Google Cloud

Professional Cloud DevOps Engineer

This is the new version of the Professional Cloud DevOps Engineer exam guide. This exam guide will be live on November 23. If you plan to take the Professional Cloud DevOps Engineer exam on or after November 23, review this exam guide. **If you plan to take the Professional Cloud DevOps Engineer exam before November 23, review the current exam guide.**

Certification exam guide

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

Section 1: Establishing and bootstrapping foundational infrastructure and organizational governance (~18% of the exam)

1.1 Designing the overall resource hierarchy for a Google Cloud organization. Considerations include:

- Organizing resources (e.g., application-centric, projects, folders)
- Shared networking (e.g., Shared VPC, VPC Network Peering, Network Connectivity Center, 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., Application Design Center, Infrastructure Manager, Config Connector, GitOps, Terraform, Helm)
- Making infrastructure changes using Google-recommended practices and blueprints (e.g., Terraform modules and blueprints)
- Automation with scripting (e.g., Python, Go)
- Implementing Provisioned Throughput for agent reliability

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

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

1.4 Managing multiple runtime 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)
- Managing serverless environments
- Safe and secure patching and upgrading practices
- Isolating agent execution environments (e.g., GKE Agent Sandbox)

1.5 Managing multiple 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, Google Cloud SDK)
- Leveraging AI to assist with development and operations (e.g., Antigravity, Gemini Cloud Assist)
- Configuring and leveraging Model Context Protocol (MCP) servers

Section 2: Building, implementing, and maintaining CI/CD pipelines, including continuous testing, for application, infrastructure, and ML workloads (~30% of the exam)

2.1 Building pipelines. Considerations include:

- CI/CD of applications and infrastructure
- Artifact management with Artifact Registry
- Deployment to hybrid and multicloud 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

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
- Managing short-lived credentials

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, Cloud Run, Cloud Monitoring)
- 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)
- Implementing feedback loops for continuous improvement

3.3 Mitigating incident impact on users. Considerations include:

- Draining/redirecting traffic
- Adding capacity
- Rollback strategies

Section 4: Implementing observability practices and troubleshooting issues (~22% 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/multicloud environments)
- Creating synthetic monitors to proactively probe application endpoints and workflows
- Creating custom metrics, including log-based metrics
- Emitting metrics or logs from services and agents (e.g., GKE, Cloud Run, Gemini Enterprise Agent Platform, Agent Development Kit, Antigravity)

4.2 Managing and analyzing logs and traces. Considerations include:

- Analyzing logs using the Logs Explorer (e.g., AI-assisted analysis)
- 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])
- Capturing and analyzing distributed traces (e.g., OpenTelemetry, trace waterfalls, span analysis, correlating trace IDs with structured logs, AI-assisted 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, Slack, Discord)
- Performing AI-assisted metrics interpretation

4.4 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 and utilization information in Google Cloud. Considerations include:

- Application and agent performance monitoring
- Active Assist insights and recommendations

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

- Observability costs
- Spot 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)
- Using Provisioned Throughput to control costs
- Optimizing token consumption