

CODE DEPLOY AND CODE PIPELINE PROJECT

Balashundaram A

OBJECTIVE:

Complete CI/CD with AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, and AWS CodePipeline

Step 1: IAM Roles for EC2-S3-CodeDeploy access

a) S3 full access

Create policy

Filter policies

Q s3

Showing 9 results

	Policy name	Used as
<input type="checkbox"/>	AmazonDMSRedshiftS3Role	None
<input checked="" type="checkbox"/>	AmazonS3FullAccess	None
<input type="checkbox"/>	AmazonS3ObjectLambdaExecutionRolePolicy	None
<input type="checkbox"/>	AmazonS3OutpostsFullAccess	None
<input type="checkbox"/>	AmazonS3OutpostsReadOnlyAccess	None
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	None
<input type="checkbox"/>	IVSRecordToS3	None
<input type="checkbox"/>	QuickSightAccessForS3StorageManagementAnalyticsReadOnly	None

b) Code-Deploy role

Create role

1

2

3

4

Attached permissions policies

The type of role that you selected requires the following policy.

Filter policies

Q Search

Showing 1 result

Policy name	Used as	Description
AWSCodeDeployRole	None	Provides CodeDeploy service access to expan...

Step 2: Server Deployment

- 1) Creating an IAM user with full **Administrative access** and login.
- 2) Creating an Amazon Linux EC2 server 1 for the developer to create and change the codes.
- 3) Creating another Linux server 2 for Code Deploy operations.

```
root@ip-172-31-5-82:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
  _ | _ | _ )  
  _ | ( _ | _ /  
  _ | \ _ | _ |  
Amazon Linux 2 AMI  
  
https://aws.amazon.com/amazon-linux-2/  
11 package(s) needed for security, out of 35 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-5-82 ~]$ sudo su -  
[root@ip-172-31-5-82 ~]# aws configure  
AWS Access Key ID [None]: AKIA6KKT2J4XQ6KIKBDN  
AWS Secret Access Key [None]: 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  
Default region name [None]: ap-south-1  
Default output format [None]: json  
[root@ip-172-31-5-82 ~]# aws s3 ls  
2021-09-23 11:11:08 createee  
[root@ip-172-31-5-82 ~]#
```

```
root@ip-172-31-43-214:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
  _ | _ | _ )  
  _ | ( _ | _ /  
  _ | \ _ | _ |  
Amazon Linux 2 AMI  
  
https://aws.amazon.com/amazon-linux-2/  
11 package(s) needed for security, out of 35 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-43-214 ~]$ sudo su -  
[root@ip-172-31-43-214 ~]#
```

In the Server 2, the following commands are executed to install CodeDeploy

- `yum install ruby -y`
-> to install ruby
- `yum install wget -y`
-> to install the wget package
- `wget https://aws-codedeploy-us-east-1.s3.amazonaws.com/latest/install`
-> wget command is used to download the files directly from the web.
- `chmod +x install`
-> changing permissions
- `./install auto`
- `service codedeploy-agent status`

```
root@ip-172-31-43-214:~  
[root@ip-172-31-43-214 ~]# service codedeploy-agent status  
The AWS CodeDeploy agent is running as PID 4326  
[root@ip-172-31-43-214 ~]#
```

Step 3: Creating the files required for the webpage from Developer machine

1) Index file - vi index.html

2) Yaml file - vi abc.yml

```
root@ip-172-31-5-82:~/deploy_dir/sampleapp
version: 0.0
os: linux
files:
- source: /index.html
  destination: /var/www/html/
hooks:
  BeforeInstall:
  - location: scripts/httpd_install.sh
    timeout: 300
    runas: root
  - location: scripts/httpd_start.sh
    timeout: 300
    runas: root
  ApplicationStop:
  - location: scripts/httpd_stop.sh
    timeout: 300
    runas: root
~
~
~
~
~
~
~
~
~
~
:wq!
```

This file helps to deploy the source code into webserver automatically.

