DAY 3 :DEVOPS

1. Creating an EC2 Instance in AWS Console

Steps:

1. Login to AWS Console: Go to: https://console.aws.amazon.com and sign in.
2. Search for EC2: In the AWS services search bar, type EC2 and click on EC2 Dashboard.
3. Launch Instance:
    - Click “Launch Instance”.
    - Give your instance a name (e.g., MyFirstEC2).
    - Choose Amazon Machine Image (AMI):
        - Select Amazon Linux 2 AMI or Ubuntu (based on your need).
    - Choose Instance Type:
        - Choose t2.micro (Free tier eligible).
4. Key Pair:
    - Create a new key pair if not already created. (See section below)
5. Network Settings:
    - Allow SSH (port 22) in the security group.
    - Optionally allow HTTP or HTTPS if you're planning to host a web app.
6. Storage:
    - Leave default or increase based on your needs.
7. Launch:
    - Click “Launch Instance”.
    - After a minute, your instance will be running!

2. Create Access Key and Secret Key (for programmatic access)

These are needed when you use AWS CLI or SDKs.

Steps:

1. Go to AWS Console → IAM
2. Click on Users → Click your username.
3. Go to the Security credentials tab.
4. Scroll to Access keys section.
5. Click Create access key.
6. Select the use case: Command Line Interface (CLI).
7. Click Next, and then click Create access key.
8. Download CSV or copy and save the Access key ID and Secret access key securely.

3. Create a Key Pair and Download It (.pem file)

Steps:

1. In the EC2 dashboard → Click Key Pairs under Network & Security.
2. Click Create key pair.
3. Enter key pair name (e.g., mykeypair).
4. Choose Key pair type as RSA.
5. Choose Private key format as .pem.
6. Click Create key pair.
7. A .pem file will automatically download — keep it safe.

4. Using Key Pair in MobaXterm (SSH Login)

Steps:

1. Install MobaXterm (if not already):
    - Download from https://mobaxterm.mobatek.net/
2. Convert PEM to PPK using PuTTYgen (only if needed):
    - Open PuTTYgen (comes with MobaXterm).
    - Click Load → select your .pem file.
    - Click Save private key → it will save as .ppk.
    - This is required if using PuTTY; MobaXterm can use .pem directly sometimes.
3. Open MobaXterm
    - Click Session → SSH.
    - Remote Host: Use public IPv4 address of EC2 instance.
    - Specify username:
        - ec2-user for Amazon Linux
        - ubuntu for Ubuntu
    - Click on Advanced SSH Settings tab:
        - Check Use private key.
        - Browse and select your .pem file.
    - Click OK to start the SSH session.

Installing Terraform on Windows

Step 1: Download Terraform

1. Visit the official Terraform website: https://developer.hashicorp.com/terraform/downloads
2. Scroll down to the **Windows** section.
3. Click on **64-bit** or **32-bit** zip file depending on your system.
4. After download, extract the ZIP file. It contains a single file: terraform.exe.

Step 2: Add Terraform to System PATH

1. Move terraform.exe to a folder (e.g., C:\terraform).
2. Press Win + S → search for Environment Variables → Open "Edit the system environment variables".
3. In the System Properties window, click on Environment Variables.
4. In System Variables, find the variable Path → Click Edit.
5. Click New and add the path to your terraform folder (e.g., C:\terraform).
6. Click OK on all windows.

 Step 3: Verify Installation

- Open **Command Prompt** (cmd) terraform -v
- You should see the installed Terraform version.

Command Description

terraform init -- Initializes the current directory with Terraform config. Downloads provider plugins.

terraform validate --Validates the .tf code files for syntax errors and best practices.

terraform plan --Shows a preview of what Terraform will do without applying any changes.

terraform apply --Applies the changes required to reach the desired state (provision infra).

terraform destroy --Deletes all the resources that were created using the .tf file.

Example Terraform .tf Snippet (for AWS EC2):

```
provider "aws" {
  region     = "us-east-1"
  access_key = "YOUR_ACCESS_KEY"
  secret_key = "YOUR_SECRET_KEY"
}

resource "aws_instance" "myec2" {
  ami           = "ami-0c02fb55956c7d316"  # Example Amazon Linux 2 AMI
  instance_type = "t2.micro"
```

```
  key_name      = "your-keypair-name"


  tags = {

    Name = "MyTerraformEC2"

  }

}
```

2: Installing Docker Desktop on Windows

 Step 1: Download Docker

1. Goto
    https://www.docker.com/products/docker-desktop/
2. Click Download for Windows (Docker Desktop Installer.exe).
3. After the file is downloaded, run the installer.

 Step 2: Install Docker

1. Follow the installation wizard.
2. When prompted:
    - Enable WSL 2 option (recommended).
    - It may also ask to install WSL 2 backend if it's not already installed (link will be provided).

 Step 3: Start Docker

1. Once installed, launch Docker Desktop from the Start menu.
2. Wait for Docker to start — it will show "Docker is running".

 Step 4: Verify Installation

- Open Command Prompt or PowerShell and type:

  ```
  docker --version
  docker run hello-world
  ```

- If installed correctly, Docker will pull a test image and show a "Hello from Docker!" message.

- **Command Description**

docker --version Check Docker version installed.

docker pull <image>-- Download an image from Docker Hub (e.g., docker pull ubuntu).

docker images --List all downloaded images.

docker run <image> --Run a container from an image (e.g., docker run ubuntu).

docker ps --List **running** containers.

docker ps -a --List **all** containers (including stopped ones).

docker stop <container_id> --Stop a running container.

docker rm <container_id>-- Remove a stopped container.

docker rmi <image_id> Remove an image from local machine.

docker exec -it <container_id> bash Execute a bash terminal inside the container.

docker build -t <name> . Build a Docker image from a Dockerfile.

docker login --Log in to Docker Hub to push/pull images.

docker push <image> --Push a local image to Docker Hub.

docker logout --Logout from Docker Hub.