CICD Pipeline with Jenkins-Provision and Deploy to EKS using Terraform

This documentation provides a step-by-step guide to set up a CICD pipeline using Jenkins, Terraform, and AWS. The pipeline will provision an AWS VPC, Jenkins EC2 server, and a Kubernetes cluster using Terraform as Infrastructure as Code (IaC) tool. Additionally, it will deploy a test application to the Kubernetes cluster.

Prerequisites

Before setting up the CICD pipeline, ensure you have the following prerequisites:

1. AWS account credentials with sufficient permissions to create resources.
2. Jenkins installed and configured.
3. Terraform installed on the Jenkins server.

Step 1: Clone the Repository

Clone the project repository from the provided source code repository URL:

git clone <repository-url>

Step 2: Configure AWS Credentials

Configure your AWS credentials on the Jenkins server. This can be done by setting environment variables or using a Jenkins plugin like the AWS Credentials Plugin.

Step 3: Set Up Jenkins Job

1. Open the Jenkins web interface and create a new Jenkins job.
2. Configure the job to fetch the source code from the cloned repository.
3. Add a build step to execute shell commands.

Step 4: Define Infrastructure as Code (IaC)

1. Navigate to the **terraform** directory in the cloned repository.
2. Explore the Terraform configuration files (**main.tf, variables.tf etc**) to define the AWS VPC, Jenkins EC2 server, and Kubernetes cluster resources as desired.
3. Define the necessary Terraform variables and input variables for customization.

Step 5: Provision Infrastructure with Terraform

In the Jenkins job's build step, use **Jenkinsfile** to execute the following Terraform commands to provision the infrastructure and deploy application to EKS:

cd terraform

terraform init

terraform validate

terraform plan

terraform apply

Step 6: Deploy Application to Kubernetes

1. Install the Kubernetes CLI (**kubectl**) on the Jenkins server. This is automatically installed if you provisioned your Jenkins server using the terraform stack. Use **Jenkinsfile2** in the pipeline if your resources are already provisioned.
2. Configure **kubectl** to connect to the Kubernetes cluster provisioned in the previous step.

3. Build your application Docker image and push it to a container registry using **Jenkinsfile3.**
4. Use **kubectl** to deploy the application to the Kubernetes cluster.

Step 7: Automate the Pipeline

To automate the CICD pipeline, configure the Jenkins job to be triggered automatically on source code changes or on a predefined schedule.

Additional Considerations

- **Security**: Ensure that proper security measures are implemented, such as securing AWS credentials, configuring appropriate security groups, and restricting access to sensitive resources.
- **Monitoring**: Set up monitoring and logging for your infrastructure and application to track performance and identify issues.
- **Scaling**: Consider scaling options for your infrastructure and application as per your requirements.

Conclusion

By following this documentation, you can set up a CICD pipeline using Jenkins, Terraform, and AWS. The pipeline will provision the required infrastructure and deploy your test application to a Kubernetes cluster. This will enable you to automate the process of provisioning resources and deploying your application, ensuring efficient and reliable software delivery.

Feel free to contribute to the project, make improvements, and provide feedback to enhance the CICD pipeline.