GCP DevOps - Interview Preparation (Terms & Interview Questions)

GCP Terms & Services for DevOps Engineer Interview

1. **GCP:** Google Cloud Platform
2. **Compute Engine:** Virtual Machines in GCP
3. **App Engine:** Platform for building and scaling web applications
4. **Cloud Functions:** Serverless execution for event-driven code
5. **Kubernetes Engine (GKE):** Managed Kubernetes service
6. **Cloud Storage:** Object storage for data
7. **Cloud SQL:** Managed relational database service
8. **Cloud Bigtable:** NoSQL database for large datasets
9. **Cloud Spanner:** Globally-distributed relational database
10. **Cloud Pub/Sub:** Real-time messaging service
11. **Cloud CDN:** Content Delivery Network for fast content delivery
12. **Cloud IAM:** Identity and Access Management
13. **Cloud Monitoring:** Monitors and troubleshoots cloud resources
14. **Cloud Logging:** Logs application and system events
15. **Cloud Trace:** Analyzes application performance
16. **Cloud Error Reporting:** Captures application errors
17. **Cloud Build:** Builds, tests, and deploys applications
18. **Cloud Deployment Manager:** Automates infrastructure deployment
19. **Cloud Shell:** Web-based command-line environment
20. **Cloud SDK:** Command-line tool for interacting with GCP
21. **VMs:** Virtual Machines
22. **Instances:** Running VMs in GCP
23. **Disks:** Persistent block storage for VMs
24. **Machine Images:** Templates for creating VMs
25. **SSH:** Secure Shell access for VMs
26. **Firewalls:** Controls inbound and outbound traffic
27. **Load Balancing:** Distributes traffic across VMs
28. **Autoscaling:** Automatically adjusts VM resources
29. **Buckets:** Storage units in Cloud Storage
30. **Object Versioning:** Maintains previous versions of objects
31. **Cloud CDN Edge Locations:** Locations for content caching
32. **Service Accounts:** Service identities for applications
33. **IAM Roles:** Permissions granted to users or service accounts
34. **Metrics:** Measurable aspects of a cloud resource
35. **Logs:** Records of events and system messages

36. **Alerts:** Notifications triggered by specific conditions
37. **Cloud Build Triggers:** Events that initiate builds
38. **Deployment Manager Templates:** Configuration files for infrastructure
39. **CI/CD:** Continuous Integration & Continuous Delivery
40. **Infrastructure as Code (IaC):** Defines infrastructure with code
41. **Terraform:** IaC tool for multi-cloud deployments
42. **Cloud Source Repositories:** Git repositories for code management
43. **Cloud Buildpacks:** Automates dependency management
44. **Cloud Armor:** Web Application Firewall (WAF)
45. **Cloud Key Management Service (KMS):** Manages encryption keys
46. **Cloud Identity and Access Management (IAM):** Identity management
47. **Secret Manager:** Stores and manages secrets
48. **Cloud Healthcare API:** Healthcare data management service
49. **Cloud Dataflow:** Stream and batch data processing service
50. **BigQuery:** Data warehouse for analytics at scale
51. **Cloud Datastore:** NoSQL document database
52. **Cloud Dataproc:** Manages Hadoop and Spark clusters
53. **Cloud Functions HTTP Trigger:** Invokes function via HTTP request
54. **Cloud Run:** Serverless platform for containerized applications
55. **Vertex AI:** Machine Learning (ML) platform
56. **Cloud TPU:** Tensor Processing Units for ML workloads
57. **Cloud Monitoring Agent:** Monitors specific resources
58. **Stackdriver (deprecated - replaced by Monitoring, Logging, Error Reporting)**
59. **Cloud Billing:** Monitors and manages GCP resource costs
60. **Cloud Billing Budget:** Sets spending limits for GCP resources
61. **Cloud VPN:** Creates a secure tunnel between on-premise network and GCP
62. **Cloud Interconnect:** Dedicated private connection between GCP and on-premise network
63. **Cloud DNS:** Managed Domain Name System service
64. **Cloud Router:** Manages virtual network routing
65. **VPC Peering:** Connects VPC networks within a region
66. **Cloud Load Balancing:** Distributes traffic across multiple backends
67. **Cloud Healthcare API:** Manages healthcare data in GCP
68. **Cloud Healthcare HL7v2 Stores:** Stores and processes HL7v2 healthcare data
69. **Cloud Healthcare Dicom Stores:** Stores and manages DICOM medical images
70. **Apigee API Management:** Manages APIs and microservices
71. **Cloud API Gateway:** Creates, publishes, and manages APIs
72. **Apigee Private Cloud:** On-premise deployment option for Apigee
73. **Cloud Monitoring Uptime Checks:** Monitors website or application availability
74. **Synthetic Monitoring:** Monitors application performance from various locations
75. **Cloud Monitoring SLOs:** Defines Service Level Objectives for applications
76. **Cloud Monitoring Alerting Policies:** Configures custom alerts based on metrics
77. **Cloud Debugger:** Debugs applications in production environment
78. **Cloud IDE:** Integrated Development Environment for GCP development

