multisoft
Systems

Google Cloud DevOps Engineer Professional Training

COURSE CONTENT

GET IN TOUCH

Multisoft Systems
B - 125, Sector - 2, Noida

(+91) 9810-306-956

info@multisoftsystems.com

www.multisoftsystems.com

About Multisoft

Train yourself with the best and develop valuable in-demand skills with Multisoft Systems. A leading certification training provider, Multisoft collaborates with top technologies to bring world-class one-on-one and certification trainings. With the goal to empower professionals and business across the globe, we offer more than 1500 training courses, which are delivered by Multisoft's global subject matter experts. We offer tailored corporate training; project Based Training, comprehensive learning solution with lifetime e-learning access, after training support and globally recognized training certificates.

About Course

Multisoft Systems presents a comprehensive Google Cloud DevOps Engineer Professional Training that equips professionals with the skills and knowledge to excel in the dynamic field of DevOps within the Google Cloud environment. This training program offers a deep dive into essential concepts, tools, and practices, enabling participants to streamline development, deployment, and operations processes.

Module 1: Applying site reliability engineering principles to a service

1.1 Balance change, velocity, and reliability of the service

- ✓ Discover SLIs (e.g., availability, latency)
- ✓ Define SLOs and understand SLAs
- ✓ Agree to consequences of not meeting the error budget
- ✓ Construct feedback loops to decide what to build next
- ✓ Eliminate toil via automation

1.2 Manage service life cycle

- ✓ Manage a service (e.g., introduce a new service, deploy, maintain, and retire it)
- ✓ Plan for capacity (e.g., quotas and limits management)

1.3 Ensure healthy communication and collaboration for operations

- ✓ Prevent burnout (e.g., set up automation processes to prevent burnout)
- ✓ Foster a learning culture
- ✓ Foster a culture of blamelessness

Module 2: Building and implementing CI/CD pipelines for a service

2.1 Design CI/CD pipelines

- ✓ Creating and storing immutable artifacts with Artifact Registry
- ✓ Deployment strategies with Cloud Build and Spinnaker
- ✓ Deployment to hybrid and multicloud environments with Anthos, Spinnaker, and Kubernetes

- ✓ Artifact versioning strategy with Cloud Build and Artifact Registry
- ✓ CI/CD pipeline triggers with Cloud Source Repositories, external SCM, and Pub/Sub
- ✓ Testing a new version with Spinnaker
- ✓ Configuring deployment processes (e.g., approval flows)

2.2 Implement CI/CD pipelines

- ✓ CI with Cloud Build
- ✓ CD with Cloud Build
- ✓ Open-source tooling (e.g., Jenkins, Spinnaker, GitLab, Concourse)
- ✓ Auditing and tracing of deployments (e.g., CSR, Artifact Registry, Cloud Build, Cloud Audit Logs)

2.3 Manage configuration and secrets

- ✓ Secure storage methods
- ✓ Secret rotation and config changes

2.4 Manage infrastructure as code

- ✓ Terraform
- ✓ Infrastructure code versioning
- ✓ Make infrastructure changes safer
- ✓ Immutable architecture

2.5 Deploy CI/CD tooling

- ✓ Centralized tools vs. multiple tools (single vs. multi-tenant)
- ✓ Security of CI/CD tooling

2.6 Manage different development environments (e.g., staging, production)

- ✓ Decide on the number of environments and their purpose

- ✓ Create environments dynamically per feature branch with GKE
- ✓ Local development environments with Docker, Cloud Code, Skaffold

2.7 Secure the deployment pipeline

- ✓ Vulnerability analysis with Artifact Registry
- ✓ Binary Authorization
- ✓ IAM policies per environment

Module 3: Implementing service monitoring strategies

3.1 Manage application logs

- ✓ Collecting logs from Compute Engine, GKE with Cloud Logging, Fluentd
- ✓ Collecting third-party and structured logs with Cloud Logging, Fluentd
- ✓ Sending application logs directly to the Cloud Logging API

3.2 Manage application metrics with Cloud Monitoring

- ✓ Collecting metrics from Compute Engine
- ✓ Collecting GKE/Kubernetes metrics
- ✓ Use Metrics Explorer for ad hoc metric analysis

