



Course Content - Google Cloud Professional

Bittu Kumar

Contact@devopsschool.com

Call: +91 700 483 5706

MICROLAND®



onmobile

Course Duration: 60 Hours

Description:

Nowadays, Google Cloud Professional Course very demanding among IT professionals or fresher's because it is one of the fastest growing Cloud Provider in the Industry. DevOpsSchool is one of the top institutes for Google Cloud Professional Course Online and classroom training program by experts in Hyderabad and Bangalore.

Prerequisites:

- Complete the Course Cloud Engineering Professional Certificate.
- Complete Associate Cloud Engineer sample questions.
- Google Cloud Associate Cloud Engineer certification exam.

Agenda:

Applying site reliability engineering principles to a service

Balance change, velocity, and reliability of the service:

- Discover SLIs (availability, latency, etc.)
- Define SLOs and understand SLAs
- Agree to consequences of not meeting the error budget
- Construct feedback loops to decide what to build next
- Toil automation

Manage service life cycle:

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



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

Building and implementing CI/CD pipelines for a service

Design CI/CD pipelines:

- Immutable artifacts with Container Registry
- Artifact repositories with Container Registry
- Deployment strategies with Cloud Build, Spinnaker
- Deployment to hybrid and multi-cloud environments with Anthos, Spinnaker, Kubernetes
- Artifact versioning strategy with Cloud Build, Container Registry
- CI/CD pipeline triggers with Cloud Source Repositories, Cloud Build GitHub App, Cloud Pub/Sub
- Testing a new version with Spinnaker
- Configure deployment processes (e.g., approval flows)

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, Cloud Build, Cloud Audit Logs)

Manage configuration and secrets:

- Secure storage methods
- Secret rotation and config changes

Manage infrastructure as code:

- Terraform / Cloud Deployment Manager
- Infrastructure code versioning
- Make infrastructure changes safer
- Immutable architecture

Deploy CI/CD tooling:

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

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

- Decide on the number of environments and their purpose
- Create environments dynamically per feature branch with GKE, Cloud Deployment Manager
- Local development environments with Docker, Cloud Code, Skaffold

Secure the deployment pipeline:

- Vulnerability analysis with Container Registry
- Binary Authorization
- IAM policies per environment

Implementing service monitoring strategies

Manage application logs:

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



OPTUM®



HCL

Manage application metrics with Stackdriver Monitoring:

- Collecting metrics from Compute Engine
- Collecting GKE/Kubernetes metrics
- Use metric explorer for ad hoc metric analysis

Manage Stackdriver Monitoring platform:

- Creating a monitoring dashboard
- Filtering and sharing dashboards
- Configure third-party alerting in Stackdriver Monitoring (i.e., PagerDuty, Slack, etc.)
- Define alerting policies based on SLIs with Stackdriver Monitoring
- Automate alerting policy definition with Cloud DM or Terraform
- Implementing SLO monitoring and alerting with Stackdriver Monitoring
- Understand Stackdriver Monitoring integrations (e.g., Grafana, BigQuery)
- Using SIEM tools to analyze audit/flow logs (e.g., Splunk, Datadog)
- Design Stackdriver Workspace strategy

Manage Stackdriver Logging platform:

- Enabling data access logs (e.g., Cloud Audit Logs)
- Enabling VPC flow logs
- Viewing logs in the GCP 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

Implement logging and monitoring access controls:

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

Optimizing service performance

Identify service performance issues:

- Evaluate and understand user impact (Stackdriver Service Monitoring for App Engine, Istio)
- Utilize Stackdriver to identify cloud resource utilization
- Utilize Stackdriver Trace/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)

Debug application code:

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



Optimize resource utilization:

- Identify resource costs
- Identify resource utilization levels
- Develop plan to optimize areas of greatest cost or lowest utilization
- Manage pre-emptible VMs
- Work with committed-use discounts
- TCO considerations
- Consider network pricing

Managing service incidents

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 (When mitigated? When all clear? etc.)
- Establish communications channels (email, IRC, Hangouts, Slack, phone, etc.)
- Scaling response team and delegation
- Avoid exhaustion / burnout
- Rotate / hand over roles
- Manage stakeholder relationships

Investigate incident symptoms impacting users:

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



GE Healthcare



Cognizant

Mitigate incident impact on users:

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

Resolve issues (e.g., Cloud Build, Jenkins):

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

Document issue in a postmortem:

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