### **Tell Me About Yourself — GCP DevOps Focused Version**

Hi, my name is UmRao, and I’m from Rajahmundry. I started my career in the IT industry in 2015 and have around 3 years of experience as a **DevOps Engineer**, with my primary focus on **Google Cloud Platform (GCP)** services and tools.

In my current role, I handle the deployment, management, and automation of cloud infrastructure on **GCP**. I work extensively with services such as **Compute Engine**, **Google Kubernetes Engine (GKE)** for container orchestration, **Cloud Storage**, and **Pub/Sub** for messaging. I also set up and manage **Cloud Monitoring** and **Cloud Logging** to ensure system performance and availability.

For infrastructure provisioning, I use **Terraform** to build, modify, and version infrastructure across GCP projects. My daily activities include managing Kubernetes clusters on **GKE**, writing Dockerfiles, building Docker images, pushing them to **Google Container Registry (GCR)**, and deploying workloads via Kubernetes manifests and Helm charts.

In terms of CI/CD, I work with **Jenkins pipelines integrated with Bitbucket**, automating Maven builds, SonarQube code analysis, and Docker image builds. The images are stored in **GCR** and deployed to GKE clusters as part of the pipeline process.

I’ve also been involved in **cloud migration projects**, supporting the movement of applications from on-premises to **GCP environments**. This includes gathering client requirements from **Statements of Work (SOW)**, creating **Low-Level and High-Level Designs (LLD/HLD)**, and handling the implementation and post-migration operational documentation.

Additionally, I handle **incident management via ServiceNow tickets**, perform **root cause analysis (RCA)**, and ensure smooth running of cloud infrastructure and services.

While GCP is my primary platform, I also have hands-on experience with **AWS services like EC2, S3, ECS, and CloudWatch**, mainly for multi-cloud environments and migration projects.

### **Summary of Tools & Technologies (GCP-Focused):**

* **Cloud**: GCP (Compute Engine, GKE, Cloud Storage, Pub/Sub, Cloud Monitoring, Cloud Logging), AWS (basic experience)
* **Infrastructure as Code**: Terraform (GCP-focused)
* **CI/CD**: Jenkins, Bitbucket, Maven, SonarQube, Docker, GCR
* **Containerization & Orchestration**: Docker, GKE, Kubernetes, Helm
* **Monitoring**: Cloud Monitoring, Cloud Logging, New Relic
* **Security**: IAM (GCP), DLP solutions, Cloud Audit Logs
* **Process & Documentation**: SOW analysis, LLD, HLD, operational handover
* **Incident Management**: ServiceNow, RCA preparation

## **GCP Cloud Platform & Infra (Compute Engine, GKE, Storage, Pub/Sub)**

1. **What is the difference between Compute Engine and GKE in GCP? When do you use each?**

**A. Compute Engine is for managing individual virtual machines, while GKE (Google Kubernetes Engine) is for deploying and managing containerized applications using Kubernetes clusters.**

1. **Can you explain how GKE handles auto-scaling for pods and nodes?**

**A. GKE uses Horizontal Pod Autoscaler (HPA) to scale pods based on CPU/memory and Cluster Autoscaler to adjust node counts based on pending workloads.**

1. **How does Pub/Sub work in GCP? Can you describe a use case where you used Pub/Sub?**
2. **GCP Pub/Sub enables asynchronous, decoupled communication between services using a publisher-subscriber model; I used it to stream logs from GKE to BigQuery for real-time analysis.**
3. **What are the best practices to secure data in GCP Cloud Storage buckets?  
     
   A.Use IAM roles, enable uniform bucket-level access, enforce encryption (default or customer-managed keys), and set up audit logging for monitoring access.**

## **Terraform (Infrastructure as Code)**

1. **How do you manage Terraform state files in GCP projects?**

**A. Store Terraform state files securely in a GCS bucket with versioning enabled and use state locking with a backend configuration to prevent concurrent modifications.**

1. **Can you explain the use of Terraform modules in your infrastructure?**

**A. Terraform modules are used to encapsulate and reuse infrastructure code, ensuring consistency, scalability, and easier maintenance across multiple GCP environments.**

1. **What precautions do you take while running terraform apply in production environments?  
     