3.3 Manage Cloud Monitoring platform

- ✓ Creating a monitoring dashboard
- ✓ Filtering and sharing dashboards
- ✓ Configure third-party alerting in Cloud Monitoring (e.g., PagerDuty, Slack)
- ✓ Define alerting policies based on SLIs with Cloud Monitoring
- ✓ Automate alerting policy definition with Terraform
- ✓ Implementing SLO monitoring and alerting with Cloud Monitoring
- ✓ Understand Cloud Monitoring integrations (e.g., Grafana, Big Query)
- ✓ Using SIEM tools to analyze audit/flow logs (e.g., Splunk, Datadog)
- ✓ Design Cloud Monitoring metrics scopes

3.4 Manage Cloud Logging platform

- ✓ Enabling data access logs (e.g., Cloud Audit Logs)
- ✓ Enabling VPC flow logs
- ✓ Viewing logs in the Google Cloud Console
- ✓ Using basic vs. advanced logging filters
- ✓ Implementing logs-based metrics
- ✓ Understanding the logging exclusion vs. logging export
- ✓ Selecting the options for logging export
- ✓ Implementing a project-level / org-level export
- ✓ Viewing export logs in Cloud Storage and BigQuery
- ✓ Sending logs to an external logging platform

3.5 Implement logging and monitoring access controls

- ✓ Set ACL to restrict access to audit logs with IAM, Cloud Logging
- ✓ Set ACL to restrict export configuration with IAM, Cloud Logging
- ✓ Set ACL to allow metric writing for custom metrics with IAM, Cloud Monitoring

Module 4: Optimizing service performance

4.1 Identify service performance issues

- ✓ Evaluate and understand user impact
- ✓ Utilize Google Cloud’s operations suite to identify cloud resource utilization
- ✓ Utilize Cloud Trace and Cloud Profiler to profile performance characteristics
- ✓ Interpret service mesh telemetry
- ✓ Troubleshoot issues with the image/OS
- ✓ Troubleshoot network issues (e.g., VPC flow logs, firewall logs, latency, view network details)

4.2 Debug application code

- ✓ Application instrumentation
- ✓ Cloud Debugger
- ✓ Cloud Logging
- ✓ Cloud Trace
- ✓ Debugging distributed applications
- ✓ App Engine local development server
- ✓ Error Reporting
- ✓ Cloud Profiler

4.3 Optimize resource utilization

- ✓ Identify resource costs
- ✓ Identify resource utilization levels
- ✓ Develop plan to optimize areas of greatest cost or lowest utilization
- ✓ Manage preemptible VMs
- ✓ Utilize committed use discounts where appropriate
- ✓ TCO considerations (e.g., security, logging, networking)
- ✓ Consider network pricing

Module 5: Managing service incidents

5.1 Coordinate roles and implement communication channels during a service incident

- ✓ Define roles (incident commander, communication lead, operations lead)
- ✓ Handle requests for impact assessment
- ✓ Provide regular status updates, internal and external
- ✓ Record major changes in incident state (e.g., When mitigated? When is all clear?)
- ✓ Establish communications channels (e.g., email, IRC, Hangouts, Slack, phone)
- ✓ Scaling response team and delegation

- ✓ Avoid exhaustion / burnout
- ✓ Rotate / hand over roles
- ✓ Manage stakeholder relationships

5.2 Investigate incident symptoms impacting users

- ✓ Identify probable causes of service failure
- ✓ Evaluate symptoms against probable causes; rank probability of cause based on observed behavior
- ✓ Perform investigation to isolate most likely actual cause
- ✓ Identify alternatives to mitigate issue

5.3 Mitigate incident impact on users

- ✓ Roll back release
- ✓ Drain / redirect traffic
- ✓ Turn off experiment
- ✓ Add capacity

5.4 Resolve issues with deployments (e.g., Cloud Build, Jenkins)

- ✓ Code change / fix bug
- ✓ Verify fix
- ✓ Declare all-clear

5.5 Document issue in a postmortem

- ✓ Document root causes
- ✓ Create and prioritize action items
- ✓ Communicate postmortem to stakeholders