**Project: Automated Infrastructure Provisioning and Application Installation**

**Project Completion Report**

In this project, we successfully achieved the goal of automating infrastructure provisioning using Terraform and installing Java, Jenkins, and Python on the EC2 instance manually by logging into the instance with an SSH key pair. Below, we provide a detailed report of the tasks performed and the outcomes achieved.

**Task 1: Launching an EC2 Instance Using Terraform**

**Description:** We used Terraform to provision an Amazon Elastic Compute Cloud (EC2) instance on AWS.

**Steps Completed:**

1. Configure the AWS CLI.
2. Configured the AWS provider in Terraform, specifying the region.
3. Defined the EC2 instance with the following details:
   * Selected an appropriate Amazon Machine Image (AMI).
   * Chose an instance type (t2.micro).
   * Specified security groups for SSH and HTTP access.
   * Used an existing SSH key pair for secure access.
4. Executed Terraform to create the EC2 instance.

**Outcome:**

* A new EC2 instance was successfully provisioned on AWS.

**Task 2: Connecting to the EC2 Instance and Manual Installation**

**Description:** We established a secure SSH connection to the provisioned EC2 instance using an SSH key pair and manually installed Java, Jenkins, and Python on the instance.

**Steps Completed:**

1. Retrieved the public IP address of the EC2 instance from the Terraform output.
2. Used SSH to connect to the instance with the SSH key pair.
3. Manually executed the following commands on the EC2 instance to install the required applications:
   * Installed OpenJDK (Java 8).
   * Installed Python.
   * Installed Jenkins.

**Outcome:**

* We gained secure SSH access to the EC2 instance.
* Java, Python, and Jenkins were successfully installed on the EC2 instance manually.

**Conclusion**

This project showcased the initial provisioning of infrastructure using Terraform and the subsequent manual installation of Java, Jenkins, and Python on the EC2 instance by logging in with an SSH key pair. While Ansible is a powerful automation tool, this project demonstrated the flexibility of performing tasks manually when needed.

The successful setup of the centralized Jenkins server on the EC2 instance will significantly contribute to your organization's DevOps goals, enabling efficient software development processes, testing, and development environments, and enhancing overall system reliability. This project marks an important step in your DevOps journey, combining both automation and manual intervention where appropriate.

Top of Form