all 10-https://docs.google.com/document/d/1fVmPPjJnBoMPdV3ZXO9rajsb6Sic8EXue9w4PuqkfEM/edit

---------------------------------------------------------------------------------------------------------------------------------------------------------

5th (save as listfiles.bat)

@echo off

REM Batch script to list files in a directory

REM Get the current directory

set "currentDir=%CD%"

REM List all files and directories

dir "%currentDir%"

---------------------------------------------------------------------------------------------------------------------------------------------------------

6th

Install AWS CLI on your machine.

Step2: Create an IAM User

In AWS management console, Go to IAM service, create an IAM user

Attach policies to the user

User is created.

Go to security credentials

Click on create access key

Download .csv file which contain access key and secret access key

Step 3: Configure AWS CLI

Enter AWS access key, secret access key , default region and default output format.

Step 4: Create an ECR Repository

You can create Repository using AWS management console or AWS cli.

Run the following command to create a new ECR repository using AWS cli. Replace repository-name with your desired repository name.

aws ecr create-repository --repository-name your-repository-name

Go to AWS management console, then go to Amazon ECR service

In the Amazon ECR console, select “Repositories” from the left navigation pane.

Click the “Create repository” button.

Enter a unique name for your repository in the “Repository name” field.

Click the “Create repository” button.

Repository is created

Step 5: Build Docker Image Locally

Using your machine, navigate to the directory containing your Dockerfile

Build your Docker image.

docker image is created

Step 6: Push Docker Image to ECR

dockerfile--

# Use an official Node runtime as a parent image

FROM node:14

# Set the working directory in the container

WORKDIR /usr/src/app

# Copy package.json and install dependencies

COPY package.json ./

RUN npm install

# Bundle app source inside the Docker image

COPY . .

# Make port 80 available to the world outside this container

EXPOSE 80

# Define environment variable

ENV NAME World

# Run app when the container launches

CMD ["node", "app.js"]

1. Go back to the AWS Management Console.

2. In the Amazon ECR console, select your repository from the list.

3. In the repository details, click the “View push commands” button.

4. Follow the provided instructions to authenticate Docker with your ECR registry and push your Docker image.

Retrieve an authentication token and authenticate your Docker client to your registry. Use the AWS CLI:

aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 211125656815.dkr.ecr.ap-south-1.amazonaws.com

2. Build your Docker image using the docker build command. You can skip this step if your image is already built.

3. After the build completes, tag your image so you can push the image to this repository:

docker tag demo-app:latest 211125656815.dkr.ecr.ap-south-1.amazonaws.com/demo-app:latest

4. Run the following command to push this image to your newly created AWS repository:

docker push 211125656815.dkr.ecr.ap-south-1.amazonaws.com/demo-app:latest

Step 7: Verify Image in ECR Console

Once the image is pushed, go back to the Amazon ECR console. Select your repository.

Check the “Images” tab to verify that your Docker image has been successfully pushed.

--------------------------------------------------------------------------------------------------------------------------------------------------