DevOps Training Tasks

Notes:

- Create a new repository in your GitHub account.
- Push all files, scripts, images, etc to this newly created repository for each task, folder structure in repository should look like below:
    - Repo-devops-tasks
        - Task-1
            - ReadME.md
            - File1
            - File2
        - Task-2
            - ReadME.md
            - File1
            - File2
- You need to create a simple document for each task and add steps followed for each task and you should add the screenshot of output of the result.
- All the documents need to be created in Markdown format. Learn how to create document in markdown format.
- Important Note: Do not push your AWS Access key and Secret key to GitHub repository, also do not push sensitive data like credentials, keys, etc.

Task 1: Docker, Docker Hub

- Create a Dockerfile for a simple web application (e.g. a Node.js or Python app)
- Build the image using the Dockerfile and run the container
- Verify that the application is working as expected by accessing it in a web browser
- Push the image to a public or private repository (e.g. Docker Hub)

Task 2: AWS, Jenkins, Docker, CICD, NodeJS

- Set up Jenkins on AWS EC2 instance.
- Create a security group for Jenkins.
- Connect to Jenkins instance using EC2 security groups
- Install Jenkins on the EC2 instance
- Creating a Dockerfile (you can use Dockerfile created in Task1)
- Install and manage some Jenkins plugins
- Create a job for automating CI/CD deployment
- Integrate Jenkins and GitHub
- Refer this blog for this task: https://devopscommunity.hashnode.dev/deploy-a-nodejs-app-using-jenkins-on-aws-ec2-instances

Task 3: Minikube, Kubernetes

- Understand Minikube
- Install Minikube either on your local laptop or on AWS EC2 instance
- Run Nginx named POD on minikube with container image “nginx:latest: and expose Nginx application on port 80

Task 4: Terraform, AWS, Sample App

Important Note: Do not push your AWS Access key and Secret key to GitHub repository

- Install Terraform on your local laptop
- Create Access keys in your AWS account
- Install AWS CLI on your local laptop
- Configure AWS on your local laptop using Access key and Secret key
- Create terraform script to create below infrastructure on AWS
    - VPC
    - Internet Gateway
    - Public Subnet
    - Private Subnet
    - Route Table
    - Route Table Association with Subnets
    - EC2 Instance
    - Security Group
    - Elastic IP

- You need to pass User-data script to EC2 Instance resource group section in terraform to install and run sample application using terraform
- Refer this blog for this task, also read my comment below this blog. https://medium.com/@sayalishewale12/terraform-hands-on-project-build-your-own-aws-infrastructure-with-ease-using-infrastructure-as-9f17640518d7