



Logging, Monitoring, and Observability in Google Cloud

Duration 3 day(s) (T-STACKD-B)

Control your infrastructure and application

Official Training



Description

Learn how to monitor, troubleshoot, and improve your infrastructure and application performance. Guided by the principles of Site Reliability Engineering (SRE), this course features a combination of lectures, demos, hands-on labs, and real-world case studies. In this course, you'll gain experience with full-stack monitoring, real-time log management and analysis, debugging code in production, and profiling CPU and memory usage.

For each lab, Qwiklabs offers a free set of resources for a fixed amount of time and a clean environment with permissions.

Goals

- Plan and implement a well-architected logging and monitoring architecture
- Measure and avoid customer pain with Service Level Indicators (SLIs) and Service Level Objectives (SLOs)
- Visualize cloud health with dashboards
- Construct automated alerts
- Monitor, troubleshoot, and improve Google Cloud infrastructure
- Write, analyze, and export Google Cloud logs
- Find code defects, identify bottlenecks, and improve performance of production code
- Optimize monitoring costs

Public

- Cloud architects, administrators, and SysOps personnel
- Cloud developers and DevOps personnel

Prerequisites

- [Google Cloud Platform Fundamentals: Core Infrastructure](#) or equivalent experience
- Basic scripting or coding ability
- Proficiency with command-line tools and Linux operating system environments

Structure

50% Theory, 50% Practice

Program

- **Introduction to Google Cloud Monitoring Tools**
 - Explain the purpose and capabilities of Google Cloud operations-focused components: Logging, Monitoring, Error Reporting, and Incident Response and Management (IRM)
 - Explain the purpose and capabilities of Google Cloud application performance management focused components: Debugger, Trace, Profiler, and Service Monitoring
- **Avoiding Customer Pain**
 - Construct a monitoring base from the four golden signals: latency, traffic, errors, and saturation
 - Define critical system measures with Service Level Indicators (SLIs)
 - Use Service Level Objectives (SLOs) and Service Level Agreements (SLAs) to measure, and avoid, customer pain
 - Achieve developer and operation harmony with SLO based error budgets
- **Monitoring Critical Systems**
 - Choose best practice monitoring project architectures
 - Differentiate Cloud IAM roles for monitoring
 - Use the default dashboards appropriately
 - Build custom dashboards to show resource consumption and application load
 - Define uptime checks to track aliveness and latency
- **Alerting Policies**
 - Develop alerting strategies
 - Define alerting policies
 - Add notification channels
 - Identify types of alerts and common uses for each
 - Construct and alert on resource groups
 - Manage alerting policies programmatically
- **Advanced Logging and Analysis**
 - Identify and choose among resource tagging approaches
 - Define log sinks (inclusion filters) and exclusion filters
 - Create metrics based on logs
 - Export logs to BigQuery
- **Working with Audit Logs**
 - Use Admin Activity, Data Access, and System Event audit logs
 - Track who, did what, and when
- **Configuring Google Cloud Services for Observability**
 - Integrate Logging and Monitoring agents into Compute Engine VMs and images
 - Enable and utilize Kubernetes Monitoring
 - Extend and clarify Kubernetes Monitoring with Prometheus
 - Expose custom metrics through code, and with the help of OpenCensus
- **Monitoring the Google Cloud VPC**
 - Collect and analyze VPC Flow, Firewall Rules, and Cloud NAT logs
 - Enable Packet Mirroring
 - Explain the capabilities of Network Intelligence Center
- **Managing Incidents**
 - Handle incidents systematically
 - Define incident management roles and communication channels
 - Mitigate incident impact

- Troubleshoot root causes
- Resolve the incident
- Document incident in a postmortem process

- **Investigating Application Performance Issues**
 - Use Error Reporting to identify and understand your application errors.
 - Debug production code to correct code defects
 - Trace latency through layers of service interaction to eliminate performance bottlenecks
 - Profile and identify resource-intensive functions in an application

- **Optimizing the Costs of Monitoring**
 - Analyze resource utilization cost for monitoring related components within Google Cloud
 - Implement best practices for controlling the cost of monitoring within Google Cloud