --classic
sudo systemctl enable --now snap.amazon-ssm-agent.amazon-ssm-agent.service
```

Verify in AWS Console: - Go to **Systems Manager → Fleet Manager → Managed Instances** - Your EC2 instance should show as **Managed**

---

3. Common Errors & Fixes

Error 1: `logs:PutLogEvents AccessDenied`

- Fix: Attach `CloudWatchLogsFullAccess` to **CodePipeline Role**

Error 2: `ec2:DescribeInstances UnauthorizedOperation`

- Fix: Attach `AmazonEC2ReadOnlyAccess` to **CodePipeline Role**

Error 3: `ssm:SendCommand AccessDenied`

- Fix: Attach `AmazonSSMFullAccess` to **CodePipeline & CodeDeploy Role**

Error 4: `InvalidInstanceId – Instances not in a valid state`

- Fix: Ensure EC2 has `AmazonSSMManagedInstanceCore` role + SSM Agent installed & running

Error 5: `can't find user for apache`

- Fix: Ensure `appspec.yml` has correct runas user (e.g., `root` or `www-data`)

---

4. Verification Checklist

1. 👈EC2 instance has IAM role with `AmazonSSMManagedInstanceCore`
2. 👈CodeDeploy agent installed and running
3. 👈SSM agent installed and EC2 appears as **Managed**
4. 👈CodePipeline role has permissions for EC2, SSM, Logs, S3
5. 👈CodeDeploy role has permissions for EC2 + SSM
6. 👈AppSpec.yml correctly configured

---

5. Example appspec.yml (Ubuntu + Apache)

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html
permissions:
  - object: /var/www/html
    owner: www-data
    group: www-data
    mode: 755
hooks:
  AfterInstall:
    - location: scripts/restart_server.sh
      runas: root
```

👈 With this configuration, your EC2 instance will be ready for deployments via CodeDeploy and CodePipeline.