79. **Cloud Source Repositories Triggers:** Automate builds based on code changes
80. **Cloud Build Logs:** View build logs for troubleshooting
81. **Deployment Manager Deployment:** Specific deployment of an infrastructure template
82. **Service Accounts Keys:** Credentials for authenticating service accounts
83. **Organization Policies:** Enforces access control policies across GCP projects
84. **Cloud Resource Manager:** Manages GCP resources and projects
85. **Cloud IAM Conditions:** Adds context-based restrictions to IAM policies
86. **Cloud Monitoring Dashboards:** Visualize and monitor resource health
87. **Cloud Monitoring Notifications:** Sends notifications based on alerts or logs
88. **Cloud Monitoring Integrations:** Integrates monitoring data with other tools
89. **Cloud Build Substitutions:** Variables used in Cloud Build configurations
90. **Cloud Build Steps:** Individual actions performed during a Cloud Build
91. **Deployment Manager Variables:** Variables used in Deployment Manager templates
92. **Service Accounts for GKE:** Authenticates GKE clusters with GCP services
93. **GKE Workloads:** Different types of workloads managed by GKE (e.g., deployments, statefulsets)
94. **GKE Node Pools:** Groups of VMs with specific configurations for GKE clusters
95. **Kubernetes:** Open-source container orchestration platform (used by GKE)
96. **Cloud Storage Transfer Service:** Transfers data between on-premise storage and Cloud Storage
97. **Cloud Storage Signed URLs:** Grants temporary access to Cloud Storage objects
98. **Cloud Storage Lifecycle Management:** Automates data lifecycle management in Cloud Storage
99. **Cloud Storage Fusing:** Mounts Cloud Storage buckets as local disks
100. **Cloud Storage Cloud KMS Encryption:** Encrypts Cloud Storage objects with Cloud KMS keys

Interview Questions & Answers

Foundational GCP Knowledge (Easy):

1. **What are the core services offered by Google Cloud Platform (GCP)?**

   Answer: GCP offers a wide range of services, including

   - ➔ Compute Engine (VMs)
   - ➔ Cloud Storage (object storage)
   - ➔ Cloud SQL (managed relational databases)
   - ➔ Cloud Functions (serverless functions)
   - ➔ Kubernetes Engine (managed Kubernetes)

➔ Cloud Monitoring (monitoring and logging).

2. **Explain the concept of Infrastructure as Code (IaC) and its benefits in GCP.**

   Answer: IaC defines infrastructure in machine-readable files (e.g., Terraform, Cloud Build) allowing for version control, easier deployments, and consistent infrastructure across environments.

3. **Describe the difference between Cloud Storage buckets and Cloud SQL instances.**

   Answer: Cloud Storage buckets are for storing unstructured data objects (blobs, images, etc.), while Cloud SQL instances are managed relational databases with structured data.

Basic DevOps Practices (Medium):

4. **What is the CI/CD pipeline, and how does it benefit software development?**

   Answer: CI/CD (Continuous Integration/Delivery/Deployment) automates building, testing, and deployment of code. It improves development speed, reduces errors, and ensures consistent releases.

5. **Explain the role of version control systems like Git in a DevOps workflow.**

   Answer: Git tracks changes in code, allowing collaboration, version history, and rollbacks if needed. It integrates with CI/CD pipelines for automated builds and deployments.

