Google Cloud

Professional Cloud DevOps Engineer

Certification exam guide

A Professional Cloud DevOps Engineer implements processes and capabilities throughout the systems development lifecycle using Google-recommended methodologies and tools. They enable efficient software and infrastructure delivery while balancing reliability with delivery speed. They optimize and maintain production systems and services for both performance and cost.

Section 1: Bootstrapping and maintaining a Google Cloud organization (~20% of the exam)

1.1 Designing the overall resource hierarchy for an organization. Considerations include:

- Organizing resources (e.g., application-centric, projects, folders)
- Shared networking (e.g., Shared VPC, VPC Network Peering, Private Service Connect)
- Multi-project monitoring and logging
- Identity and Access Management (IAM) roles and organization-level policies
- Creating and managing service accounts
- Data residency

1.2 Managing infrastructure. Considerations include:

- Infrastructure-as-code tooling and managed services (e.g., Infrastructure Manager, Cloud Foundation Toolkit, Config Connector, GitOps, Terraform, Helm)
- Making infrastructure changes using Google-recommended practices and blueprints
- Automation with scripting (e.g., Python, Go)

1.3 Designing a CI/CD architecture stack in Google Cloud, hybrid, and multi-cloud environments. Considerations include:

- Continuous integration (CI) with Cloud Build
- Continuous delivery (CD) with Cloud Deploy, including Kustomize and Skaffold
- Artifact Registry configuration
- Widely used third-party tooling (e.g., Git, Jenkins, Argo CD, Packer, kpt)
- Security of CI/CD tooling

1.4 Managing multiple environments (e.g., staging, production). Considerations include:

- Managing ephemeral environments
- Managing configuration and policy
- Managing Google Kubernetes Engine (GKE) clusters across an enterprise (e.g., fleets)
- Safe and secure patching and upgrading practices

1.5 Enabling secure cloud development environments. Considerations include:

- Configuring and managing cloud development environments (e.g., Cloud Workstations, Cloud Shell)
- Bootstrapping environments with required tooling (e.g., custom images, IDE, Cloud SDK)
- Leveraging AI to assist with development and operations (e.g., Gemini Code Assist, Gemini Cloud Assist, Gemini CLI)

Section 2: Building and implementing CI/CD pipelines, including continuous testing, for application, infrastructure, and machine learning workloads (~25% of the exam)

2.1 Designing pipelines. Considerations include:

- CI/CD of applications and infrastructure
- Artifact management with Artifact Registry
- Deployment to hybrid and multi-cloud environments (e.g., GKE)
- CI/CD pipeline triggers
- Configuring deployment processes (e.g., approval flows)

2.2 Implementing and managing pipelines. Considerations include:

- Auditing and tracking deployments (e.g., Artifact Registry, Cloud Build, Cloud Deploy, Cloud Audit Logs)
- Deployment strategies (e.g., canary, blue/green, rolling, traffic splitting, feature flags) and defining success metrics based on application or ML pipeline telemetry
- Troubleshooting and mitigating deployment issues

2.3 Managing pipeline configuration and secrets. Considerations include:

- Key management (e.g., Cloud Key Management Service)
- Configuration and secret management (e.g., Secret Manager, Certificate Manager, Parameter Manager, Workload Identity Federation)
- Build versus runtime secret injection

2.4 Securing the deployment pipeline. Considerations include:

- Artifact Analysis and vulnerability scanning
- Software supply chain security (e.g., Binary Authorization, Supply-chain Levels for Software Artifacts [SLSA] framework)
- IAM policies based on environment

Section 3: Applying site reliability engineering practices (~18% of the exam)

3.1 Balancing change, velocity, and reliability of the service. Considerations include:

- Defining SLIs (e.g., availability, latency), SLOs, and SLAs
- Error budgets (e.g., Cloud Service Mesh definitions)
- Opportunity cost of risk and reliability (e.g., number of “nines”)

3.2 Managing service lifecycle. Considerations include:

- Service management (e.g., planning, deployment, maintenance, retirement)
- Capacity planning (e.g., quotas, limits, reservations, Dynamic Workload Scheduler)
- Autoscaling (e.g., managed instance groups, Cloud Run, GKE)

3.3 Mitigating incident impact on users. Considerations include:

- Draining/redirecting traffic
- Adding capacity
- Rollback strategies

Section 4: Implementing observability practices and troubleshooting issues (~25% of the exam)

4.1 Instrumenting and collecting telemetry. Considerations include:

- Collecting and importing logs (e.g., Ops Agent, OpenTelemetry, Cloud Audit Logs, VPC Flow Logs, Cloud Service Mesh)
- Optimizing logs (e.g., filtering, sampling, exclusions, cost management, source considerations)
- Collecting metrics (e.g., from applications, platforms, networking, Cloud Service Mesh, Google Cloud Managed Service for Prometheus, hybrid/multi-cloud environments)
- Creating synthetic monitors to proactively probe application endpoints and workflows
- Creating custom metrics, including log-based metrics

4.2 Managing and analyzing logs. Considerations include:

- Analyzing logs using the Logs Explorer and the Logging query language
- Exporting and retaining logs (e.g., routing to BigQuery, Pub/Sub, Cloud Storage)
- Handling sensitive data (e.g., using log processors to redact personally identifiable information [PII], protected health information [PHI])
- Using Gemini Cloud Assist for AI-powered log analysis

4.3 Managing metrics, dashboards, and alerts. Considerations include:

- Analyzing metrics using the Metrics Explorer
- Managing dashboards (e.g., creating, filtering, sharing, playbooks, PromQL)
- Configuring alerting and alerting policies (e.g., SLIs, SLOs, cost control)
- Integrating with third-party alerting tools (e.g., webhooks, PagerDuty, Rootly)
- Leveraging Gemini Cloud Assist for metrics interpretation

4.4 Capturing and analyzing distributed traces. Considerations include:

- Utilizing tracing frameworks (e.g., OpenTelemetry)
- Analyzing trace waterfalls and spans
- Correlating trace IDs with structured logs
- Employing Gemini Cloud Assist for trace analysis

4.5 Troubleshooting issues. Considerations include:

- Infrastructure issues
- CI/CD pipeline issues
- Application issues
- Observability issues
- Performance and latency issues

Section 5: Optimizing performance and cost (~12% of the exam)

5.1 Collecting performance information in Google Cloud. Considerations include:

- Application performance monitoring
- Active Assist insights and recommendations

5.2 Implementing FinOps practices for optimizing resource utilization and costs. Considerations include:

- Observability costs
- Spot virtual machines (VMs)
- Optimizing resource usage for cost and efficiency
- Infrastructure cost planning (e.g., committed-use discounts, sustained-use discounts, network tiers)
- Leveraging Google Cloud recommenders (e.g., cost, security, performance, manageability, reliability)
- Optimizing individual workload costs (e.g., GKE, Cloud Run, Compute Engine)