**Lambda Function (15 mins)**

1. Create Lambda Function and Lambda Role
   1. FunctionName: WorkshopFunction
   2. Runtime : Python 3.11
   3. Create new default execution role option
   4. Update the code as under :

def lambda\_handler(event, context):

body='Hello from Lambda!. ARN: '+context.invoked\_function\_arn

return {

'statusCode': 200,

'body': body

}

1. Publish Lambda Version. Update the body string and create another version.
2. Create two aliases – dev and prod. Point both of them to Version 1 of Lambda
3. Create Function URL for both the aliases. Go to Alias->Configuration->Function URL-> Create. Set Auth Type to None.
4. Note down the URL for Dev and Prod Alias

**Code Deploy(15 mins)**

1. Create Code Deploy Service Role, select service as CodeDeploy and then select CodeDeploy for Lambda. Give the name : CodeDeployServiceRole . Review and Create the Role. Add below 2 policies :
   * arn:aws:iam::aws:policy/AmazonS3FullAccess
   * arn:aws:iam::aws:policy/service-role/AWSCodeDeployRoleForLambda
2. Create S3 bucket with name : workshop-bucket-<initials>
3. Create and Upload appspec.yaml as below at root folder in the above bucket:

Resources:

- myLambdaFunction:

Type: AWS::Lambda::Function

Properties:

Name: WorkshopFunction

Alias: dev

CurrentVersion: 1

TargetVersion: 2

1. In CodeDeploy, Create New Application with Name : WorkshopApplication. Select Compute platform as Lambda and click on Create Application
2. Create Deployment Groups
   1. Application name : dev
   2. select the Code Deploy Service Role we created in Step 1
   3. Deployment Configuration: CodeDeployDefault.LambdaAllAtOnce
3. Create Deployment
   1. Select Deployment Group as dev
   2. Revision Location as : s3://workshop-bucket-dm1/appspec.yaml
   3. Revision File type as : .yaml

**CodeCommit/CodeBuild:**

1. Download User->Security Credentials -> HTTPS Git credentials for AWS CodeCommit
2. Create CodeCommit Repository with name : workshop-ci-cd-repo
3. Upload the code to repo
4. Create new Code Build Project :
   1. Name : workshop-build
   2. Select source as CodeCommit, give repo and branch name
   3. Select Managed Image, Amazon Linux 2 OS, Runtime : Standard, Image : ending with 5.0, Image Version : Latest, Environment : linux EC2
   4. New Service Role
      1. Name : codebuild-workshop-build-service-role
   5. Update and add below 2 policies to codebuild-workshop-build-service-role
      * + arn:aws:iam::aws:policy/AmazonS3FullAccess
        + arn:aws:iam::aws:policy/AWSLambda\_FullAccess
5. Create and Test Build

**CodePipeline:**

Create New Pipeline with

* 1. name : workshop-pipeline
  2. role : New Service Role
  3. Source Provide : Code Commit
  4. Select Repo name and branch name
  5. Build Provider as Code Build
  6. Select Project Name
  7. Deploy Provider as Code Deploy
  8. Select Application Name and Deployment Group
  9. Review and Submit