



Google Cloud Platform (GCP) & Development & Operations Syllabus

GCP Topics Covered

Setting up a cloud solution environment

1.1 Setting up cloud projects and accounts. Activities include:

- Creating a resource hierarchy
- Applying organizational policies to the resource hierarchy
- Granting members IAM roles within a project
- Managing users and groups in Cloud Identity (manually and automated)
- Enabling APIs within projects
- Provisioning and setting up products in Google Cloud's operations suite

1.2 Managing billing configuration. Activities include:

- Creating one or more billing accounts
- Linking projects to a billing account

- Establishing billing budgets and alerts
- Setting up billing exports

1.3 Installing and configuring the command line interface (CLI), specifically the Cloud SDK (e.g., setting the default project)

Planning and configuring a cloud solution (~17.5% of the exam)

2.1 Planning and estimating Google Cloud product use using the Pricing Calculator

2.2 Planning and configuring compute resources. Considerations include:

- Selecting appropriate compute choices for a given workload (e.g., Compute Engine, Google Kubernetes Engine, Cloud Run, Cloud Functions)
- Using preemptible VMs and custom machine types as appropriate

2.3 Planning and configuring data storage options. Considerations include:

- Product choice (e.g., Cloud SQL, BigQuery, Firestore, Spanner, Bigtable)
- Choosing storage options (e.g., Zonal persistent disk, Regional balanced persistent disk, Standard, Nearline, Coldline, Archive)

2.4 Planning and configuring network resources. Tasks include:

- Differentiating load balancing options
- Identifying resource locations in a network for availability
- Configuring Cloud DNS

Deploying and implementing a cloud solution

3.1 Deploying and implementing Compute Engine resources. Tasks include:

- Launching a compute instance using the Google Cloud console and Cloud SDK (gcloud) (e.g., assign disks, availability policy, SSH keys)
- Creating an autoscaled managed instance group using an instance template
- Generating/uploading a custom SSH key for instances
- Installing and configuring the Cloud Monitoring and Logging Agent
- Assessing compute quotas and requesting increases

3.2 Deploying and implementing Google Kubernetes Engine resources. Tasks include:

- Installing and configuring the command line interface (CLI) for Kubernetes (kubectl)
- Deploying a Google Kubernetes Engine cluster with different configurations including AutoPilot, regional clusters, private clusters, etc.
- Deploying a containerized application to Google Kubernetes Engine
- Configuring Google Kubernetes Engine monitoring and logging

3.3 Deploying and implementing Cloud Run and Cloud Functions resources. Tasks include, where applicable:

- Deploying an application and updating scaling configuration, versions, and traffic splitting
- Deploying an application that receives Google Cloud events (e.g., Pub/Sub events, Cloud Storage object change notification events)

3.4 Deploying and implementing data solutions. Tasks include:

- Initializing data systems with products (e.g., Cloud SQL, Firestore, BigQuery, Spanner, Pub/Sub, Bigtable, Dataproc, Dataflow, Cloud Storage)
- Loading data (e.g., command line upload, API transfer, import/export, load data from Cloud Storage, streaming data to Pub/Sub)

3.5 Deploying and implementing networking resources. Tasks include:

- Creating a VPC with subnets (e.g., custom-mode VPC, shared VPC)
- Launching a Compute Engine instance with custom network configuration (e.g., internal-only IP address, Google private access, static external and private IP address, network tags)
- Creating ingress and egress firewall rules for a VPC (e.g., IP subnets, network tags, service accounts)
- Creating a VPN between a Google VPC and an external network using Cloud VPN
- Creating a load balancer to distribute application network traffic to an application (e.g., Global HTTP(S) load balancer, Global SSL Proxy load balancer, Global TCP Proxy load balancer, regional network load balancer, regional internal load balancer)

3.6 Deploying a solution using Cloud Marketplace. Tasks include:

- Browsing the Cloud Marketplace catalog and viewing solution details
- Deploying a Cloud Marketplace solution

3.7 Implementing resources via infrastructure as code. Tasks include:

- Building infrastructure via Cloud Foundation Toolkit templates and implementing best practices
- Installing and configuring Config Connector in Google Kubernetes Engine to create, update, delete, and secure resources

