

## **GOOGLE CLOUD CERTIFIED - Professional Cloud Developer**

### **Duration**

4 Days

### **Pre-requisite**

None

### **Target Audience**

Programmers/Developers/Testers

### **Certification Body**

Google

### **Introduction**

Professional Cloud Developer

A Professional Cloud Developer builds scalable and highly available applications using Google recommended practices and tools that leverage fully managed services. This individual has experience with next generation databases, runtime environments and developer tools. They also have proficiency with at least one general purpose programming language and are skilled with using Stackdriver to produce meaningful metrics and logs to debug and trace code.

The Professional Cloud Developer exam assesses your ability to:

- Design highly scalable, available, and reliable cloud-native applications
- Build and test applications
- Deploy applications
- Integrate Google Cloud Platform services

- Manage application performance monitoring

## **Workshop Content**

### **Section 1: Designing highly scalable, available, and reliable cloud-native applications**

#### 1.1 Designing performant applications and APIs. Considerations include:

- Infrastructure as a Service vs. Container as a Service vs. Platform as a Service (e.g., autoscaling implications)
- Portability vs. platform-specific design
- Evaluating different services and technologies
- Operating system versions and base runtimes of services
- Geographic distribution of Google Cloud services
- Microservices
- Defining a key structure for high write applications using Cloud Storage, Cloud Bigtable, Cloud Spanner, or Cloud SQL
- Session management
- Deploying and securing an API with cloud endpoints
- Loosely coupled applications using asynchronous Cloud Pub/Sub events
- Health checks
- Google-recommended practices and documentation

#### 1.2 Designing secure applications. Considerations include:

- Applicable regulatory requirements and legislation
- Security mechanisms that protect services and resources

- Storing and rotating secrets
- IAM roles for users/groups/service accounts
- HTTPs certificates
- Google-recommended practices and documentation

#### 1.3 Managing application data. Tasks include:

- Defining database schemas for Google-managed databases (e.g., Cloud Datastore, Cloud Spanner, Cloud Bigtable, BigQuery)
- Choosing data storage options based on use case considerations, such as:
  - Cloud Storage signed URLs for user-uploaded content
  - Using Cloud Storage to run a static website
  - Structured vs. unstructured data
  - ACID transactions vs. analytics processing
  - Data volume
  - Frequency of data access in Cloud Storage
- Working with data ingestion systems (e.g., Cloud Pub/Sub, Storage Transfer Service)
- Following Google-recommended practices and documentation

#### 1.4 Re-architecting applications from local services to Google Cloud Platform. Tasks include:

- Using managed services
- Using the strangler pattern for migration
- Google-recommended practices and documentation

## **Section 2: Building and Testing Applications**

2.1 Setting up your development environment. Considerations include:

- Emulating GCP services for local application development
- Creating GCP projects

2.2 Building a continuous integration pipeline. Considerations include:

- Creating a Cloud Source Repository and committing code to it
- Creating container images from code
- Developing unit tests for all code written
- Developing an integration pipeline using services (e.g., Cloud Build, Container Registry) to deploy the application to the target environment (e.g., development, test, staging)
- Reviewing test results of continuous integration pipeline

2.3 Testing. Considerations include:

- Performance testing
- Integration testing
- Load testing

2.4 Writing code. Considerations include:

- Algorithm design
- Modern application patterns
- Efficiency
- Agile methodology

### **Section 3: Deploying applications**

3.1 Implementing appropriate deployment strategies based on the target compute environment (Compute Engine, Google Kubernetes Engine, App Engine). Strategies include:

- Blue/green deployments
- Traffic-splitting deployments
- Rolling deployments
- Canary deployments

3.2 Deploying applications and services on Compute Engine. Tasks include:

- Launching a compute instance using GCP Console and Cloud SDK (gcloud) (e.g., assign disks, availability policy, SSH keys)
- Moving a persistent disk to different VM
- Creating an autoscaled managed instance group using an instance template
- Generating/uploading a custom SSH key for instances
- Configuring a VM for Stackdriver monitoring and logging
- Creating an instance with a startup script that installs software
- Creating custom metadata tags
- Creating a load balancer for Compute Engine instances

3.3 Deploying applications and services on Google Kubernetes Engine. Tasks include:

- Deploying a GKE cluster
- Deploying a containerized application to GKE
- Configuring GKE application monitoring and logging

- Creating a load balancer for GKE instances
- Building a container image using Cloud Build

