Enhanced DevOps Interview Introduction:

Hi, my name is Sumantra Upadhyay. I’m currently working with Reliance, and I have a total of 8 years of professional experience,

out of which the last 4 years relevat experiance in DevOps.

In my current role, I primarily work with Infrastructure as Code (IaC) using Terraform, where I follow a modular and reusable code approach

for provisioning cloud infrastructure. I have a strong command over writing and managing Terraform configurations using best practices such as input variables, outputs, workspaces, and remote backends.

I have handson exp in version control systems like Git, and I use GitHub extensively for source code management and collaboration. I’m familiar with branching strategies, pull requests, code reviews, and managing repositories in an enterprise setup.

On the CI/CD side, I have hands-on experience with Azure DevOps Pipelines and GitHub Actions, where I design and implement YAML-based automated pipelines to deploy infrastructure and applications seamlessly.

I also work with service connections, environment approvals, and pipeline templates to ensure secure and scalable delivery.

Apart from this, I’m also familiar with Terraform security and compliance tools like TFSec, Checkov, and TFLint for static code analysis, and I monitor infrastructure health using tools like Prometheus and Grafana.

Apart from this, I also integrate security and compliance checks within the CI/CD pipeline using static code analysis tools like TFSec, Checkov, and TFLint for Terraform. I also implement infrastructure monitoring as part of post-deployment validation using Prometheus and Grafana to ensure system health and reliability.

My experience spans across managing both on-prem and cloud infrastructure, and I always focus on automating repetitive tasks,

improving reliability, and maintaining infrastructure as code in a version-controlled and auditable manner.

Project Usecase:

In my organisation its own project where multiple servirs and devices are connected with each other when the tempreture is goes beyond its certain threshold

it is huge breakdown for the system and huge loss for the company and also is there is any fault is come related with its software faulure and hardware faulure

then services getting impacted, so for that there is basically application team who has injusted the monitoring solutions for those application which is running on monolithic and microsevices both on which frontend is runnig on react backend is .net, python and database few in onprem and few is on cloud so this is overall architecture of application

so for i'm working as a Devops er so for that we have developed the High level diagram once it is getting approved by the cloud architect and microsoft architect

then me and along with my team creating a low level diagram

Improved DevOps Project Use Case for Interview

In my organization, we are managing an in-house project where multiple servers and network devices are interconnected. The architecture is critical, as any increase in temperature beyond a defined threshold can lead to system breakdowns, resulting in significant business loss.

Moreover, any kind of software or hardware failure can severely impact services. To address this, the Application Team has integrated monitoring solutions to track the health and performance of applications, which are built on both monolithic and microservices architectures.

The frontend is developed in React,

The backend is built using .NET and Python,

The databases are hosted both on-premises and on cloud platforms.

This forms the overall architecture of our application ecosystem.

As a DevOps Engineer, my responsibility starts once the high-level solution architecture is finalized and approved by the Cloud Architect and Microsoft Solution Architect. Based on that:

I collaborate with my team to design a Low-Level Architecture Diagram (LLD).

We then start working on implementing Infrastructure as Code (IaC) using Terraform, following a modular approach.

Version control and collaboration are managed through Git and GitHub.

For CI/CD, we use Azure DevOps Pipelines and GitHub Actions, depending on the project needs.

We’ve integrated observability using Prometheus and Grafana to monitor both infrastructure and application-level metrics.

This DevOps pipeline ensures:

Fast and reliable deployment cycles, Automated rollback in case of failure, Infrastructure standardization across environments,

Proactive monitoring to avoid service downtime.