6. **What are some best practices for securing GCP resources?**

   Answer: Best practices include IAM (Identity and Access Management) for granular access control, Cloud Key Management Service (KMS) for encryption keys, and Cloud Security Command Center for security posture monitoring.

GCP-Specific DevOps Tools (Medium):

7. **How can Cloud Build be used for building and deploying applications in GCP?**

Answer: Cloud Build is a serverless build service that automates building, testing, and deploying applications based on triggers (e.g., code changes in Git).

8. **Explain the purpose of Cloud Deployment Manager (CDM) in infrastructure provisioning.**

   Answer: CDM manages infrastructure deployments using declarative templates, allowing for consistent and repeatable infrastructure creation in GCP.

9. **What are the advantages of using Container Registry for container image management in GCP?**

   Answer: Container Registry stores and manages Docker container images used in deployments. It offers secure storage, versioning, and access control for container images.

Scenario-Based Problem Solving (Medium):

10. **Describe your approach to troubleshooting a deployment failure in a GCP environment.**

    Answer: (This is where you can showcase your problem-solving skills):

    - ➔ Gather logs from Cloud Monitoring and application logs.
    - ➔ Analyze error messages to identify the root cause (e.g., configuration issue, resource exhaustion).
    - ➔ Utilize debugging tools specific to the application framework.
    - ➔ Implement a fix and re-deploy the application.

Advanced GCP Services (Medium-Hard):

1. **Explain the benefits of using Cloud Load Balancing to distribute traffic across multiple instances in GCP.**

   Answer: Cloud Load Balancing distributes incoming traffic across VMs or containers, improving application scalability, availability, and fault tolerance.

2. **Describe how Cloud Spanner can be used for managing globally distributed databases in GCP.**

   Answer: Cloud Spanner is a globally-distributed relational database service that offers strong consistency, high availability, and automatic schema management across geographically dispersed locations.

3. **How can Cloud CDN (Content Delivery Network) be leveraged to optimize website performance for global users?**

   Answer: Cloud CDN caches static content (images, JS, etc.) at geographically distributed edge locations, reducing latency and improving website loading times for users worldwide.

CI/CD Pipeline Design and Automation (Hard):

4. **Explain your approach to designing a CI/CD pipeline for a microservices architecture in GCP.**

   Answer: Discuss breaking down the pipeline into stages (build, test, deploy) for each microservice. Utilize tools like Cloud Build triggers for code changes and service discovery mechanisms like Service Mesh for communication.

5. **How can infrastructure as code (IaC) be integrated with CI/CD pipelines for automated infrastructure provisioning?**

   Answer: IaC tools like Terraform can be integrated with CI/CD pipelines. Code changes trigger IaC deployments, ensuring infrastructure updates are in sync with application code changes.

Monitoring and Observability (Hard):

6. **Describe how Cloud Monitoring and Stackdriver Logging can be used for proactive application health monitoring in GCP.**

   Answer: Cloud Monitoring provides real-time metrics and alerting for infrastructure and application health. Stackdriver Logging collects and analyzes application logs for troubleshooting and debugging.

7. **Explain the concept of metrics, logs, and traces (ELT) in the context of cloud monitoring.**

   Answer: ELT refers to ingesting data into a monitoring system: Metrics (quantitative measurements), Logs (event data with timestamps), and Traces (distributed request paths). Analyzing all three provides a comprehensive view of application health.

Security and Compliance (Hard):

8. **How can Cloud IAM be used to implement the principle of least privilege for access control in GCP?**

   Answer: Cloud IAM allows granular control over who can access what resources in GCP. The principle of least privilege grants users only the minimum permissions required for their tasks.

9. **Explain the role of Cloud Key Management Service (KMS) in securing encryption keys in GCP.**

   Answer: KMS provides a centralized location for managing and controlling encryption keys used for data encryption at rest and in transit within GCP.

Scenario-Based Problem Solving (Hard):

10. **A critical application deployed on Kubernetes Engine (GKE) experiences high CPU usage. Describe your approach to diagnosing and resolving the issue.**

    Answer: (Demonstrate troubleshooting and analytical skills):

    - ➔ Utilize GKE monitoring tools to identify pods with high CPU consumption.
    - ➔ Analyze container logs to understand the cause (e.g., resource-intensive code, unexpected workload).
    - ➔ Optimize application code or scale resources (CPU/memory) as needed for affected pods.

