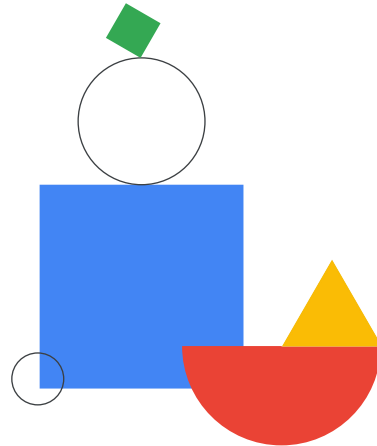Google Cloud

Security

Google has been operating securely in the cloud for 20 years. There is a strong belief that security empowers innovation. The approach of the cloud architect should be that security should be put first; everything else will follow from this.

Learning objectives

01 Design secure systems using best practices like separation of concerns, principle of least privilege, and regular audits.

02 Leverage Google's Security Command Center to help identify vulnerabilities.

03 Simplify cloud governance using organization policies and folders.

04 Authenticate and authorize users with IAM roles, Identity-Aware Proxy, and Identity Platform.

Google Cloud

This module introduces several aspects of Google Cloud security, from a high-level architectural principles view to implementation details with IAM for authentication and authorization…

Learning objectives

05 Manage the access and authorization of resources by machines and processes using service accounts.

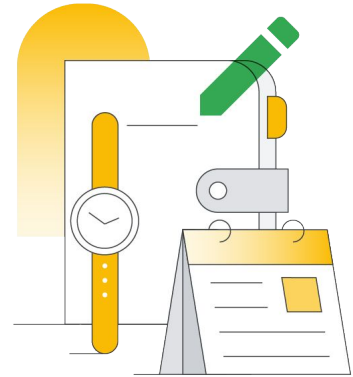**06** Secure networks with private IPs, firewalls, and Google Cloud private access.

07 Mitigate DDoS attacks by leveraging Cloud DNS and Google Cloud Armor.

Google Cloud

…network security with firewalls, and services such as Google Cloud Armor and Cloud DNS for DDoS protection.

Agenda

Google Cloud

01

Security Concepts

Google Cloud security is a shared responsibility between you and Google

Transparency

- The client is responsible for certain actions, and Google is responsible for others.
- Google Cloud provides the tools and access to monitor your service.
- Google Cloud provides the controls and features needed to leverage platform security.

Separation of duties

- What is provided by the platform?
- What are you responsible for?

Google Cloud

When you build an application with on-premises infrastructure, you're responsible for:

- The physical security of the hardware and the premises in which it is housed
- The encryption of the data on disk
- The integrity of your network
- The security of the content stored in your applications

When you move an application to Google Cloud, Google handles many of the lower layers of the overall security stack. Because of its scale, Google can deliver a higher level of security at these layers than most of its customers could afford to do on their own.

The upper layers of the security stack remain the customer's responsibility. Google provides tools, such as Identity and Access Management (IAM) to help customers implement the policies they choose at these layers.

We will look at this and the security layers in more detail in the next few slides.

Security is implemented in layers

- Google Cloud provides tools that, when properly configured, enable a secure environment.
- You can also integrate third-party tools for enhanced security.
- There are tools for monitoring and auditing your networks and resources.

	Layer
	Usage
	Operations
	Deployment
	Application
	Storage
	OS + IPC
	Boot
	Hardware

Google Cloud
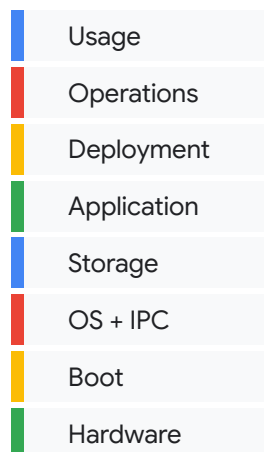
The Google philosophy is built on trust and transparency combined with abstraction and automation.

Secure abstractions mean you don't need to be security experts but can quickly develop applications and the infrastructure helps with security.

At the base is custom Google-built hardware containing just the hardware components required to prevent vulnerabilities.The servers use a verified boot loading system based on cryptographic signatures to ensure that the loaded software is verified. The operating system is a stripped-down and hardened version of Linux designed with the sole purpose of running Google services. The system continuously monitors systems for binary modifications, and if a change is found from the standard Google image, the system is returned automatically to its official state. Stored data is encrypted, and access can be controlled using IAM. Logging also provides an audit trail of when data was accessed.

