**Home Work Task-2025-04-03**

**1.Difference between a VPC Network and a Subnet:**

VPC Network

[**Virtual Private Cloud (VPC) networks**](https://cloud.google.com/vpc/docs/vpc) are global resources. Each VPC network consists of one or more IP address ranges called *subnets*.

It acts as a container for resources in the cloud.

It allows you to define and control your cloud environment's IP address range, routing, and security settings with smaller segment called subnets

Subnet:

A **Subnet** is a smaller segment within a VPC network, where you divide the larger VPC into isolated IP address ranges to organize and manage resources more efficiently.

Each subnet typically corresponds to a specific availability zone or region.

A network must have at least one subnet before we can use it

* **Isolation & Security**: A VPC network allows users to isolate their resources within a virtual network, separate from others.
* **Routing & Traffic Control**: VPC networks manage routing rules and control the flow of traffic between different resources and subnets. You can also use VPC Peering to connect multiple VPC networks.
* **Access & Permissions**: With IAM and firewall rules, a VPC network helps control who can access the resources and how they can interact with them.

### **2. Role in Google Cloud**

* **VPC Network in Google Cloud**:
  + **Isolation & Security**: A VPC network allows users to isolate their resources within a virtual network, separate from others.
  + **Routing & Traffic Control**: VPC networks manage routing rules and control the flow of traffic between different resources and subnets. You can also use VPC Peering to connect multiple VPC networks.
  + **Access & Permissions**: With IAM and firewall rules, a VPC network helps control who can access the resources and how they can interact with them.
* **Subnet in Google Cloud**:
  + **Resource Organization**: Subnets help organize and segment resources in different regions or availability zones, allowing efficient management of resources across multiple locations.
  + **IP Allocation**: Subnets define specific ranges of IP addresses that can be used by resources like virtual machines, load balancers, and databases within the subnet.
  + **Traffic Segmentation**: Each subnet can have different firewall rules and routing configurations, allowing for more granular control over traffic within the network.

***2.Role of IAM in GCP with an Example:***

The basic roles in IAM are Admin (roles/admin), Writer (roles/writer), and Reader (roles/reader). IAM also has three legacy basic roles that existed prior to the introduction of IAM: Owner (roles/owner), Editor (roles/editor), and Viewer (roles/viewer).

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Role | Explanation | Example |
| 1. | Reader | Permissions for read-only actions that don't affect state, such as viewing (but not modifying) existing resources or data. | 4868 assigned permissions accessapproval.requests.get  accessapproval.requests.list |
| 2. | Writer | All of the permissions in the Reader role, *plus* permissions for actions that modify state, such as changing existing resources. | 9662 assigned permissions accesscontextmanager.accessLevels.create  accesscontextmanager.accessLevels.delete |
| 3. | Admin (roles/admin) | All of the permissions in the Writer role, *plus* permissions for actions like the following:   * Completing sensitive tasks, like managing tag bindings for Compute Engine resources * Managing roles and permissions for a project and all resources within the project * Setting up billing for a project | 10994 assigned permissions  accessapproval.requests.approve  accessapproval.requests.dismiss |

### **Purpose and Importance of IAM in Cloud Security**

* **Centralized Access Control**: IAM provides a centralized way to manage access to Google Cloud resources, offering a unified system for controlling permissions.
* **Security**: By controlling who can access your cloud resources and what they can do, IAM enhances the security of your environment. Properly configured IAM helps minimize the risks of unauthorized access, data breaches, and potential misuse of cloud resources.
* **Granular Permissions**: IAM enables fine-grained control, allowing you to assign different levels of access to different users or services, from full administrative privileges to read-only access.
* **Audit and Compliance**: IAM logs and tracks activities, which helps organizations maintain a history of who accessed what resources, ensuring compliance with regulatory standards.

