**Assignment of DevOps Engineer**

**1. Version Control:**

Git Repository: Create a public repository on GitHub to host the project. Let's name it simple-web-application.

**2. CI/CD Pipeline:**

CI/CD Service: We'll use Jenkins to set up our CI/CD pipeline.

Stages:

Build: Jenkins will clone the repository and copy the HTML file to the artifact directory.

Test: Since it's a simple HTML page, we might not have traditional unit tests. However, we can add a script to check the HTML file for validity.

Deploy: After successful build and test stages, Jenkins will deploy the HTML file to a staging environment.

**3. Infrastructure as Code (IaC):**

We Use terraform to create an EC2 instance on amazon web service.

**4. Configuration Management:**

We'll use Ansible to automate server configuration and setup, though in this case, it will be minimal since we're deploying a static HTML file.

**Now Steps: -**

1. On GitHub, create a public repository named simple-web-application.
2. Clone that public repo into your local system.
3. Using VS Code, create a Terraform configuration file named main.tf where you define the infrastructure setup, such as an EC2 instance.
4. Run the following commands in the terminal:

* terraform init (to download all the dependencies)
* terraform plan (to check for errors in the code)
* terraform apply (to create an EC2 instance based on the configuration)

1. After creating the EC2 instance, access it from your local host through the command line and install Git, Java, and Jenkins on the instance.
2. Configure the security group of the instance to allow traffic on port 8080, ensuring that Jenkins can run on the EC2 instance.
3. Push your web application code to GitHub and add webhooks in your repository to trigger Jenkins builds upon code changes.
4. Obtain the public IP of the EC2 instance and access Jenkins in a web browser using <public\_ip>:8080.
5. Log in to Jenkins and install the necessary plugins, such as "Publish Over SSH," and configure it to connect to your EC2 instance.
6. Create a pipeline project in Jenkins, connecting it to your GitHub repository and using the Jenkinsfile defined in your repository for pipeline configuration.
7. In your GitHub repository, create a Jenkinsfile and an Ansible file named deploy.yml.
8. After completing the setup, trigger a build in Jenkins to start the CI/CD process.

Note: Ensure all configuration files and scripts, including Terraform configuration (main.tf), Jenkinsfile, and Ansible playbook (deploy.yml), are uploaded to your GitHub repository.

The steps I provided outline a basic DevOps assignment, focusing on setting up a CI/CD pipeline for a simple web application. It covers key aspects of DevOps practices, including version control, infrastructure as code, configuration management, and automation.

Here's a breakdown of the DevOps principles and practices demonstrated in your assignment:

1. Version Control: Utilizing Git and GitHub for version control ensures that changes to the codebase are tracked, facilitating collaboration and version management among team members.
2. Infrastructure as Code (IaC): Using Terraform to provision and manage infrastructure allows for consistent and reproducible deployments, promoting scalability and automation in infrastructure management.
3. Configuration Management: Employing Ansible for server configuration automates the setup and maintenance of server environments, ensuring consistency and reducing manual intervention.
4. Continuous Integration/Continuous Deployment (CI/CD): Implementing a CI/CD pipeline with Jenkins automates the build, test, and deployment processes, enabling rapid delivery of changes to the staging environment, enhancing efficiency, and reducing the likelihood of errors.
5. Automation: Throughout the assignment, automation plays a crucial role in streamlining processes, reducing manual effort, and ensuring consistency in deployments and configurations.

Overall, your assignment reflects a practical application of DevOps principles and tools to enhance the development and deployment workflows of a web application.

Created by: -

Ujjwal Garg