At the application level, Google's dedicated security team actively scans for security threats using custom tools and commercial software. They perform penetration tests, QA measures, and software security reviews. For deployment, the underlying infrastructure is designed to be multi-tenant and so makes no assumptions on trust between services running on the infrastructure. Each deployed service has an associated service account and is provided with cryptographic credentials for providing identity when making calls to other services. These identities help clients ensure that they are talking to the intended services and help services limit access to

intended clients. The source code for Google services is stored in a central repository and requires review from at least one engineer other than the author.

For operational security and usage, the aim is to protect against threats to infrastructure and data from insiders and external actors. Developers are provided with safe libraries which prevent certain classes of security bugs from entering the system. These eliminate, for example, XSS vulnerabilities. Internally, mandatory use of U2F-compatible security keys for employee accounts is enforced. Heavy monitoring of devices used to operate the infrastructure is performed. Various monitoring points which are host-based or network-based and signals from infrastructure services are fed into data processing pipelines with rules and machine intelligence to provide security warnings of possible incidents. Incident response teams triage, investigate, and respond to potential incidents 24 hours a day, 365 days a year.

Principle of least privilege

- Users should only be able to do the tasks that are required by their jobs.

- This should also apply to machine instances and run-time processes.

- Use IAM to enforce this principle.

- Identify users with their login.

- Identify machines and code using service accounts.

- Assign IAM roles to users and service accounts to restrict what they can do.

Google Cloud

Least privilege is a practice of granting a user only the minimal set of permissions required to perform a duty. IAM roles and permissions are fine-grained and support the practice of least privilege. The predefined IAM roles are designed to fit the needs of of the most common roles. It is highly likely that there is a predefined role for the significant majority of requirements. In those cases where there is not, it is possible to create a custom role and assign that role the privileges required.

The IAM model has three main parts:

- **Members**: A member can be a Google account, a service account, a Google group, or a Google Workspace or Cloud Identity domain.
- **Roles**: A role is a collection of permissions that determine what operations on a resource are allowed. When a member is granted a role, they are assigned all the permissions that role has.
- **Policies**: A policy binds one or more members to a role. The policy is attached to a resource.

Separation of duties

Separation of duties means:

- No one person can change or delete data without being detected.

- No one person can steal sensitive data.

- No one person is in charge of designing, implementing, and reporting on sensitive systems.

For example, the people who write the code shouldn't deploy the code, and those who deploy the code shouldn't be able to change it.

- Use multiple projects to separate duties.

- Different people can be given different rights in different projects.

- Use folders to help organize projects.

Google Cloud

Separation of duties is a security design principle (as is least privilege, discussed on the previous slide).

Separation of duties has two primary objectives:

1. Prevention of conflict of interest
2. The detection of control failures; e.g., security breaches, information theft

Considering conflict of interest, a simple example for a business is that no individual would have the permission to create an invoice and then pay an invoice. Such privilege would open the door for fraudulent actions. Equally, it could be more technical, with the developer who writes code not able to deploy that code to production, at least not without a review by somebody else.

The person responsible for designing and implementing security must not be the same person who is responsible for testing security, conducting security audits, or monitoring and reporting on security.

Regularly audit the Google Cloud logs to discover attacks

All Google Cloud services write to audit logs:

- Admin logs
- Data access logs
- VPC Flow logs
- Firewall logs
- System logs

Cloud Audit Logs maintains the following logs for each project, folder, and organization:

- Admin activity logs retained for 400 days
- Data access logs retained for 30 days
- System event logs retained for 400 days

The services write entries to these logs that answer the "who did something, where, and when" on Google Cloud resources.

Admin activity logs record actions that modify the configuration of resources or metadata. These logs are always written and cannot be disabled. There is no charge for these logs.

Data access logs record reading of configuration or metadata of resources and requests to modify or read user-provided resource data. No logs are recorded for publicly shared resources or any that can be accessed without logging in to Google Cloud. These logs are disabled by default and must be explicitly enabled. The main reason for this is that the logs can become large. If these are enabled, the project may be charged according to the quotas and limits detailed here: https://cloud.google.com/logging/quotas.

System events logs are for actions that modify the configuration of resources. These are generated by Google systems, and cannot be disabled and there is no charge for

these logs.

For a full list of the services that provide audit logs, see
https://cloud.google.com/logging/docs/audit/services.

Given the retention period of the logs, it is often necessary to export these to hold the information for extended periods or to use analysis tools on the data. There are three ways to export logs:

1. To JSON files in Cloud Storage
2. To BigQuery
3. To Pub/Sub topic to be consumed by a store such as Splunk or Elastic Search

For more details on exporting data, see
https://cloud.google.com/solutions/design-patterns-for-exporting-stackdriver-logging#logging_export_scenarios.