   A. Use terraform plan, review changes carefully, use targeted applies, and ensure approvals in CI/CD pipelines.**
2. **How do you handle Terraform variable files (.tfvars) securely in your projects?**

**A. Store sensitive variables in GCP Secret Manager or encrypted files, and exclude .tfvars from version control using .gitignore.**

## **CI/CD (Jenkins, Bitbucket, SonarQube, Docker, GCR)**

1. **What does your typical Jenkins pipeline look like for a GCP deployment?**

**A. Checkout code, validate Terraform, run plan, approval step, apply, then trigger post-deployment validations.**

1. **How do you integrate SonarQube into your Jenkins pipeline?**

**A. Use SonarQube scanner plugin in Jenkins to analyze code quality during the build stage.**

1. **How do you handle Docker image versioning and tagging in GCR?**

**A. Tag images using Git commit IDs, build numbers, or semantic versioning for traceability.**

1. **What is the role of Bitbucket in your CI/CD workflow?**

**A. Bitbucket hosts the source code and triggers Jenkins pipelines via webhook on PR or push.**

## **Containerization & Orchestration (Docker, Kubernetes, Helm)**

1. **Can you explain the difference between a Kubernetes Deployment and a StatefulSet?**

**A. Deployment manages stateless pods, while StatefulSet manages stateful applications with stable identities and storage.**

1. **How do you manage Helm chart versions and releases?**

**A. Use semantic versioning for charts and maintain a versioned Helm repository for releases.**

1. **How do you handle secret management in GKE clusters?**

**A. Use Kubernetes Secrets encrypted with KMS or integrate with GCP Secret Manager.**

1. **What are readiness and liveness probes in Kubernetes and how have you used them?**

**A. Readiness checks pod availability for traffic; liveness restarts unhealthy pods—I’ve configured both in production apps.**

## **Monitoring & Logging (Cloud Monitoring, Cloud Logging, New Relic)**

1. **What key metrics do you monitor in GKE clusters using Cloud Monitoring?**

**A. CPU, memory usage, disk I/O, pod status, and node health.**

1. **How do you configure custom logs and alerts in Cloud Logging?**

**A. Use log-based metrics and alert policies to trigger notifications for specific log patterns.**

1. **Have you used New Relic with GCP? If so, for what purpose?**

**A. Yes, for APM, infrastructure monitoring, and detailed transaction tracing in applications.**

## **Security (IAM, DLP, Audit Logs)**

1. **How do you manage IAM roles and permissions in GCP projects?**

**A. Follow the principle of least privilege using predefined or custom IAM roles with audit reviews.**

1. **What is Cloud Audit Logs in GCP and why is it important?**

**A. Captures admin and data access logs for security, compliance, and forensic analysis.**

1. **Can you describe how DLP works in GCP and give an example of its use in your projects?**

**A. DLP inspects and redacts sensitive data like PII; used for securing logs in GCS buckets.**

## **Process & Documentation (SOW, LLD/HLD, RCA, ServiceNow)**

1. **How do you gather technical requirements from an SOW?**

**A. Extract scope, deliverables, constraints, and acceptance criteria through detailed review and client discussions.**

1. **What is the difference between LLD and HLD? Can you give an example from your migration projects?**

**A. HLD defines architecture overview; LLD details configurations like IPs, ports—as used in VM migration plans.**

1. **Can you walk us through an RCA you prepared recently for a production incident?**

**A. Analyzed root cause of pod crash due to memory leak, recommended resource limits and code fix.**

1. **How do you manage and resolve ServiceNow tickets related to cloud incidents?**

**A. Prioritize based on impact, investigate logs/metrics, apply fixes, and document resolution in the ticket**