Ensuring successful operation of a cloud solution

4.1 Managing Compute Engine resources. Tasks include:

- Managing a single VM instance (e.g., start, stop, edit configuration, or delete an instance)
- Remotely connecting to the instance
- Attaching a GPU to a new instance and installing necessary dependencies
- Viewing current running VM inventory (instance IDs, details)
- Working with snapshots (e.g., create a snapshot from a VM, view snapshots, delete a snapshot)
- Working with images (e.g., create an image from a VM or a snapshot, view images, delete an image)
- Working with instance groups (e.g., set autoscaling parameters, assign instance template, create an instance template, remove instance group)
- Working with management interfaces (e.g., Google Cloud console, Cloud Shell, Cloud SDK)

4.2 Managing Google Kubernetes Engine resources. Tasks include:

- Viewing current running cluster inventory (nodes, pods, services)
- Browsing Docker images and viewing their details in the Artifact Registry
- Working with node pools (e.g., add, edit, or remove a node pool)
- Working with pods (e.g., add, edit, or remove pods)
- Working with services (e.g., add, edit, or remove a service)
- Working with stateful applications (e.g. persistent volumes, stateful sets)

- Managing Horizontal and Vertical autoscaling configurations
- Working with management interfaces (e.g., Google Cloud console, Cloud Shell, Cloud SDK, kubectl)

4.3 Managing Cloud Run resources. Tasks include:

- Adjusting application traffic-splitting parameters
- Setting scaling parameters for autoscaling instances
- Determining whether to run Cloud Run (fully managed) or Cloud Run for Anthos

4.4 Managing storage and database solutions. Tasks include:

- Managing and securing objects in and between Cloud Storage buckets
- Setting object life cycle management policies for Cloud Storage buckets
- Executing queries to retrieve data from data instances (e.g., Cloud SQL, BigQuery, Spanner, Datastore, Bigtable)
- Estimating costs of data storage resources
- Backing up and restoring database instances (e.g., Cloud SQL, Datastore)
- Reviewing job status in Dataproc, Dataflow, or BigQuery

4.5 Managing networking resources. Tasks include:

- Adding a subnet to an existing VPC
- Expanding a subnet to have more IP addresses
- Reserving static external or internal IP addresses
- Working with CloudDNS, CloudNAT, Load Balancers and firewall rules

4.6 Monitoring and logging. Tasks include:

- Creating Cloud Monitoring alerts based on resource metrics
- Creating and ingesting Cloud Monitoring custom metrics (e.g., from applications or logs)
- Configuring log sinks to export logs to external systems (e.g., on-premises or BigQuery)
- Configuring log routers
- Viewing and filtering logs in Cloud Logging
- Viewing specific log message details in Cloud Logging
- Using cloud diagnostics to research an application issue (e.g., viewing Cloud Trace data, using Cloud Debug to view an application point-in-time)
- Viewing Google Cloud status

Configuring access and security

5.1 Managing Identity and Access Management (IAM). Tasks include:

- Viewing IAM policies
- Creating IAM policies
- Managing the various role types and defining custom IAM roles (e.g., primitive, predefined and custom)

5.2 Managing service accounts. Tasks include:

- Creating service accounts
- Using service accounts in IAM policies with minimum permissions

- Assigning service accounts to resources
- Managing IAM of a service account
- Managing service account impersonation
- Creating and managing short-lived service account credentials

5.3 Viewing audit logs

DevOps Topic Covered

Learning Objective

- ❖ **Understanding DevOps**

Topic Covered

- What is DevOps?
- Evolution of Software Methodologies
 - Waterfall
 - Agile
 - Disadvantages of traditional SDLC
- Why DevOps?
- Dev Challenges v/s DevOps Solution
- Ops Challenges v/s DevOps Solution
- Stages Of DevOps Lifecycle
 - Continuous Development
 - Continuous Testing
 - Continuous Integration
 - Continuous Deployment
 - Continuous Monitoring
- The Various DevOps Tools Introduction
- Roles and Responsibilities of a DevOps Engineer
- How DevOps fits in the whole Software Development Lifecycle

Physical and Virtual Computing Environment

Learning Objective

- ❖ **Computing Environments and Getting Started with Virtualization**

Topic Covered

- Computing Environments and Operating Systems
 - Personal Computing, Server- Client
- Understanding Virtualization