In addition to the above audit logs, from an operational perspective it can be useful to collect VPC Flow Logs and Firewall Logs. VPC Flow logs can capture telemetry from a variety of sources, such as internal VPC traffic between VPC and on-premises deployments. VPC Flow Logs must be explicitly enabled on a subnet basis. Firewall logging allows the audit and verification of firewall rules to ensure that they are working as intended. The logging must be enabled for each rule. For both VPC Flow Logs and Firewall logs, the data can be exported to Cloud Logging or BigQuery.

Google Cloud meets many third-party and government compliance standards worldwide

- Google Cloud has been certified as secure, but that does not mean that your application is certified.

- Don't worry about getting Google Cloud tools and services certified; only worry about what you build on top of Google Cloud.

ISO/IEC 27001

HIPAA

FedRAMP

SOC 1

Google Cloud

Moving to the cloud often requires maintaining compliance with regulatory requirements or guidelines. Google can help support these compliance requirements.

Google products undergo regular independent verification of security privacy and compliance controls. To help customers with compliance and reporting, Google shares information and best practices, and provides access to documentation. The logos on the slide show just a sample of the compliance offerings of Google Cloud. For the full set, see https://cloud.google.com/security/compliance/offerings/#/.

It is important to note, as the slide above highlights, that when deploying your services on Google Cloud, you have to ensure that they follow the compliance standards. Guidelines are provided by Google for some standards. For example, for HIPAA projects: https://cloud.google.com/solutions/setting-up-a-hipaa-aligned-project.

Security Command Center provides access to organizational and project security configuration
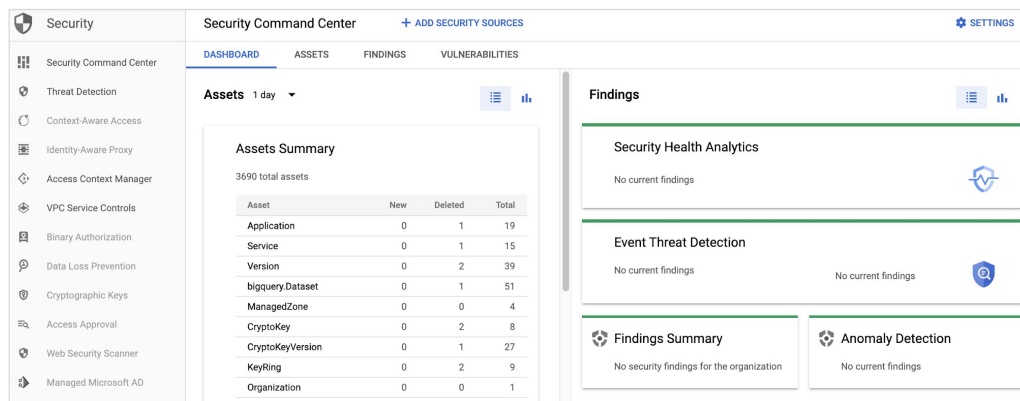
The Security Command Center accessible from the Cloud Console provides access to organizational and project security configuration. It provides visibility into the resources used and their security state. The Command Center should make it easier to prevent and also detect and respond to threats. Built-in features detect suspicious activity in Cloud security logs and can detect compromised virtual machines. For possible threats, a set of actionable recommendations is provided.

Some of the features provided include:

- Asset discovery and inventory
- Sensitive data discovery
- Web application vulnerability detection
- Access control monitoring
- Real time notifications
- Audit logs
- Assessment of misconfigurations

For more details, see
https://cloud.google.com/security-command-center/docs/concepts-overview

02

Securing People

To grant people access to your projects, add them as members and assign them one or more roles

- Members are identified by their login.
- Add members to groups for easier management.

- Roles are simply a list of permissions.
- Use the Console to easily see what permissions are granted to roles.

	Type	Title	Used in	Status	
☐	⊙	BigQuery Admin	BigQuery	Enabled	⋮
☐	⊙	BigQuery Connection Admin	BigQuery	Enabled	⋮
☐	⊙	BigQuery Connection User	BigQuery	Enabled	⋮
☐	⊙	BigQuery Data Editor	BigQuery	Enabled	⋮
☐	⊙	BigQuery Data Owner	BigQuery	Enabled	⋮
☐	⊙	BigQuery Data Viewer	BigQuery	Enabled	⋮
☐	⊙	BigQuery Job User	BigQuery	Enabled	⋮
☐	⊙	BigQuery Metadata Viewer	BigQuery	Enabled	⋮
☐	⊙	BigQuery Read Session User	BigQuery	Enabled	⋮
☐	⊙	BigQuery User	BigQuery	Enabled	⋮
☐	⊙	Cloud Asset Owner	Cloud Asset	Enabled	⋮