#### 3.4 Deploying an application to App Engine. Considerations include:

- Scaling configuration
- Versions
- Traffic splitting
- Blue/green deployment

#### 3.5 Deploying a Cloud Function. Types include:

- Cloud Functions that are triggered via an event (e.g., Cloud Pub/Sub events, Cloud Storage object change notification events)
- Cloud Functions that are invoked via HTTP

#### 3.6 Creating data storage resources. Tasks include:

- Creating a Cloud Repository
- Creating a Cloud SQL instance
- Creating composite indexes in Cloud Datastore
- Creating BigQuery datasets
- Planning and deploying Cloud Spanner
- Creating a Cloud Storage bucket
- Creating a Cloud Storage bucket and selecting appropriate storage class
- Creating a Cloud Pub/Sub topic

#### 3.7 Deploying and implementing networking resources. Tasks include:

- Creating an auto mode VPC with subnets

- Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, Tags, Service accounts)
- Setting up a domain using Cloud DNS

### 3.8 Automating resource provisioning with Deployment Manager

### 3.9 Managing Service accounts. Tasks include:

- Creating a service account with a minimum number of scopes required
- Downloading and using a service account private key file

## **Section 4: Integrating Google Cloud Platform Services**

### 4.1 Integrating an application with Data and Storage services. Tasks include:

- Enabling BigQuery and setting permissions on a dataset
- Writing an SQL query to retrieve data from relational databases
- Analyzing data using BigQuery
- Fetching data from various databases
- Enabling Cloud SQL and configuring an instance
- Connecting to a Cloud SQL instance
- Enabling Cloud Spanner and configuring an instance
- Creating an application that uses Cloud Spanner
- Configuring a Cloud Pub/Sub push subscription to call an endpoint
- Connecting to and running a CloudSQL query
- Storing and retrieving objects from Google Storage
- Publishing and consuming from Data Ingestion sources

- Reading and updating an entity in a Cloud Datastore transaction from an application
- Using the CLI tools
- Provisioning and configuring networks

#### 4.2 Integrating an application with Compute services. Tasks include:

- Implementing service discovery in Google Kubernetes Engine, App Engine, and Compute Engine
- Writing an application that publishes/consumes from Cloud Pub/Sub
- Reading instance metadata to obtain application configuration
- Authenticating users by using OAuth2 Web Flow and Identity Aware Proxy
- Using the CLI tools
- Configuring Compute services network settings (e.g., subnet, firewall ingress/egress, public/private IPs)

#### 4.3 Integrating Google Cloud APIs with applications. Tasks include:

- Enabling a GCP API
- Using pre-trained Google ML APIs
- Making API calls with a Cloud Client Library, the REST API, or the APIs Explorer, taking into consideration:
  - batching requests
  - restricting return data
  - paginating results
  - caching results
- Using service accounts to make Google API calls

- Using APIs to read/write to data services (BigQuery, Cloud Spanner)
- Using the Cloud SDK to perform basic tasks

## **Section 5: Managing Application Performance Monitoring**

### 5.1 Installing the logging and monitoring agent

### 5.2 Managing VMs. Tasks include:

- Debugging a custom VM image using the serial port
- Analyzing a failed Compute Engine VM startup
- Sending logs from a VM to Stackdriver

### 5.3 Viewing application performance metrics using Stackdriver. Tasks include:

- Creating a monitoring dashboard
- Viewing syslogs from a VM
- Writing custom metrics and creating metrics from logs
- Graphing metrics
- Using Stackdriver Debugger
- Streaming logs from the GCP Console
- Reviewing stack traces for error analysis
- Setting up log sinks
- Viewing logs in the GCP Console
- Profiling performance of request-response
- Profiling services
- Reviewing application performance using Stackdriver Trace and Stackdriver

### Logging

- Monitoring and profiling a running application

5.4 Diagnosing and resolving application performance issues. Tasks include:

- Setting up time checks and other basic alerts
- Setting up logging and tracing
- Setting up resources monitoring
- Troubleshooting network issues
- Debugging/tracing cloud apps
- Troubleshooting issues with the image/OS
- Using documentation, forums and Google support

### **Locations offered**

#### **On-demand Anytime Anywhere Instructor led training available**

Chennai, Bangalore, Hyderabad, Pune & other major cities in India, US, UK, Middle east, Dubai & Singapore.

### **Offers**

Early bird / Group / Corporate Group / Festive offers available