ADL EXPS BY<3

Task #	Task Description
1	Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE, write and run a simple Python program in IDE.
2	Setup AWS Cloud9 IDE, Launch AWS Cloud9 IDE, write and run a simple HTML script in IDE.
3	Collaborate with other users from within Cloud9 IDE.
4	Collaborate with other users and change access level for each user from within Cloud9 IDE.
5	Build an Application using AWS CodePipeline, deploy Sample Application on EC2 instance.
6	Build an Application using AWS CodePipeline, deploy Sample Application on EC2 instance, make changes to application code and deploy.
7	Install Terraform on Windows machine. Build, apply, and destroy AWS EC2 using Terraform. 1. Terraform 2. Terraform -version 3. Teraform init 4. Teraform plan 5. Terraform apply 6. Terraform destroy
8	Test TypeScript code using SonarQube.
9	Test Java code using SonarQube.

10	Test Python code using SonarQube.
	
	sonar.projectKey=python
	sonar.projectName=python
	sonar.projectVersion=1.0
	sonar.sources=C:\sonar-scanner-5.0.1.3006-windows\conf

11	Create a Hello world Lambda function using Python.
12	Create a Hello world Lambda function using Java.
13	Create a Hello world Lambda function using Node.js.

14	Create AWS Lambda function to log "an object has been added" on adding the object to S3 bucket.
	
	import json
	import boto3
	s3=boto3.client('s3')
	def lambda_handler(event,context):
	    bucket="q14bucket"
	    dataToUpload = {}
	    dataToUpload['PID'] = '211121'
	    dataToUpload['DEPT'] = 'INFT'
	    dataToUpload['NAME'] = 'Brijraaj'

	`    dataToUpload['FILE'] = 'brij'`
	`    fileName = 'brij' + '.json'`
	`    uploadByteStream= bytes(json.dumps(dataToUpload).encode('UTF-8'))`
	`    s3.put_object(Bucket=bucket,Key=fileName,Body=uploadByteStream)`
	`print('an object has been added')`

15	Create AWS Lambda function to visualize invocations.
16	Create an AWS Lambda function to log "I got output".
17	Create EC2 instance with the following configurations: - OS: Ubuntu (free tier) - Instance type: t2.micro - Key pair: .ppk 1. Connect to the created instance 2. Display present working directory pwd
18	Create EC2 instance with the following configurations: - OS: Ubuntu (free tier) - Instance type: t2.micro - Key pair: .ppk 3. Connect to the created instance 4. Run a command to switch to superuser 1. sudo adduser <new-username> 2. sudo usermod -aG sudo <new-username> 3. sudo su - <new-username>
19	Create an empty bucket in N. Virginia: - Add an object in the bucket - Delete the object from the bucket - Delete the bucket
20	Create an IAM role with the following policies: - s3fullaccess - awsbasiclambdaexecutionrole
21	Create a user with the username "adlUser" and add the user to group "adlGroup".
22	Deploy an AWS Elastic Beanstalk environment. 1. Create S3 bucket with EC2 instance 2. Create IAM role 3. Deploy Elastic Beanstalk environment 4. Create AWS CodePipeline and fork GitHub repository and Deploy
23	Create EC2 instance with the following configurations:
	- OS: Ubuntu (free tier)
	- Instance type: t2.micro

	- Key pair: .ppk
	- Install Docker and check its version
24	Create EC2 instance with the following configurations:
	- OS: Ubuntu (free tier)
	- Instance type: t2.micro
	- Key pair: .ppk
	- Install Docker
	- Enable Docker and then check Docker status