***3.Differences between IAM Policies and Service Accounts:***

|  |  |  |
| --- | --- | --- |
| **Aspect** | **IAM Policies** | **Service Accounts** |
| **Purpose** | Define access control and permissions for users, groups, or service accounts to GCP resources. | Represent a non-human entity (like an application or service) to access GCP resources. |
| **Scope** | Applied to users, groups, and service accounts to grant them specific roles and permissions. | Used by applications or services to interact with GCP resources programmatically. |
| **Association** | Associated with identities (users, groups, etc.) to define what they can do on resources. | Associated with virtual machines or applications to authenticate and authorize API access. |
| **Types of Entities Involved** | Users, groups, or service accounts (who are granted roles). | Service accounts (that are linked to resources like VMs, or APIs). |
| **Usage Example** | Granting a user the "Editor" role for a GCP project. | A VM uses a service account to authenticate and access Cloud Storage to read data. |
| **Access Control** | IAM policies grant permissions to users and service accounts to perform actions on resources. | Service accounts authenticate applications or VMs, enabling them to access specific resources based on the permissions assigned. |

In summary, **IAM Policies** govern what actions identities (users or service accounts) can perform on GCP resources, while **Service Accounts** represent machines or applications that need programmatic access to those resources.

### **2. Explanation of Their Functions and How They Are Applied in GCP**

* **IAM Policies**:
  + **Function**: IAM policies are used to define access control for resources in Google Cloud. These policies specify **who (identity)** can perform **what actions (permissions)** on **which resources**. IAM policies can be assigned to users, groups, or service accounts through roles (predefined or custom).
* **Application**: IAM policies are applied at various levels such as the **project**, **folder**, or **organization** level.
* **Service Accounts**:
  + **Function**: Service accounts provide **identity** for **applications** or **automated processes** that need to interact with Google Cloud resources. They authenticate applications to use specific cloud services and resources. **Application**: Service accounts are created for specific services or applications and assigned **roles** that define their permissions.

### Differences

* **IAM Policies**:
  + Govern the **permissions** granted to users and services within the Google Cloud environment.
  + IAM policies define access at a **higher level**, allowing administrators to control who can perform actions on resources across the cloud environment.
  + Examples include giving a user access to **view** or **edit** resources in a project.
* **Service Accounts**:
  + Are used to represent **automated services** or **applications**, providing them with a unique identity to authenticate and interact with Google Cloud resources.
  + Service accounts are assigned **roles** that define the level of access for the service, allowing for secure, programmatic access to resources.
  + Examples include allowing a service account to authenticate and access a specific **Google Cloud Storage bucket**.

***4.Differences Between Various Storage Classes in GCP:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Storage Class** | **Description** | **Use Case** | **Cost (Lowest to Highest)** | **Availability** | **Durability** |
| **Standard** | Designed for frequently accessed data. | Hosting websites, content delivery, active data, etc. | Medium | High | 99.999999999% |
| **Nearline** | For data that is accessed less than once a month. | Backup and disaster recovery, infrequently accessed data. | Low | High | 99.999999999% |
| **Coldline** | For data that is accessed less than once a year. | Long-term storage, archiving, backup, and disaster recovery. | Very low | High | 99.999999999% |
| **Archive** | For data that is rarely accessed, typically for long-term archival. | Archival storage, regulatory data retention, historical data. |  |  |  |

* **Standard**: Designed for frequently accessed data, offering high performance and lower latency. This is ideal for active applications and real-time data processing.
* **Nearline**: Best for infrequently accessed data (less than once per month), such as backups or disaster recovery. It offers a good balance between cost and performance for occasional data access.
* **Coldline**: Aimed at long-term storage for data that is rarely accessed (less than once a year). It is ideal for archiving and compliance storage, where low cost is more important than quick access.
* **Archive**: The most cost-efficient storage class for **rarely accessed, long-term archival** data. This is designed for data that you need to keep for extended periods (e.g., backups, logs, regulatory data), but doesn't need to be accessed often.