<https://chatgpt.com/share/de55ea22-3f61-40e5-808a-f9d7bc053ae4>

pulse

and

kcast-api swaggers

PAI-POC

KCT

Always latest image is selected

ecr

branches:

ai ==> diretcly upload image (batch)

mdm branch ==> latest docker immage

mcm ==>zip

we uplaod zip file

application load balancer ==>used for chat application

environmental variable==>secret manager ==> secret key

two ec2==>

three pipeline for three branches

one for kcast

cityhall AWS ===>client aws

=============================

prerequisite: installed git and IAM authentication

generate https for code commit repository

go in IAM ==> Users==> check the permission is given or not AWS Cloud build admin access

permissions ==https cedential==>generate credentials==> downlaod and save on local machine

========================

create repo in code commit==>create repository ==> clone it on local machine ==> take empty folder ==> cd to new folder==> clone ==> provide the credentials ==> cd to the clone folder ==>git add -A

git status

git commit -m "uploading the code"

push the code in code commit repo==> git push

to verify ==> go to code commit repositories==> choose the repository

create an ec2 instance for code deploy(if exist then access through the CLI)

================

code deploy ==> applications ==> create application==> ec2/platform

==> create deployment group(serice role as defined) ==> environment configuration(Amazon ec2 instances)[key value : our instance]==> create deploymnet group

==================

code pipeline ==>pipeline==>create pipeline[name ,new service role,auto generated]

source provider==> code commit[select repo,branch select things as per requirement]

deploy==> aws code deploy==> select deployment group

step1: pipeline setting

step2: add source tag

step3: add build stage

step 4: deploy action provider

create pipeline [ see the executions]

to verify code deployed sucessfully==> go to ec2 instances[copy ipv address paste in browser to see successfull deployment

**STEPS**

**Prerequisites**

**1.check aws cli installed or not with aws --version**

1. Installed Git

2. IAM Authentication [code build admin access

3. pull the code into your local machine create a empty folder and then clone git in it and copy code in new repo where the git has been clone if code size is big then directly clone in the code file

**Generate HTTPS for CodeCommit Repository**

1. Go to IAM

2. Under Users, check if the necessary permissions are given, specifically AWS Cloud Build Admin Access.

3. Under Permissions, navigate to HTTPS Credentials.

4. Generate credentials and download them. Save these credentials on your local machine.

**Create Repository in CodeCommit**

1. Create a Repository: Go to CodeCommit, create a new repository.

2. Clone the Repository: On your local machine, open a terminal and navigate to an empty folder.

cd /path/to/your/folder

3. Clone the Repository: Use the credentials to clone the repository.

git clone https://git-codecommit.region.amazonaws.com/v1/repos/YourRepositoryName

Enter the generated credentials when prompted.

4. Navigate to the Cloned Folder:

cd YourRepositoryName

5. Add Your Code:

git add -A

git status

git commit -m "Uploading the code"

6. Push the Code to CodeCommit:

git push

7. Verify the Push:

- Go to CodeCommit repositories.

- Select your repository and verify the code is uploaded.

Create EC2 Instance for CodeDeploy

**1. \*\*Create or Access an EC2 Instance\*\*:**

- If an instance already exists, access it through the CLI.

- Otherwise, create a new EC2 instance.

**Setup CodeDeploy**

1. \*\*Create Application\*\*:

- Go to CodeDeploy.

- Under \*\*Applications\*\*, create a new application.

- Select \*\*EC2/On-premises\*\* as the compute platform.

**2. \*\*Create Deployment Group\*\*:**

- Use the specified service role.

- Configure the environment to use Amazon EC2 instances.

- Add key-value pairs to identify your instance.

- Create the deployment group.

**Setup CodePipeline**

1. \*\*Create Pipeline\*\*:

- Go to CodePipeline.

- Create a new pipeline.

- Provide a name and select a new service role (auto-generated).

2. \*\*Add Source Stage\*\*:

- Source Provider: CodeCommit.

- Select the repository and branch as per your requirement.

3. \*\*Add Build Stage\*\* (Optional):

- Configure this stage if you have build steps (using AWS CodeBuild or similar).

4. \*\*Add Deploy Stage\*\*:

- Deploy Provider: AWS CodeDeploy.

- Select the application name and deployment group created earlier.

5. \*\*Create Pipeline\*\*:

- Review and create the pipeline.

- The pipeline will automatically trigger and execute.

**### Verify Deployment**

1. \*\*Monitor Pipeline Executions\*\*:

- Go to the pipeline details and check the execution status.

2. \*\*Verify EC2 Deployment\*\*:

- Access your EC2 instance.

- Copy the public IP address.

- Paste the IP address into your browser to verify successful deployment.