- Understanding Hypervisor
 - Hosted
 - Baremetal

Hands-On

- Setting up a Virtual Machine with Linux OS

Linux for Devops

Learning Objective

- ❖ **Understanding Linux OS and Basic Shell Commands**

Topic Covered

- What is Linux and Open Source
- Linux Distributions
- Understanding Local User in Linux
- Understanding Linux Hierarchy
- Understanding Shell
- Understanding Absolute and Relative Path
- Working with Text Editors
- Creating Files and Directory Using CLI
- Understanding Linux Command and Using Options for Linux Commands
- Understanding Globbing, Pipe, Tee
- Understanding Variables, Operators

Hands-On

- Executing Basic Shell Commands - cd, ls, vim, nano, touch, mkdir
- File and Directory Management - cp, mv, rm
- Commands with Pattern Matching
- Writing Simple Shell Scripts

Programming with Python

Learning Objective

- ❖ **Getting Started with Python and Writing Python Programs**

Topic Covered

- Introduction to Python: What is Python and why to learn Python as a DevOps engineer?
- Installation and Setup Local Development Environment
- Write Simple Python program
- Python IDE vs simple File Editor
- Strings and Number Data Types
- Variables
- Encapsulate Logic with Functions
- Accepting User Input
- Conditionals (if / else) and Boolean Data Type
- Error Handling with Try / Except
- While Loops
- Lists and For Loops
- Comments in Python
- Sets
- Built-In Functions
- Dictionary Data Type
- Modularize your project with Modules
- Packages, PyPI and Pip
- Object Oriented Programming: Classes and Objects

Hands-On

- Installation and Setup Local Development Environment
- Writing Python Scripts
- Project: Countdown App
- Project: Automation with Python (Working with Spreadsheets)
- Project: API Request to GitLab

VCS with Git

Learning Objective

- ❖ Understanding Git & GitHub (Managing Source Code and What is Version Control System(VCS)?
- ❖ Understanding AWS Code Commit

Topic Covered

- Why VCS?
- VCS tools
- Distributed VCS
- What is Git & Why Git?
- Features Of Git
- Git Workflow
- Git Configurations
- Creating Git Repository
- Syncing Repositories
- Adding Origin
- Pushing changes
- Pulling changes
- Clone operation
- Concepts of Branches
- Merge Requests
- Deleting Branches
- Resolving Merge Conflicts
- Git Ignore
- Git Stash
- Merging Branches

Hand-on Lab:

- Launch EC2 Instance (Windows) and Configure Git
- Launch EC2 Instance (Linux) and Configure Git

- ❑ Configuring all Git Operations (Creating Local and Setting up Remote Repository in GitHub and Code Commit), push, pull, clone, creating branches, merge.

Build & Package Manager Tools

Learning Objective

- ❖ What are Build Tools and Package Managers?

Topic Covered

- ❑ How to build an artifact?
- ❑ How to run the application artifact?
- ❑ How to publish the application artifact to the artifact repository?
- ❑ Build Tools for Java (gradle and maven examples)
- ❑ Dependency Management in Software Development
- ❑ Package Manager in JavaScript applications - Build and run applications in JS
- ❑ Build Tools
- ❑ Why Build Tools are relevant for DevOps Engineers?

Artifact Repository Manager with Nexus

Learning Objective

- ❖ Understanding Artifact Repository Manager, Types

Topic Covered

- ❑ What is an Artifact Repository Manager?
- ❑ Install and run Nexus on Cloud Server
- ❑ Different Repository Types (proxy, hosted, etc.) explained
- ❑ Different Repository Formats (maven, docker, npm, etc.) explained
- ❑ Upload Jar File to Nexus (maven and gradle projects)
- ❑ Nexus API and Repository URLs
- ❑ Blob stores

- Browsing Components - Components vs Assets
- Cleanup Policies
- Scheduled Tasks

Continuous Integration with Jenkins

Learning Objective

- ❖ Understanding Integration with Jenkins

Topic Covered

- Challenges before Continuous Integration
- What is Continuous Integration?
- Benefits of Continuous Integration
- Tools of Continuous Integration
- Introduction to Jenkins
- Jenkins Plugins
- Build Setup in Jenkins
- Jenkins Pipeline (Use Cases)
- Create a simple Pipeline Job
- Full Jenkinsfile Syntax Demo
- Create a full Pipeline Job
- Build Java App
- Build Docker Image
- Push to Private DockerHub
- Create a Multi-Branch Pipeline Job
- Credentials in Jenkins
- Jenkins Shared Library
- WebHooks - Trigger Jenkins Jobs automatically
- Versioning Application in Continuous Deployment
- Concepts of Versioning in Software Development
- Increment Application version from Jenkins Pipeline
- Set new Docker Image version from Jenkins Pipeline
- Commit Version Bump from Jenkins Pipeline

