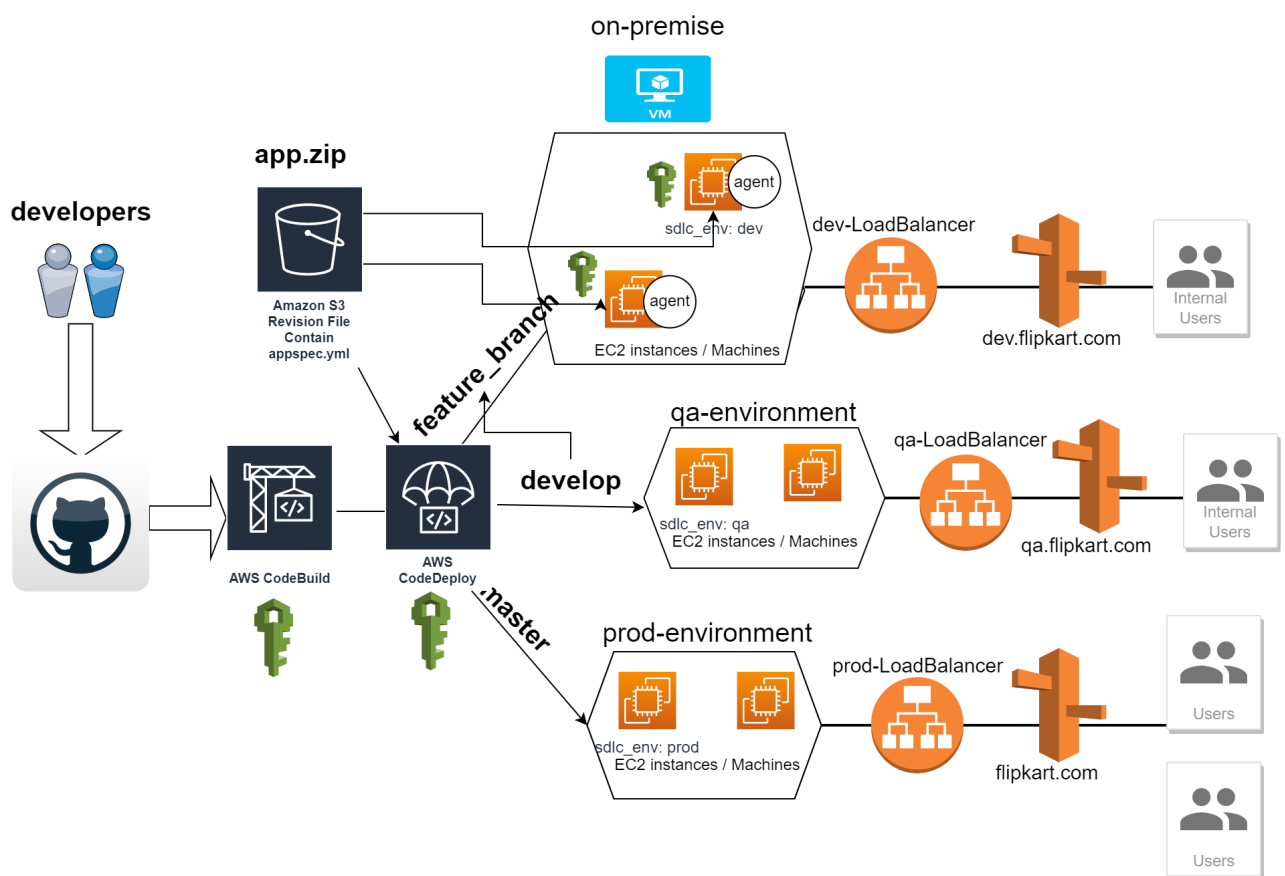


# AWS CICD

- AWS CICD
  - Configuration and Setup for Sample CodeDeploy Application
    - Creating a CodeDeploy Application
    - CodeDeploy with S3 having Source Code
      - Start Deployment using AWS Console
    - Viewing the CodeDeploy Agent Logs and Deployment logs on the EC2 instance.
    - Lifecycle Event hooks in appspec.yml file
    - Environment variables in CodeDeploy
    - Troubleshooting Scenarios

