Table of Contents

[Recommendation: 2](#_Toc186214137)

[Expected Advanced Experiment Scenarios (On High-level): 2](#_Toc186214138)

[HDFC Infra Architecture Overview 2](#_Toc186214139)

[High-Level Multi-Cloud Architecture: 2](#_Toc186214140)

[Cloud Providers: 2](#_Toc186214141)

[Network Security Tools: 2](#_Toc186214142)

[Applications and Storage: 2](#_Toc186214143)

[Applications: 2](#_Toc186214144)

[Databases & Storage: 2](#_Toc186214145)

[Message Queues: 2](#_Toc186214146)

[Network Components: 2](#_Toc186214147)

[Multi-Cloud Network Connectivity: 2](#_Toc186214148)

[Kubernetes Cluster Details: 2](#_Toc186214149)

[**Approach**: (Architecture) 2](#_Toc186214150)

[Requirements: 2](#_Toc186214151)

[Network Components: 3](#_Toc186214152)

[Kubernetes Cluster Details: 3](#_Toc186214153)

[Monitoring 3](#_Toc186214154)

[API Gateway 3](#_Toc186214155)

[Client Expectations 3](#_Toc186214156)

[Preferred Scenarios: 3](#_Toc186214157)

[Workflow: 3](#_Toc186214158)

[Strategies to Ensure App Resilience 3](#_Toc186214159)

[Experiment Strategies: 3](#_Toc186214160)

[Balanced Approach for Security (Depends on feasibility): 4](#_Toc186214161)

[Action Items 4](#_Toc186214162)

[Plan Of Action: 4](#_Toc186214163)

[Project management: 4](#_Toc186214164)

[Documentation: 5](#_Toc186214165)

[Resource requirement: 5](#_Toc186214166)

[Timeline: 5](#_Toc186214167)

# Recommendation:

* To setup entirely a **New** **Multi-cloud** infra environment to simulate the desired experiments as per the ask from the HDFC client.

# Expected Advanced Experiment Scenarios (On High-level):

1. Pod kill in a slow manner.
2. Storage burst scenarios.
3. Network burst scenarios(network latency & partition).
4. Disruption of HPA (Horizontal Pod Autoscaler).
5. **Disruption of Node Auto-scaler.**
6. **Crippling nodes.**
7. Slow app kill: Gradual degradation in app performance (from optimal to null) over 5 minutes to 1 hour.
8. Multi-cloud resilience testing.
9. **Denial-of-service attack.**

# HDFC Infra Architecture Overview

## High-Level Multi-Cloud Architecture:

### Cloud Providers:

* + - GCP: Messaging systems.
    - AWS: Interacts with GCP services.
    - Azure: Facilitates data link interactions.
    - On-Prem: Servers hosted in HDFC Datacentre (Mumbai).

## Network Security Tools:

* + WAF (Web Application Firewall): Ensures network resiliency.
  + Palo Alto: Hosted in a cloud environment.

## 

## Applications and Storage:

### Applications:

* + - Net Banking.
    - Mobile Banking.

### Databases & Storage:

* + - AeroSpike DB, Postgres DB, and Cloud Buckets (Mobile Banking).
    - AWS-hosted MySQL DB.

### Message Queues:

* + - Pub/Sub, Kafka.

### Network Components:

* + - Load balancers and additional network components.

## Multi-Cloud Network Connectivity:

* + GCP ↔ AWS → On-Prem → Azure.
  + Direct communication exists between GCP and AWS.
  + AWS forwards requests to On-Prem servers.

## Kubernetes Cluster Details:

* + **Cluster Size:** 5 nodes.
  + **Workloads:** ~20 pods.

# **Approach**: (Architecture)

No of microservices: 10 (Gpay App)

No of microservices: 9 (Bank-of-anthos)

A diagram of a diagram

Description automatically generated

## Requirements:

To simulate similar infrastructure within our SIDGS infrastructure eco system. We need the following:

* **Multiple integrated cloud environments** needed to setup the infrastructure, therefore to run the advanced experiments.
  + **GCP**
  + **AWS**
* We need three applications (**eg**: bank-of-anthos) which should be created in-house.
* Two web applications
* One mobile application
* We need to have a middleware like Kafka to simulate streaming.
* We need **Databases** mentioned below
* AWS-hosted MySQL DB
* GCP-hosted PGSQL DB

## Network Components:

* Load balancers.

## Kubernetes Cluster Details:

* + **Cluster Size:** 5 nodes.
  + **Workloads:** ~20 pods.

## Monitoring

* + ELK Stack

## API Gateway

* + Apigee

# Client Expectations

## Preferred Scenarios:

* + Gradual service degradation leading to complete unavailability (slow kill).
  + Staggered injection of chaos experiments.
  + Monitoring and adherence to a KPI of 250ms latency during experiments.

## Workflow:

* + Run chaos experiments.
  + Showcase outcomes of the experiments.
  + Re-architect infrastructure based on findings to enhance resilience and performance.
  + Repeat process to refine infrastructure design for optimal performance and resilience.

# Strategies to Ensure App Resilience

## Experiment Strategies:

* + Design experiment (approximately 10) to evaluate application resilience against predefined KPIs (e.g., 250ms latency).
  + Anticipate KPI breaches during resilience-focused re-architecture.
  + Reiterate the re-architecture cycle to optimize for both performance and resilience.

## Balanced Approach for Security (Depends on feasibility):

* + Address the trade-offs between **security**, **performance**, and **resilience**.
  + Develop a strategy to maintain equilibrium among these factors without significant degradation in app functionality.

# Action Items

1. **Team:** Finalize advanced experiment scenarios and set up environments for chaos testing.
2. **Architects:** Develop an initial plan for re-architecting based on potential experiment outcomes.
3. **Security Team:** Evaluate the impact of security measures on performance and resilience.
4. **All Stakeholders:** Align on a balanced approach for performance, resilience, and security.

# Plan Of Action:

* Step-1: Identifying parallel activities, eg: With our current existing infrastructure. We should be ready with all experiment codes.
* Step-2: Identifying dependent activities.
* Step-3: Setting up Kafka, Apigee, ELK stack.
* Step-4: Enhance the Kubernetes environment.
* Step-5: Finalize the applications apart from bank-of-anthos(creating microservices)
* Step-6: Setting up the database and establishing connections.
* Step-7: Integration of all the above components and Testing.

# Project management:

* The project is to be executed in Agile manner with proper project management principles in place.

**DevOps**: Prefer to have DevOps setup in place

# Documentation:

Documents & Manuals, Readme files should be prepared and uploaded over SharePoint on a regular basis.

# Resource requirement:

* + Need **Platform Engineers**
  + Need application developers
  + Need Project manager
  + Need DevOps Engineer
  + Need a **Solution** **Architect** & **Cloud Architect**

Timeline:

Minimum one month as a placeholder by the end of January