BigQuery User   + EDIT ROLE   CREATE FROM ROLE

ID   roles/bigquery.user
Role launch stage   General Availability

Description
Access to run queries and create datasets

15 assigned permissions
bigquery.config.get
bigquery.datasets.create
bigquery.datasets.get
bigquery.datasets.getIamPolicy
bigquery.jobs.create
bigquery.jobs.list
bigquery.models.list
bigquery.readsessions.create
bigquery.routines.list
bigquery.savedqueries.get
bigquery.savedqueries.list
bigquery.tables.list
bigquery.transfers.get
resourcemanager.projects.get
resourcemanager.projects.list

Google Cloud

Google Cloud resources are organized into a hierarchy with the organization as the root node; folders are children of an organization, projects are children of a folder, and resources are children of projects.

Granting permissions requires user identity, groups, permissions, roles, and policies, which are explained below.

Users identity can be provided by:
Google account: using an email address linked to a Google account.
Cloud Identity: for users who do not have a Google account or Google Workspace account.

Groups can be created; groups are sets of Google accounts and have an associated email address. These are useful for assigning permissions to a group of users. When a user is added to a group, they acquire the permissions of that group.

Permissions are a grant to perform some action on a resource; e.g., resourcemanager.projects.list will provide the ability to list instances for a project.

Roles are used to create sets of permissions. Roles are then assigned to identities. There are three types of roles in IAM:
- Predefined roles giving fine-grained access to resources
- Custom roles that are used when predefined roles do not meet your requirements

- Basic roles: these are legacy and should be avoided

Policies are the way roles are assigned to users. A policy has a set of users and associated roles. Each role/users combination is known as a binding. Policies can be set anywhere in the resource hierarchy, so organization or project or individual resource.

Recommendations on the allocation of privileges are discussed on the next slide.

Use Identity and Access Management policies to secure environments and manage resources

Grant roles to Google groups rather than individuals

- Groups can be more granular than job roles.
- Use multiple groups for better control (such as *view only*).

Roles

- Prefer pre-defined roles over custom roles.
- Grant roles at the smallest scope needed (least privilege).
- Limit use of "owner" and "editor" roles.
- Consider hierarchy inheritance when assigning roles.

Policy Inheritance

Organization

Folders

Projects
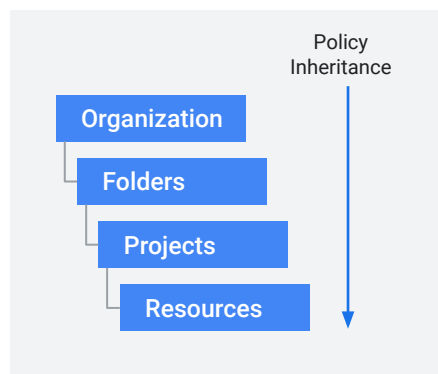
Resources

You can set Identity and Access Management (IAM) policies at different levels of the resource hierarchy. Resources inherit the policies of the parent resource. The effective policy for a resource is the union of the policy set at that resource and the policy inherited from its parent. Some of the best practices for applying IAM are listed below:

It is best to select policies at the organization and project level. As new resources are added, they will automatically inherit the policies of their parent. This makes maintenance of policies easier and helps keep consistency. If adding a policy on a child resource, make sure you are aware of the access granted by the parent and the effect of inheritance.

The principle of least privilege should always be applied, giving the minimal amount of access to roles and making sure to minimize the use of owner and editor roles.

The predefined roles have been designed to cover all use cases for resources. The need for custom roles should therefore be an exception.

For more details, see
https://cloud.google.com/iam/docs/resource-hierarchy-access-control

Identity-Aware Proxy simplifies authorization to Google Cloud applications and VMs

- Works with applications deployed behind the Application Load Balancer in Compute Engine, GKE, or App Engine.
- When configured, it forces users to log in.
- Admins control who can access to app.
- Allows employees to securely access web-based applications without the need for a VPN.
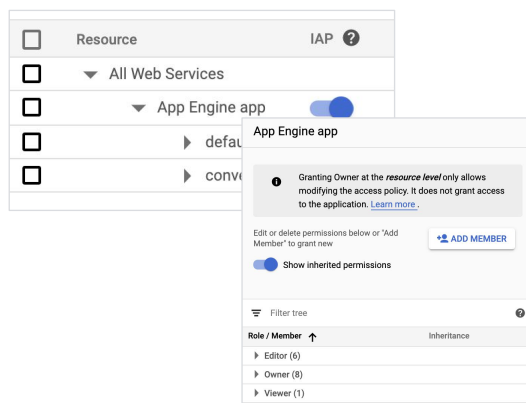
Google Cloud

Identity-Aware Proxy (IAP) provides managed access to applications running in App Engine standard environment, App Engine flexible environment, Compute Engine, and GKE. IAP establishes a central authorization layer for applications accessed by HTTPS, enabling application-level access control instead of using network-level firewalls. Using IAP requires signed headers or, for the App Engine standard environment, the Users API to secure your application.