3) Creating httpd install , start and stop files which is getting called in yaml file.

vi httpd_install.sh

```
#!/bin/bash
```

```
yum install -y httpd
```

vi httpd_start.sh

```
#!/bin/bash
```

```
systemctl start httpd
```

```
systemctl enable httpd
```

vi httpd_stop.sh

```
#!/bin/bash
```

```
systemctl stop httpd
```

```
systemctl disable httpd
```

Step 4: Creating CodeDeploy Application and Pushing the code to S3 bucket from Server 1

a) Creating a Bucket with **Public** policy and **Bucket Versioning** as Enabled.

b) Creating an application in the code deploy using CLI,

aws deploy create-application --application-name sampleapp

```
[root@ip-172-31-5-82 scripts]# aws deploy create-application --application-name sampleapp
{
  "applicationId": "be74e27b-61e6-4241-b1c6-a493d9cab9de"
}
[root@ip-172-31-5-82 scripts]#
```

c) Uploading the code to S3 by executing the command below,

aws deploy push --application-name sampleapp --s3-location s3://bucname/sampleapp.zip

d) Now in the s3 bucket sampleapp.zip is present.

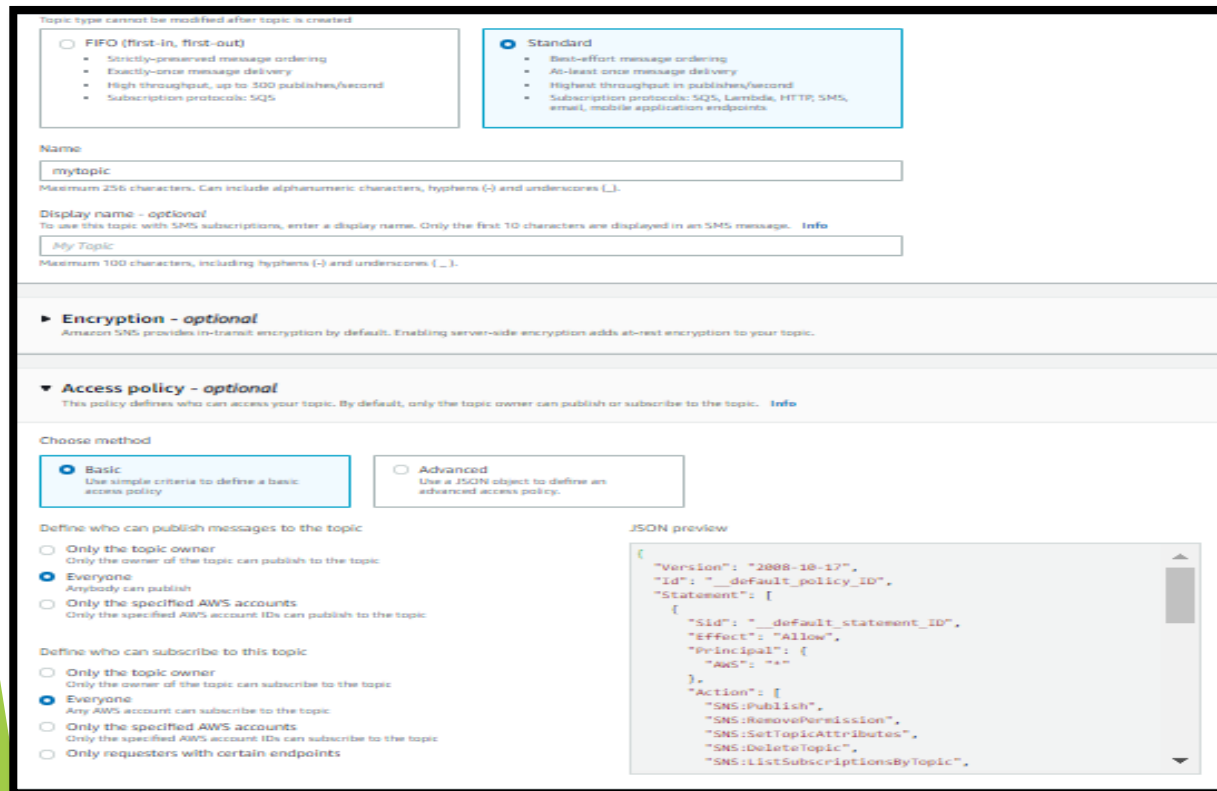
Step 5: Creating a **Deployment Group** to include Server 2 using AWS Management Console.

Step 6: Creating a **Deployment** which pushes code to the Server 2.

Step 7: Creating a **Pipeline** enabling automatic deployment the moment when the new version reaches the S3 repository.



And DONE! Now whenever the Developer changes the code and pushes it to S3 repository the Pipeline recognizes it and triggers services like Code Commit and Code Deploy resulting in changes in the webpage.



Enabling the SNS Notification service to alert us for the deployment success, fail, start, stop etc.