Hand-on Lab:

- Launch EC2 Instance (Linux) and Install Jenkins
- Creating a simple freestyle job
- Configure Git Repository and Build a java application
- Build docker images and push it to docker hub
- Create a simple and multi pipeline

Containerization

Learning Objective

- ❖ **Understanding Traditional(Physical and Virtual) Application Deployment Methods) and Containerization**
- ❖ **Advantages**
- ❖ **Understanding Docker and Components**
- ❖ **Managing Docker in a Standalone Instance**

Topic Covered

- Virtualization vs Containerization
- What are Containers and Advantages of Containers
- Architecture of Docker Container
- Components of Docker
 - Images
 - Registries (Docker Hub, Elastic Container Registry)
- Managing Docker Service
- Running a Container(Attached/Detached), Logging in to Container
- Starting / Stopping / Restarting Containers
- Container Networking
 - Bridge
 - Host
 - Overlay
- Managing Storage for Containers
- Understanding Docker File

- Docker Hub - Pushing Images to Repository

Hand-on Lab:

- Launch EC2 Instance (Windows and Linux) and Configure Docker Engine
- Run a Simple Standalone Webapp Container
- Run an Ubuntu Container to Check Connectivity between Containers
- Creating Volume for Containers and Mounting it Persistently
- Creating a Custom Container Image from Another Container / Docker File

Containerization Orchestration

Learning Objective

- ❖ **Understanding What is Container Orchestration**
- ❖ **Kubernetes**
- ❖ **ECS/EKS**

Topic Covered

- Introduction to Kubernetes
- Understand the Main Kubernetes Components
- Node, Pod, Service, Ingress, ConfigMap, Secret, Volume, Deployment, StatefulSet
- Kubernetes Architecture
- Minikube and Kubectl - Local Setup
- Main Kubectl Commands - K8s CLI
- Create and Debug Pod in a Miniclustter
- Kubernetes YAML Configuration File
- Create and Configure Deployment and Service Components
- Organizing your components with K8s Namespaces
- Kubernetes Service Types
- Making your App accessible from outside with Kubernetes Ingress
- Persisting Data in Kubernetes with Volumes
- Persistent Volume
- Persistent Volume Claim

- Storage Class
- ConfigMap and Secret Kubernetes Volume Types
- Deploying Stateful Apps with StatefulSet
- Deploying Kubernetes cluster on a Managed Kubernetes Service (K8s on Cloud)
- Helm - Package Manager of Kubernetes
- Creating a ECS Cluster
- Creating a EKS Cluster

Infrastructure as Code with Terraform & Ansible

Learning Objective

- ❖ **Terraform and it's components**

Topic Covered

- What is Terraform? How it works
- Architecture
- Providers
- Resources & Data Sources
- Variables & Output Values
- Environment variables in Terraform
- Terraform commands
- Terraform State
- Provisioners
- Modules
- Remote State
- Terraform & AWS

Hand-on Lab:

- Create Security Group and Provision EC2 windows or Linux Instance using Terraform
- Configure Terraform in Jenkins
- Automate provisioning EC2 instance from Jenkins pipeline and deploy the application with docker-compose

Learning Objective

❖ Automation with Ansible

Topic Covered

- What is Ansible, Uses and How Ansible Works.
- Architecture
 - Control Node
 - Managed Node
 - Inventory
 - Module
 - Play and Playbook
- Managing Static Inventory
- Creating Ansible Project Directory and Configurations
- Understanding Ansible Ad-hoc Commands
- Privilege Escalation Configuration
- Understanding YAML and Writing Simple Ansible Playbook
- Using Variables in Ansible Playbook
- Using Loop
- Using Conditions
- Ansible Roles from Ansible Galaxy

Hand-on Lab:

- Launch an EC2 Linux Instance and Configure Ansible Engine
- Setting Up Linux Managed Nodes
- Writing a Playbook for Deploying Web Application
- Deploying Applications from Ansible Galaxy
- Project: Ansible & Terraform
- Project: Run Docker applications
- Project: Run Ansible from Jenkins Pipeline