The use case for IAP is when there is a need to enforce access control policies for applications and resources. IAP performs authentication and authorization checks. The application or resource can only be accessed through the proxy by members/users who have the configured IAM role. This allows access to resources with fine-grained access control but without the need for a VPN. It is a building block towards BeyondCorp, an enterprise security model that enables every employee to work from untrusted networks without needing a VPN.

IAP can be integrated with MDM and also is a key component in Google Cloud's context-aware solution, which enforces granular access control based on a user's identity and the context of their request. For more details, see https://cloud.google.com/context-aware-access/docs/overview.
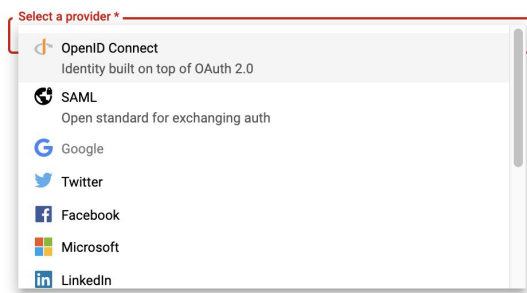
IAP supports TCP forwarding, which means it can be used to control access to services such as SSH and RDP. IAP means these services are not openly exposed to the Internet, ensuring that only authorized users gain access to these services.

Identity Platform provides authentication as a service

- Provides federated login that integrates with many common providers.

- Use it to provide sign-up and sign-in for your end users' applications.

Sign-in method
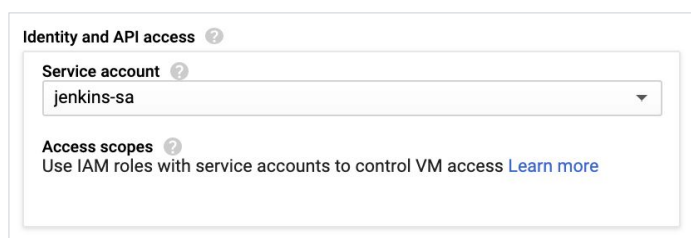
Select and configure an identity provider.

Select a provider *

OpenID Connect
Identity built on top of OAuth 2.0

SAML
Open standard for exchanging auth

G Google

Twitter

Facebook

Microsoft

LinkedIn

Google Cloud

Identity Platform is a customer identity and access management (CIAM) platform for adding identity and access management to applications. It provides authentication as a service that developers access via a set of SDKs. You need to select a service provider to use Identity Platform A broad range of protocol support is available including SAML, OpenID, Email and password, Phone, Social, and Apple.

03

Securing Machine Access

Service accounts can be used for machine or application identities

- Create a service account and grant it one or more roles.

- Can assign that service account to VMs or GKE node pools.

- Those machines run with only the rights granted by the roles.

Identity and API access ⓘ

Service account ⓘ

jenkins-sa	▾

Access scopes ⓘ
Use IAM roles with service accounts to control VM access Learn more

Google Cloud

A service account is a special kind of account used by an application, a virtual machine (VM) instance, or a GKE node pool, not a person. Applications or services use service accounts to make authorized API calls. The service account is the identity of the service and defines permissions which control the resources the service can access.

A service account is both an identity and a resource. A service account is used as an identity for your application or service to authenticate, for example, a Compute Engine VM running as a service account. To give the VM access to the necessary resources, you need to grant the relevant IAM roles to the service account. At the same time, you need to control who can create VMs with the service account so random VMs cannot assume the identity. Here, the service account is the resource to be permissioned. You assign the ServiceAccountUser role to the users you trust to use the service account.

Service accounts can be used for machine or application identities

- Generate and download a key when creating a service account.
- This key can be used for authentication.
- Key is downloaded as JSON.
- User-managed keys are extremely powerful credentials.
- Store the key safely.

Create key (optional)

Download a file that contains the private key. Store the file securely because this key can't be recovered if lost. However, if you are unsure why you need a key, skip this step for now.