- Example Setup



## Configuration and Setup for Sample CodeDeploy Application

- Create an IAM user and configure aws cli on git bash using aws configure command.
- Launch an EC2 instance with Amazon Linux AMI.
- Attach IAM Role to this EC2 Instance which will have **AmazonEC2RoleforAWSCodeDeploy** Policy attached, as CodeDeploy Agent should have access to list/get Objects from S3.
- Add below shell commands to install CodeDeploy Agent on the EC2 using EC2 UserData.

```
#!/bin/bash
sudo yum update -y
```

```
sudo yum install -y ruby wget tree
wget https://aws-coddeploy-eu-west-1.s3.eu-west-1.amazonaws.com/latest/install
chmod +x ./install
sudo ./install auto
sudo service coddeploy-agent status
sudo service coddeploy-agent enable
```

- Tag the EC2 instance with Name and Environment Values as
  - **Name** : **Webserver**
  - **SDLC\_ENVIRONMENT** : **dev**

Alternatively, a Terraform Code/CF template to launch an EC2 instance with above commands as Userdata.

---

## Creating a CodeDeploy Application

- Go to **CodeDeploy Service** > **Create Application** > Select Compute Platform as "**EC2/On-Premises**"
- Create a Deployment Group with name **DevEC2DeploymentGroup**
- Create a IAM Role for CodeDeploy Service that will have access to EC2 service ( create an IAM Role with trusted entity **coddeploy.amazonaws.com**)
  - CodeDeploy service needs to read EC2 instance tags.
  - Attach **arn:aws:iam::aws:policy/service-role/AWSCodeDeployRole** policy to this role.
- The **Deployment type** selected by default is **in-place deployment**:
- Under **Environment configuration**, select **Amazon EC2 Instances**
  - Specify the right Tag Group **Key:Value** that is set on the EC2 instance.

--

- For Deployment Configuration, select the **CodeDeployDefault.OneAtATime**
- Deployment configuration
  - **CodeDeployDefault.OneAtATime** : Recommended Approach
    - Status is **Succeeded** if the application revision is deployed to all of the instances. The exception to this rule is that if deployment to the last instance fails, the overall deployment still succeeds.
    - Status is **Failed** if the application revision fails to be deployed to any but the last instance.
  - **CodeDeployDefault.AllAtOnce** :
    - Status is **Succeeded** if the application revision is deployed to one or more of the instances.
    - Status is **Failed** if the application revision is not deployed to any of the instances.
  - **CodeDeployDefault.HalfAtATime** :
    - Status is **Succeeded** if the application revision is deployed to at least half of the instances.

---

## CodeDeploy with S3 having Source Code

- Create a new or use existing S3 bucket and **enable versioning**, replace below **<S3BUCKETNAME>** with your bucket name.

- Upload your existing **app.zip** file into the bucket, this file is called as **Revision File** that contains application specification code i.e **appspec.yml** and required source code for application that needs to be deployed on EC2.
- The deployment can be started using the **AWS CodeDeploy Service UI** or by using **AWS CLI command**, replace appropriate content in the command below as per your account.

--

## Start Deployment using AWS Console

- Once Deployment Group is created, Navigate to **CodeDeploy Application** > Click on specific **Deployment Group** > Click on **Create Deployment**.
  - Under **Revision location**, specify the S3 URI Path of **app.zip** revision file from S3 Bucket Path that was previously uploaded.

--

- Once the Deployment will start, there will be a **Deployment ID** created like **d-GC3B88R0A**
- Click on **Deployment Id** to see the event details executed on specific EC2 instance.

The screenshot displays the AWS CodeDeploy console for a deployment named **d-JECP34OBF**. The interface includes a breadcrumb trail: **Developer Tools > CodeDeploy > Deployments > d-JECP34OBF**. At the top right, there are buttons for **Copy deployment** and **Retry deployment**.

**Deployment status:** A progress bar indicates "Installing application on your instances" with "1 of 1 instances updated" and a status of **Succeeded** at 100%.

**Deployment details:** A table provides key information:

Application <a href="#">codedeploy-demo-application-testing</a>	Deployment ID <b>d-JECP34OBF</b>	Status <b>Succeeded</b>
Deployment configuration <a href="#">CodeDeployDefault-OneAtATime</a>	Deployment group <a href="#">DevDeploymentGroup</a>	Initiated by User action
Deployment description deploy app.zip		

**Revision details:** Another table shows revision information:

Revision location <a href="#">s3://aws-devops-testbucket/app.zip</a>	Revision created 1 hour ago	Revision description Application revision registered by Deployment ID: d-JECP34OBF
---	--------------------------------	---

**Deployment lifecycle events:** A table lists events for the instance **i-04bc5d0b93baeed4e**.

Instance ID	Duration	Status	Most recent event	Events	Start time	End time
<a href="#">i-04bc5d0b93baeed4e</a>	8 seconds	<b>Succeeded</b>	ValidateService	<a href="#">View events</a>	Jan 11, 2022 8:41 AM (UTC+5:30)	Jan 11, 2022 8:41 AM (UTC+5:30)

- On **View Events** on a specific EC2 instance ARN is displayed : **arn:aws:ec2:ap-south-1:<ACCOUNT\_ID>:instance/i-0789556ecdf884653**
  - Details under this Page shows all the Event Hooks that are executed by the **CodeDeploy Agent** on the EC2.

