**AWS\_DevOps\_CI/CD**

Pulling code from CodeCommit, building the project on CodeBuild, deploying the application on CodeDeploy and making a CI/CD pipeline using Code Pipeline.

A diagram of a software application

Description automatically generated

Steps:

1. **Create repository in CodeCommit:**

Name: demo-app

[Create a user (as root user cannot create repos)

Username: ci\_cd\_user]

1. **Give access to CodeCommit for User:**

Security Credentials -> HTTPS Git credentials for AWS CodeCommit -> Generate credentials

1. **Clone in local repo:**

Open VSCode -> terminal

$git clone git clone <https://git-codecommit.us-east-1.amazonaws.com/v1/repos/demo-app>

* Provide git credentials.

Modify IAM permissions -> Attach AWSCodeCommitPowerUser permission

1. **Create an index.html file**

In terminal:

$ git add .

$ git commit -m "added sample file"

$ git push origin master

1. **CodeBuild**

Create build project(name : demo-app-build)

Source provider-> CodeCommit

Repository-> demo-app

Branch->master

A screenshot of a computer

Description automatically generated

Create servicerole

Create a buildspec.yml

Start build.

#Successful

Edit build-> Provide artifactory-> S3 location (save artifacts)

1. **CodeDeploy**

Create application

Application name: demo-app-application

Compute platform: Ec2

Create deployment groups-> (name)demo-app-depl-grp

Attach Service role: code-deploy-depl-grp with perm:

A screenshot of a computer

Description automatically generated

Create an ec2 instance: demo-app-ec2

[[[###ERROR - If the codedeploy is not assuming service role, then update the trust relation of role:

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "",

"Effect": "Allow",

"Principal": {

"Service": [

"codedeploy.amazonaws.com"

]

},

"Action": "sts:AssumeRole"

}

]

}

]]]

Disable load balancing and create deployment group.

1. **Setup CodeDeploy agent on ec2:**

Connect to ec2 instance

$vi install.sh

#!/bin/bash

# This installs the CodeDeploy agent and its prerequisites on Ubuntu 22.04.

sudo apt-get update

sudo apt-get install ruby-full ruby-webrick wget -y

cd /tmp

wget https://aws-codedeploy-us-east-1.s3.us-east-1.amazonaws.com/releases/codedeploy-agent\_1.3.2-1902\_all.deb

mkdir codedeploy-agent\_1.3.2-1902\_ubuntu22

dpkg-deb -R codedeploy-agent\_1.3.2-1902\_all.deb codedeploy-agent\_1.3.2-1902\_ubuntu22

sed 's/Depends:.\*/Depends:ruby3.0/' -i ./codedeploy-agent\_1.3.2-1902\_ubuntu22/DEBIAN/control

dpkg-deb -b codedeploy-agent\_1.3.2-1902\_ubuntu22/

sudo dpkg -i codedeploy-agent\_1.3.2-1902\_ubuntu22.deb

systemctl list-units --type=service | grep codedeploy

sudo service codedeploy-agent status

$bash install.sh

Create appspec.yml in vscode

Create scripts: install\_nginx.sh and start\_nginx.sh

Push it to repo

Build the code-> to push the artifacts into S3

Developer Tools 🡪CodeDeploy🡪Applications🡪demo-app-application🡪

Create deployment

Now Ec2 don’t have permission to talk to S3 and CodeDeploy

Create another role: ec2-code-deploy

A screenshot of a computer

Description automatically generated

Attach IAM role to the instance

Go to session

$sudo service codedeploy-agent restart

$sudo service codedeploy-agent status

A screenshot of a computer

Description automatically generated

**CodePipeline**

Creata pipeline: demo-app-pipeline

Create new service role

Add suggested items.

Run pipeline.

A screenshot of a computer

Description automatically generated

Modify index.html and verify ip address changes.

A screenshot of a computer

Description automatically generated

##Successful

References:

[1]<https://www.youtube.com/watch?v=p5i3cMCQ760&list=RDCMUClboKaFwE4JHnga2hVmOasQ&index=2>

[2] <https://www.youtube.com/watch?v=IUF-pfbYGvg>

[3] <https://www.trainwithshubham.com/blog/setting-up-aws-codedeploy-agent-on-ubuntu-ec2>

Install cloudwatch agent on instance

$wget https://amazoncloudwatch-agent.s3.amazonaws.com/oracle\_linux/amd64/latest/amazon-cloudwatch-agent.rpm