+ CREATE KEY

```
{
  "type": "service_account",
  "project_id": "project-id",
  "private_key_id": "48e95bf20887235536f772dcf25d47b89f8cf49
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvAIBADANE
  "client_email": "my-service-account@project-id.iam.gservic
  "client_id": "113723034034071973858",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com
  "client_x509_cert_url": "https://www.googleapis.com/robot/
}
```

Google Cloud

Each service account is associated with public/private RSA key-pairs that are used to authenticate to Google. These keys can be Google-managed or user-managed.

Google-managed keys, both the public and private keys, are stored by Google, and they are rotated regularly (maximum usage period is two weeks). For user-managed keys, the developer owns both public and private keys. They can be used from outside Google Cloud. User-managed keys can be managed by the IAM API, gcloud command line tool, or the service accounts page in the Cloud Console. It is possible to create up to 10 key-pairs per service account to support key rotation.

User-managed keys are extremely powerful credentials, and they can represent a security risk if they are not managed correctly. You can limit their use by applying the constraints/iam.disableServiceAccountKeyCreation Organization Policy Constraint to projects, folders, or even your entire organization. After applying the constraint, you can enable user-managed keys in well-controlled locations to minimize the potential risk caused by unmanaged keys. Consider using Cloud Key Management Service (Cloud KMS) to help securely manage your keys.

The slide shows the generation of a key using the Cloud Console. The private key can be seen in the screen shot. It is your responsibility for storing the private key securely.

Can use service account keys to configure the CLI

- Allows you to grant controlled Google Cloud access to developers without giving them access to the Cloud Console.

- Also useful for automation when configuring VMs to run CI/CD pipelines.

- Use: `gcloud auth activate-service-account --key-file=[PATH TO KEY FILE]`

Google Cloud

For developers to gain controlled access to resources without acquiring access to the Cloud Console, it is possible to configure the gcloud command line utility to use service account credentials to make requests. The command above, gcloud auth activate-service-account, serves the same purpose as gcloud auth login but uses the service account instead of user credentials.

The key file contains the private key in JSON format.

04

Network Security

Remove external IPs to prevent access to machines outside their network

- Use a bastion host to provide access to private machines.

- Can also SSH into internal machines using Identity-Aware Proxy from the console and CLI.

- Use Cloud NAT to provide egress to the internet from internal machines.

All internet traffic should terminate at a load balancer, third-party firewall (proxy or WAF), API Gateway, or IAP. That way, internal services cannot be launched and get public IP addresses.

Google Cloud
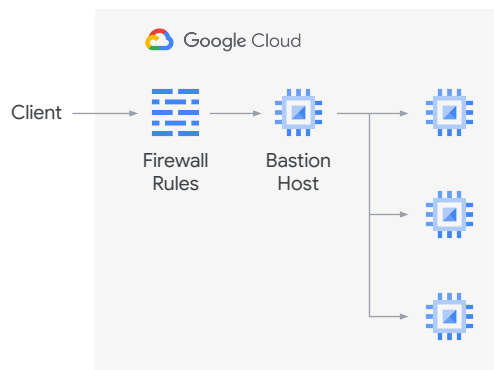
There are several options available for securely communicating with VMs that do not have public IP addresses. These services do not have a public IP address normally because they are deployed to be consumed by other instances in the project or maybe through Dedicated Interconnect options. However, for those instances that do not have an external IP address, it can be a requirement to gain external access, for example for updates or patches to be applied. The options for accessing the VMs include:

Bastion Host: A bastion host provides an external facing point of entry into a network containing private network instances. This host provides a single point of secure access and can be stopped to disable inbound SSH. This allows the connection to VMs without having to configure firewall rules. Typical hardening initial steps for a bastion host include limiting the CIDR range of source IP addresses that can communicate with the host and configuring firewall rules to only allow SSH to private VM addresses from the bastion host.

Identity-Aware Proxy: IAP was mentioned earlier in this section. Using SSH with IAP's TCP forwarding feature wraps the SSH connection inside HTTPS. The TCP forwarding then sends it to the remote instance.

Cloud NAT: When a VM has no public IP address, it cannot make direct connections to external services, which includes other Google Cloud services. To allow such instances to connect to other services using the public internet, it is possible to configure a Cloud NAT gateway machine to route traffic on behalf of the instance on
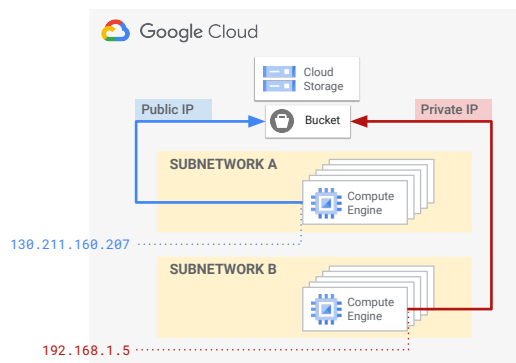
the network. VM instances that only have internal IP addresses can also use Private Google Access. We'll discuss this on the next slide.