Developer Tools > CodeDeploy > Deployments > d-JECP34QBF > arn:aws:ec2:us-east-1:082923708139:instance/i-04bc5dbb93baeed4e

**arn:aws:ec2:us-east-1:082923708139:instance/i-04bc5dbb93baeed4e**

Deployment details		
Application <a href="#">codedeploy-demo-application-testing</a>	Deployment ID d-JECP34QBF	Status ✔ Succeeded
Deployment configuration <a href="#">CodeDeployDefault.OneAtATime</a>	Deployment group <a href="#">DevDeploymentGroup</a>	Initiated by User action
Deployment description deploy app.zip		

Revision details		
Revision location <a href="#">s3://aws-devops-testbucket/app.zip</a>	Revision created 1 hour ago	Revision description Application revision registered by Deployment ID: d-JECP34QBF

Event	Duration	Status	Error code	Start time	End time
ApplicationStop	less than one second	✔ Succeeded	-	Jan 11, 2022 8:41 AM (UTC+5:30)	Jan 11, 2022 8:41 AM (UTC+5:30)
DownloadBundle	less than one second	✔ Succeeded	-	Jan 11, 2022 8:41 AM (UTC+5:30)	Jan 11, 2022 8:41 AM (UTC+5:30)
Beforeinstall	3 seconds	✔ Succeeded	-	Jan 11, 2022 8:41 AM (UTC+5:30)	Jan 11, 2022 8:41 AM (UTC+5:30)
Install	less than one second	✔ Succeeded	-	Jan 11, 2022 8:41 AM (UTC+5:30)	Jan 11, 2022 8:41 AM (UTC+5:30)
Afterinstall	less than one second	✔ Succeeded	-	Jan 11, 2022 8:41 AM (UTC+5:30)	Jan 11, 2022 8:41 AM (UTC+5:30)
ApplicationStart	less than one second	✔ Succeeded	-	Jan 11, 2022 8:41 AM (UTC+5:30)	Jan 11, 2022 8:41 AM (UTC+5:30)
ValidateService	less than one second	✔ Succeeded	-	Jan 11, 2022 8:41 AM (UTC+5:30)	Jan 11, 2022 8:41 AM (UTC+5:30)

- Access the Application using the Public IP of the EC2 instance in browser.
- Login into the EC2, validate the httpd webserver status:
  - `sudo service httpd status`
- Check for files downloaded under the `/var/www/html/`

## Viewing the CodeDeploy Agent Logs and Deployment logs on the EC2 instance.

- The folder where related **revisions**, **deployment history**, and **deployment scripts** on the instance are stored.

```
tree /opt/codedeploy-agent
tree /opt/codedeploy-agent/deployment-root
```

- Output of above command is below directory structure.

```
/opt/codedeploy-agent/deployment-root
├── deployment-instructions
│   ├── fb00c799-7826-4e8a-834b-98e83c278506-cleanup
│   ├── fb00c799-7826-4e8a-834b-98e83c278506-install.json
│   ├── fb00c799-7826-4e8a-834b-98e83c278506_last_successful_install
│   └── fb00c799-7826-4e8a-834b-98e83c278506_most_recent_install
├── deployment-logs
│   └── codedeploy-agent-deployments.log
├── fb00c799-7826-4e8a-834b-98e83c278506
│   └── d-0TY70HROA
│       ├── bundle.tar
│       ├── deployment-archive
│       │   ├── appspec.yml
│       │   ├── buildspec.yml
│       │   ├── index.html
│       │   └── scripts
│       └── after_install.sh
```

```

├── install_dependencies.sh
├── start_server.sh
├── stop_server.sh
├── validate_service.sh
├── logs
│   └── scripts.log
└── ongoing-deployment