Advanced GCP Services (Hard):

1. **Explain the use cases for Cloud Functions and Cloud Run serverless offerings in a DevOps workflow.**

   Answer: Discuss how Cloud Functions (event-triggered, code snippets) and Cloud Run (containerized microservices) can be used for building serverless applications, reducing infrastructure management overhead and scaling automatically.

2. **Describe your approach to implementing a disaster recovery (DR) strategy for a critical application deployed in GCP.**

   Answer: Discuss utilizing tools like Cloud Spanner replication or regional deployments for database redundancy. Leverage Cloud Storage versioning and backups for data protection. Utilize tools like Cloud Deployment Manager for consistent infrastructure recovery across regions.

CI/CD Pipeline Optimization (Hard):

3. **How can canary deployments be used to mitigate risk during application rollouts in GCP?**

   Answer: Explain deploying a new application version to a small subset of users first (canary) to identify and fix issues before a full rollout. Utilize tools like Cloud Load Balancing for controlled traffic routing during canary deployments.

4. **Describe the benefits of using infrastructure as code (IaC) testing tools like Terraform Cloud or Cloud Build for IaC validation.**

   Answer: Discuss integrating IaC testing tools into the CI/CD pipeline to validate IaC configurations before deployment, preventing errors and ensuring infrastructure adheres to best practices.

Monitoring and Distributed Tracing (Hard):

5. **Explain the concept of distributed tracing and its importance in troubleshooting microservices applications in GCP.**

   Answer: Discuss how distributed tracing tracks requests across multiple microservices, providing a holistic view of application behavior for troubleshooting performance issues and identifying bottlenecks. Utilize tools like Cloud Trace for distributed tracing in GCP.

6. **How can Cloud Monitoring dashboards and alerting be used to proactively identify and respond to application health issues?**

   Answer: Explain configuring custom dashboards in Cloud Monitoring to visualize key application metrics. Implement alerts based on thresholds to trigger notifications for potential problems, allowing for proactive intervention before issues escalate.

Security and Compliance (Hard):

7. **Describe the role of Cloud Identity and Access Management (Cloud IAM) in securing access to GCP resources for service accounts.**

   Answer: Explain how Cloud IAM controls access for service accounts used by applications to access GCP resources. Utilize service accounts with least privilege to minimize security risks.

8. **How can Cloud Security Command Center (SCC) be used to assess and improve the security posture of a GCP environment?**

   Answer: Discuss SCC's role in identifying security vulnerabilities, compliance issues, and recommending best practices for hardening GCP resources. Leverage SCC insights to continuously improve the security posture of your cloud environment.

Scenario-Based Problem Solving (Very Hard):

9. **A stateful application deployed on Kubernetes Engine (GKE) experiences data loss during pod restarts. Describe your approach to ensuring data persistence for this application.**

   Answer: (Showcase in-depth knowledge of container storage):

   - ➔ Identify the data storage needs of the application (persistent disks, databases).
   - ➔ Utilize Persistent Volumes (PVs) and Persistent Volume Claims (PVCs) to manage persistent storage for containers, ensuring data survives pod restarts.
   - ➔ Consider managed databases like Cloud SQL for stateful applications requiring relational data storage.

10. **A mission-critical application deployed on GCP experiences a sudden performance degradation. You suspect a network latency issue. Describe your approach to diagnosing and resolving the problem.**

    Answer: (Demonstrate advanced troubleshooting skills):

    - ➔ Utilize Cloud Monitoring network metrics to identify latency spikes or packet loss.
    - ➔ Analyze Cloud Load Balancing logs to pinpoint affected instances or regions.
    - ➔ Consider network tracing tools like Cloud Trace to identify the path of network requests and pinpoint bottlenecks.
    - ➔ Scale resources or adjust network configurations as needed to resolve latency issues.

Hope you find this document helpful for your GCP DevOps Interview.

For more such content you can check : https://techyoutube.com/

Now, to Support, just follow me on below socials (No Cheating Please)

Telegram: https://t.me/LearnDevOpsForFree

Twitter: https://twitter.com/techyoutbe

Youtube: https://www.youtube.com/@T3Ptech