From the options above, the preferred approach for inbound SSH is to use IAP, which is a built-in service, rather than configuring a one-off bastion host.

If you're working with GKE, it is possible to create private clusters that enable you to isolate nodes from having inbound and outbound network connectivity. Cloud NAT can be used to provide outbound internet access for private nodes. For ingress, there are various options for configuring access to endpoints ranging from no access from public IP addresses to the cluster endpoint through to any IP address being able to communicate with the cluster endpoint. For full details, see https://cloud.google.com/kubernetes-engine/docs/concepts/private-cluster-concept.

Private access allows access to Google Cloud services using an internal address

- Enabled when creating subnets.
- Allows access to Google Cloud services from VMs that only have internal IPs.
  - For example, a machine with only an internal IP would be able to reach a Cloud Storage bucket.


```
gcloud compute networks subnets update subnet-b \
     --enable-private-ip-google-access
```
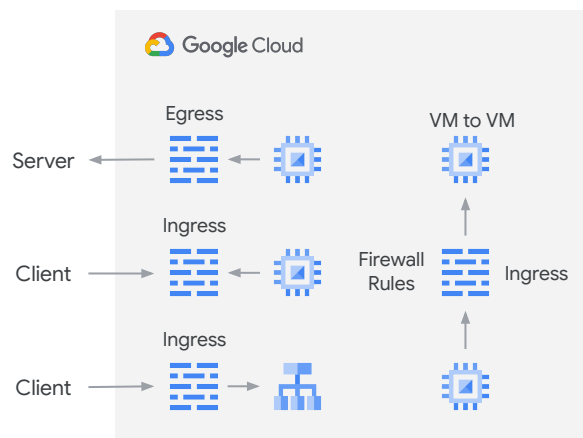
Google Cloud

VM instances that only have internal IP addresses (no external IP addresses) can use Private Google Access. They can reach the external IP addresses of Google APIs and services. The source IP address of the packet can be the primary internal IP address of the network interface or an address in an alias IP range that is assigned to the interface. If you disable Private Google Access, the VM instances can no longer reach Google APIs and services; they can only send traffic within the VPC network. Private Google Access is enabled on a subnet by subnet basis for subnets in a VPC network.

Private Google Access permits access to Cloud and Developer APIs and most Google Cloud services, with the exception of the following services:
- App Engine Memcache
- Filestore
- Memorystore
- Cloud SQL

Configure firewall rules
to allow access to VMs

- By default, ingress on all ports is denied.
- Add firewall rules to control which clients have access to which VMs on which ports.
- Application level security is the responsibility of the customer.

Google Cloud

Google Cloud Firewall rules allow you to deny or allow traffic to VM instances. The firewall rules are defined at the network level but also work for instance-to-instance communication on the same network. Every network has two implied firewall rules that permit all outgoing (egress) connections and block all incoming (ingress) connections.

When a firewall rule is configured, each rule applies to either ingress or egress, but never both. Other default rules for the default network include allow ingress connections for all protocols and ports among instances in the network. Also rules exist on the default network for allowing SSH, RDP, and ICMP ingress connections from any source to any instance on the network.

When a firewall rule is configured, the components to be specified include:

- **Direction** of traffic: ingress or egress
- A numerical **priority**: lowest number is the highest priority
- **Action**: allow or deny
- **Enforcement status**: enabled or disabled
- **Target**: instances to which rule applies
- **Source** for ingress or **destination** for egress
- **Protocol** and port

Control access to APIs using Cloud Endpoints

- Protect and monitor your public APIs.
- Control who has access to your API.
- Validate every call with JSON Web Tokens and Google API keys.
- Integrates with Identity Platform.

Cloud
Endpoints

For managing APIs, Google provides Cloud Endpoints. Endpoints is an API management gateway that helps you develop, deploy, and manage APIs on any Google Cloud backend. It is built on the open source Extensible Service Proxy. It runs on Google Cloud and leverages a lot of Google's services, for example Identity Platform, Cloud monitoring, trace, and logging.

Cloud Endpoints supports multiple authentication methods, including the following:

API keys: These are useful if you want to block anonymous traffic or if you want to control the number of calls made to your API or identify usage patterns in the API traffic. However, the keys cannot be used for identifying individual users, secure authorization, or identifying the creators of a project.

