**1. Task Allocation with Kubernetes and Infrastructure as Code (IaC)**

Your DevOps team manages a Kubernetes cluster using Infrastructure as Code (IaC) with

Terraform. You need to allocate a task to scale the application based on increased traffic.

**Question:**

How would you allocate the task of scaling the application in the Kubernetes cluster using

Terraform and ensure it's implemented efficiently?

**Tasks given**

Task 1: Explain how to identify the need for scaling based on traffic metrics or other indicators.

Task 2: Describe the process of creating or updating Terraform code to adjust the desired

replica count of the application.

Task 3: Provide guidelines for testing the scaling changes and deploying them to the

Kubernetes cluster while minimizing downtime.

**Task 1: Identifying the Need for Scaling**

* Monitor how your application is doing by looking at things like how much CPU it's using or how many people are using it.
* Decide when you should make your application bigger (scale up) or smaller (scale down) based on what you see.

**Task 2: Adjusting Desired Replica Count with Terraform**

* Use Terraform to tell your system how many copies (replicas) of your application should be running.
* If you need more copies, tell Terraform to make more. If you need fewer, tell it to reduce the number.

**Task 3: Testing and Deploying Scaling Changes**

* Before making changes in the real system, practice in a safe copy (staging) of your application.
* When you're sure your changes won't break things, apply them to the real application slowly and carefully to avoid problems.
* Keep an eye on your system to make sure it stays healthy. If something goes wrong, be ready to go back to the way things were before.