Lab 3A: Introduction to Docker using AWS CloudShell

Lab Title:

Getting Started with Docker in AWS CloudShell

Objective:

Introduce students to basic Docker concepts and commands using AWS CloudShell. This lab serves as a prerequisite to Docker image creation, ECR pushing, and ECS deployment.

Duration:

1.5 hours

Pre-requisites:

- AWS Free Tier account
- AWS CloudShell access (pre-installed with Docker and AWS CLI)

Part A: Introduction and Setup (10 mins)

1. Open CloudShell

- Go to: https://console.aws.amazon.com/cloudshell
- Choose a region (e.g., ap`-south-1`)

2. Verify Docker Installation

```
docker --version
docker info
```

Expected output should confirm Docker is installed and running.

Part B: Run and Explore a Docker Container (20 mins)

1. Run a Simple Container

```
docker run hello-world
```

- This fetches the `hello-world` image from Docker Hub and runs it
- It prints a welcome message if successful

2. Run an Interactive Ubuntu Container

```
docker run -it ubuntu /bin/bash
```

- Explore inside the container:

```
ls
uname -a
exit
```

3. List Running and Stopped Containers

```
docker ps          # Running containers
docker ps -a       # All containers
```

Part C: Docker Images and Lifecycle (30 mins)

1. Search and Pull an Image

```
docker search alpine
docker pull alpine
```

2. Run a Command Using Alpine Image

```
docker run alpine echo "Docker is awesome!"
```

3. Inspect Images and Clean Up

```
docker images              # List images
docker rmi alpine          # Remove image
docker system prune -f     # Clean all stopped containers/images
```

Part D: Create Your Own Dockerfile (30 mins)

1. Create a Simple Web App

```
mkdir ~/docker-lab && cd ~/docker-lab
echo '<h1>Hello from Docker!</h1>' > index.html
```

2. Create a Dockerfile

```
echo -e 'FROM nginx:alpine\nCOPY . /usr/share/nginx/html' > Dockerfile
```

3. Build the Docker Image

```
docker build -t my-web-app .
```

4. Run the Image

```
docker run -d -p 8080:80 my-web-app
```

5. Preview the App

In CloudShell:

- Click **Actions** > **Preview > Preview port 8080**
- You should see "Hello from Docker!"

Part E: Wrap-Up and Cleanup (10 mins)

1. Stop and Remove Containers

```
docker ps
# Use CONTAINER ID from above

docker stop <id>
docker rm <id>
```

2. Remove Images

```
docker rmi my-web-app
```