Advanced Cloud Engineering Certification (Post■PCA)

A professional■grade certification focused on GCP platform engineering, Kubernetes at scale, Terraform■first infrastructure, CI/CD automation, security engineering, and Agentic AI with Vertex AI.

Target Audience: Senior Cloud Engineers, Platform Engineers, DevOps/SREs, AI■adjacent engineers preparing for real production systems beyond entry■level certifications.

Domain 1 – Advanced GCP Infrastructure & Architecture

Exam Weight: 20%

Tests the ability to design, implement, and govern complex GCP environments using Infrastructure as Code. Focus is on organization design, network topology, IAM boundaries, cost controls, and operational safety.

Lab 1: Multi■Environment GCP Landing Zone

Problem: Design and deploy a multi■project GCP landing zone using Terraform.

Success Criteria:

- Environment■isolated projects (dev/stage/prod)

- Shared VPC implemented

- Least■privilege IAM enforced

- Remote Terraform state managed securely

Architecture: Terraform → GCP Org → Projects → Shared VPC → IAM

Hints: Use Terraform modules, backend in GCS, and folders for policy enforcement.

Extensions: Add org policies, budget alerts, and cost anomaly detection.

Difficulty Multiplier (Hard Mode): Automated break■glass access with TTL■based IAM revocation.

Domain 2 – Kubernetes & GKE Platform Engineering

Exam Weight: 20%

Evaluates mastery of Kubernetes platform design on GKE, including security hardening, network isolation, workload identity, and production■grade operations.

Lab 2: Production■Grade GKE Platform

Problem: Deploy a hardened private GKE cluster supporting multiple teams.

Success Criteria:

- Private GKE cluster

- Workload Identity enabled

- Network policies enforced

- Separate node pools per workload class

Architecture: Terraform → GKE → Namespaces → Node Pools → Services

Hints: Apply Pod Security Standards and workload identity from day one.

Extensions: GitOps with ArgoCD or Flux.

Difficulty Multiplier (Hard Mode): Multi■cluster GKE with service mesh (Istio/Anthos).

Domain 3 – CI/CD & Platform Automation

Exam Weight: 15%

Covers continuous delivery, infrastructure pipelines, and platform automation using GitHub Actions and policy■driven workflows.

Lab 3: GitHub Actions for Terraform & GKE

Problem: Build a CI/CD pipeline validating infrastructure and deploying workloads.

Success Criteria:

- terraform fmt / validate / plan

- Approval gates by environment

- Automated GKE deployment

Architecture: GitHub → Actions → Terraform → GKE

Hints: Use OIDC authentication instead of static credentials.

Extensions: Policy■as■Code with OPA or Sentinel.

Difficulty Multiplier (Hard Mode): Ephemeral preview environments per pull request.

Domain 4 – Data & Stateful Systems

Exam Weight: 15%

Focuses on running stateful workloads in cloud■native architectures, balancing reliability, security, and operational simplicity.

Lab 4: Stateful Microservices on GKE

Problem: Run application workloads backed by Cloud SQL securely.

Success Criteria:

- Private Cloud SQL instance

- Secrets managed via Secret Manager

- Backups and restores tested

Architecture: GKE → Cloud SQL → Secret Manager

Hints: Use sidecar or Cloud SQL Auth Proxy.

Extensions: Read replicas and failover testing.

Difficulty Multiplier (Hard Mode): Zero■downtime migration to AlloyDB.

Domain 5 – Vertex AI & Agentic Systems

Exam Weight: 20%

Assesses ability to design, deploy, and operate AI■enabled systems using Vertex AI, including agentic workflows and foundation models.

Lab 5: Agentic AI Service with Vertex AI

Problem: Build an AI agent capable of task reasoning and tool usage.

Success Criteria:

- Vertex AI endpoint deployed

- Prompt orchestration implemented

- Inference logging enabled

Architecture: Client → API → Vertex AI → Agent → Tools

Hints: Start with Gemini models and simple prompt chains.

Extensions: Persistent memory and tool calling.

Difficulty Multiplier (Hard Mode): Multi■agent coordination with task decomposition.

Domain 6 – Security, Reliability & Resilience Engineering

Exam Weight: 10%

Tests zero■trust design, security engineering, observability, and failure modeling within distributed systems.

Lab 6: Zero Trust GCP Architecture

Problem: Implement a zero■trust service architecture.

Success Criteria:

- Identity■Aware Proxy enforced

- No public IPs

- Centralized audit logging

Architecture: User → IAP → GKE → Internal Services

Hints: Adopt BeyondCorp principles.

Extensions: SIEM integration and alerting.

Difficulty Multiplier (Hard Mode): Chaos engineering with automated rollback.