Automating Infrastructure Provisioning with Terraform & Ansible

Part 1: Office Roleplay Dialogue

Scenario: A DevOps Engineer, Hiroshi, is discussing infrastructure automation with his colleague, Arjun.

Arjun: Hiroshi, we need a faster way to handle **infrastructure provisioning**. Manual setup is slowing us down.

Hiroshi: I agree. We should use **Terraform** for defining infrastructure as code. It'll make deployments consistent.

Arjun: That sounds good, but what about configuration? Would **Ansible** be a better fit for that?

Hiroshi: Exactly. **Terraform** sets up the infrastructure, while **Ansible** handles **configuration management** like installing packages.

Arjun: That makes sense. So, with this approach, we'd have a fully automated, **declarative infrastructure**, right?

Hiroshi: Right. We define what we need, and the tools ensure it matches that state. No more manual errors.

Arjun: Great! Can you help me set up the Terraform scripts? I want to make sure I follow best practices.

Hiroshi: Sure! We'll start by defining the resources in Terraform and then use Ansible for post-deployment configurations.

Arjun: Sounds perfect. Once we automate this, managing infrastructure will be so much easier.

Hiroshi: Exactly! Let’s get started with a test deployment.

Part 2: Comprehension Questions

1. What problem are Hiroshi and Arjun trying to solve?

(A) Slow manual infrastructure setup

(B) Network security issues

(C) Server overheating

(D) Low disk space

2. Which tool does Hiroshi suggest for defining infrastructure as code?

(A) Kubernetes

(B) Terraform

(C) Docker

(D) Jenkins

3. What does Ansible handle in this setup?

(A) Infrastructure provisioning

(B) Load balancing

(C) Configuration management

(D) Database queries

4. What is the main benefit of using declarative infrastructure?

(A) It ensures infrastructure matches a defined state

(B) It speeds up hardware upgrades

(C) It removes the need for coding

(D) It prevents all server downtime

Part 3: Key Vocabulary Definitions in Japanese

1. **Infrastructure provisioning (インフラプロビジョニング)** – サーバーやネットワークの設定を自動化してインフラを準備するプロセス。

2. **Terraform (テラフォーム)** – インフラをコードとして管理し、自動構築するためのツール。

3. **Ansible (アンシブル)** – 構成管理やアプリケーションデプロイメントを自動化するツール。

4. **Configuration management (構成管理)** – システムの設定やソフトウェアのバージョンを自動的に管理するプロセス。

5. **Declarative infrastructure (宣言型インフラストラクチャ)** – 望ましいシステムの状態を定義し、自動的に維持する方法。

Part 4: Questions & Correct Answers

1. **What problem are Hiroshi and Arjun trying to solve?**
   ✅ (A) Slow manual infrastructure setup

2. **Which tool does Hiroshi suggest for defining infrastructure as code?**
   ✅ (B) Terraform

3. **What does Ansible handle in this setup?**

   ✅ (C) Configuration management

4. **What is the main benefit of using declarative infrastructure?**

   ✅ (A) It ensures infrastructure matches a defined state