```

- **Deployment Group ID** : This is the deployment group directory's name having an ID like `fb00c799-7826-4e8a-834b-98e83c278506`.
  - Each deployment group directory contains one subdirectory for each attempted deployment in that deployment group.
- **Deployment ID** directory represent each deployment in a deployment group with directory's name is its ID `d-0TY70HROA`. Each directory contains:
  - **bundle.tar** : a compressed file with the contents of the deployment's revision. Use a zip decompression utility if you want to view the revision.
  - **deployment-archive** : contains the contents of the deployment's revision.
  - **logs** :contains a **scripts.log** file that lists the output of all scripts specified in the deployment's AppSpec file.
    - `cat /opt/codedeploy-agent/deployment-root/fb00c799-7826-4e8a-834b-98e83c278506/d-0TY70HROA/logs/scripts.log`

- The folder on the instance where log files related to CodeDeploy Agent operations are stored.








```
cat /var/log/aws/codedeploy-agent/codedeploy-agent.log
```

- To determine the location of the last successfully deployed application revision

```
cd /opt/codedeploy-agent/deployment-root/deployment-instructions
tree .
```

- STOP the CodeDeploy Agent:
  - `sudo service codedeploy-agent status`
  - `sudo service codedeploy-agent stop`
- Navigate to the above steps to start a new deployment.

- As the CodeDeploy Agent is not running, All Application Hook Events will be in **Pending** status

Revision details					
Revision location <a href="#">s3://aws-codedeploy-test-bucket/app.zip</a>		Revision created 31 minutes ago		Revision description Application revision registered by Deployment ID: d-R14OKEHIK	
Event	Duration	Status	Error code	Start time	End time
ApplicationStop	-	 Pending	-	-	-
DownloadBundle	-	 Pending	-	-	-
BeforeInstall	-	 Pending	-	-	-
Install	-	 Pending	-	-	-
AfterInstall	-	 Pending	-	-	-
ApplicationStart	-	 Pending	-	-	-
ValidateService	-	 Pending	-	-	-

- To ensure the Deployment is successful, start the CodeDeploy Agent.

- `sudo service codedeploy-agent start`

---

## Lifecycle Event hooks in appspec.yml file

- ApplicationStop** – This deployment lifecycle event occurs even before the application revision is downloaded.
- DownloadBundle**- During this deployment lifecycle event, the CodeDeploy agent copies the application revision files to a temporary location.
- BeforeInstall**– You can use this deployment lifecycle event for preinstall tasks, such as decrypting files and creating a backup of the current version.
- Install** – During this deployment lifecycle event, the CodeDeploy agent copies the revision files from the temporary location to the final destination folder.
- AfterInstall** – You can use this deployment lifecycle event for tasks such as configuring your application or changing file permissions.
- ApplicationStart** – You typically use this deployment lifecycle event to restart services that were stopped during ApplicationStop.
- ValidateService** – This is the last deployment lifecycle event. It is used to verify that the deployment was completed successfully.
- Structure of 'hooks' Section

```
hooks:
  deployment-lifecycle-event-name:
    - location: script-location
```

```
timeout: timeout-in-seconds
runas: user-name
```

- Below is the sample appspec.yml file

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html/
hooks:
  ApplicationStop:
    - location: scripts/stop_server.sh
      timeout: 300
      runas: root

  BeforeInstall:
    - location: scripts/install_dependencies.sh
      timeout: 300
      runas: root

  AfterInstall:
    - location: scripts/after_install.sh
      timeout: 300
      runas: root

  ApplicationStart:
    - location: scripts/start_server.sh
      timeout: 300
      runas: root

  ValidateService:
    - location: scripts/validate_service.sh
      timeout: 300
```

## Environment variables in CodeDeploy

```
APPLICATION_NAME
DEPLOYMENT_ID
DEPLOYMENT_GROUP_NAME
DEPLOYMENT_GROUP_ID
LIFECYCLE_EVENT
```

- One more EC2 instance can be added by following the same steps as above to have pre-requisite setup.
- Once Deployment is created again, the `DevDeploymentGroup` will find all instances with same tagging mechanism and deploy the revision file.

## Troubleshooting Scenarios

- InstanceAgent::Plugins::CodeDeployPlugin::CommandPoller: Missing credentials - please check if this instance was started with an IAM instance profile
- Make Sure instance is started with IAM role attached,if you attach IAM Role to EC2 after the instance is already running, the Instance needs to be restarted. If codedeploy agent is installed and later IAM Role is attached, make sure Codedeploy agent is restarted or EC2 is restarted.