Phase-1 Submission Template

Student Name: VISHAL S

Register Number: 511823205043

Institution: PODHIGAI COLLEGE ENGINEERING &TECH

Department: B.TECH-INFORMATION TECHNOLOGY

Date of Submission: 12-05-2025

1.Problem Statement

Manual cloud infrastructure provisioning is slow, error-prone, and difficult to scale. This project implements Infrastructure as Code using Terraform for automated resource provisioning and Ansible for configuration management, enabling faster, consistent, and scalable cloud deployments.

2.Objectives of the Project

1. **Automate Cloud Infrastructure Provisioning using Terraform to define and deploy infrastructure components (e.g., networks, virtual machines, storage) in a reproducible and scalable way.**
2. **Configure and Manage Systems using Ansible for post-provisioning tasks such as software installation, service configuration, and system updates.**
3. **Ensure Infrastructure Consistency across environments (development, staging, production) through version-controlled Infrastructure as Code.**
4. **Enhance Deployment Efficiency by minimizing manual intervention and enabling rapid infrastructure changes and rollbacks.**
5. **Improve Collaboration through modular, reusable code and integration with CI/CD pipelines for continuous delivery of infrastructure updates.**

3.Scope of the Project

The project focuses on automating cloud infrastructure provisioning using Terraform and managing system configurations with Ansible. It includes creating scalable, version-controlled infrastructure and ensuring consistent environment setups. The scope covers provisioning, configuration, and integration with CI/CD pipelines for deployment automation.

4. Resources & Tools Used

1. Terraform – For automating cloud infrastructure provisioning (e.g., EC2, VPC, S3).

2. Ansible – For configuration management and software installation on provisioned servers.

3. Cloud Provider – AWS, Azure, or GCP as the infrastructure deployment platform.

4. Git & CI/CD Tools – Git for version control, and Jenkins/GitHub Actions for deployment automation.

5. Secrets Management – Ansible Vault or cloud-native tools (e.g., AWS Secrets Manager) for secure credentials handling.

5.High-Level Methodology

1. Requirement Analysis: Identify cloud resources and configurations needed (e.g., servers, networks, storage).

2. Design Infrastructure Architecture: Define a scalable and modular architecture tailored to the selected cloud provider.

3. Terraform Implementation: Write Terraform code to provision infrastructure components (e.g., EC2 instances, VPCs).

4. Ansible Configuration: Develop Ansible playbooks to automate software installation and configuration on provisioned resources.

5. Version Control and Collaboration: Use Git for managing code and enabling team collaboration.

6. Testing and Validation: Perform testing in isolated environments to validate infrastructure and configuration scripts.

7. Monitoring and Maintenance: Implement monitoring tools and regularly update infrastructure and configurations as needed.

6.Tools and Technologies

1.Programming Language	**-**	**HCL,YAML,Python,Bash/Shell scripting**
2.Notebook/IDE	**-**	**Terraform,Ansible,Jenkins**
3.Platform/IDEs	**-**	**AWS, VS code**
4. Deployments & Monitering	**-**	**Prometheus,ELK stack**
5.Optional Frameworks	**-**	**kustomize,ArgoCD,Helm,etc**

7.Team Members and Roles

1. Vishal S - Handles infrastructure automation with Terraform and Ansible.
2. Prasanth A - Sets up and manages cloud services .
3. Prakash S - Configures servers and installs software using Ansible.
4. Kalaiarasan K - Takes care of security, permissions, and protecting sensitive data.