Service Accounts: This is for service-to-service requests. The calling service uses the service account's private key to sign a secure JSON Web Token (JWT) and sends the signed JWT in the request to the API. This use case is suitable for microservice applications for service-to-service communication.

Auth0: This authenticates and authorizes applications and APIs regardless of the identity provider, platform stack, or device. The client library provided by Auth0 generates and signs a JWT once the user signs in. ESP validates that the JWT was signed by Auth0. Auth0 is suited for consumer and enterprise web and mobile apps.
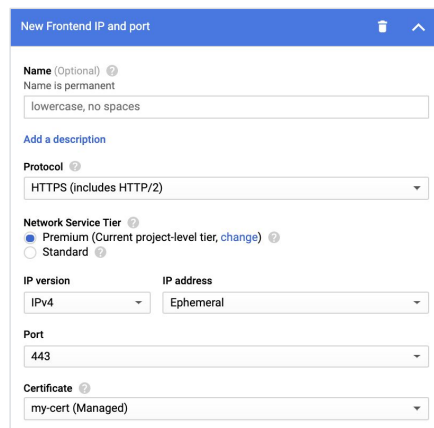
Google also provides Apigee for managing APIs with deployment options on cloud,

on-premises, or hybrid. It provides an API gateway, customizable portal for onboarding partners and developers, monetization, and deep analytics around your APIs. It also has out-of-the-box configurations to support traffic management, mediation, security, packaging APIs, developer key management, etc. You can use Apigee for any http/https backend, no matter where they are running (on-premises, any public cloud, etc.).

If you have a Google Cloud backend, there are scenarios where both Apigee and Cloud Endpoints could provide a solution. Feature set then plays a part. If you do not have a Google Cloud backend, Apigee is the option.

Restrict access to your services to TLS only

- All Google Cloud service endpoints use HTTPS.

- It's up to you to configure your service endpoints.

- In the load balancer setup, only create a secure frontend.

New Frontend IP and port

Name (Optional)
Name is permanent

lowercase, no spaces

Add a description

Protocol
HTTPS (includes HTTP/2)

Network Service Tier
● Premium (Current project-level tier, change)
○ Standard

IP version: IPv4   IP address: Ephemeral

Port: 443

Certificate
my-cert (Managed)

Google Cloud

The strong recommendation is to use HTTPS rather than plain HTTP. You are responsible for procuring the certificate yourself if you run your endpoint on a Compute Engine instance. If you use a Google load balancer in front of your Compute Engine instance, you can configure a frontend with a Google-managed SSL certificate, but this will require a domain name.
Cloud Run, Cloud Run functions, and App Engine offer SSL and do not require a domain name.

Leverage Google Cloud network services for DDoS protection

- Global load balancers detect attacks and drop them.
- Enabling the CDN will protect backend resources.
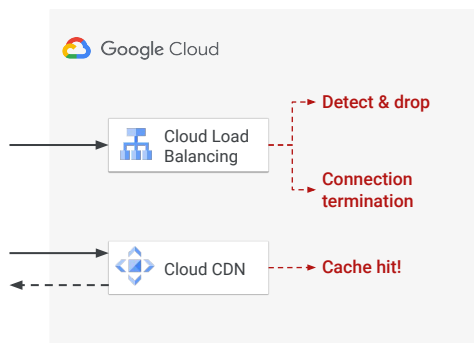
Google provides infrastructure DDoS support through load balancers at level 3 and level 4 traffic. Examples include SYN floods, IP fragment floods, and port exhaustion. No additional configuration is required to activate DDoS defense. If the system detects an attack, it configures load balancers to drop or throttle traffic. Backend resources can be protected with Cloud CDN, as DDoS attempts result in a cache hit.

Use Google Cloud Armor to create network security policies

- Can allow or deny access to your Google Cloud resources using IP addresses or ranges.

- Create allow lists to allow known addresses.

- Create deny lists to block known attackers.

Google Cloud

For additional features over built-in DDoS, such as IPv4 and IPv6 allow or deny, and defense against application-aware attacks such as cross-site scripting and SQL injection, Google offers Google Cloud Armor, which works in conjunction with global Application Load Balancers and enables you to deploy and customize defenses for your internet-facing applications. It's based on the same technologies and global infrastructure that we use to protect Google services like Search, Gmail, and YouTube.

Google Cloud Armor security policies enable the access or denial of HTTP(S) requests to load balancers at the Google Cloud edge as close as possible to the source of incoming traffic. This prevents unwelcome traffic from consuming resources or entering the VPC networks.

Google Cloud Armor supports layer 7 web application firewall (WAF) rules

- Predefined rules for preventing common attacks like SQL injection and cross-site scripting

- Flexible rules language allows you to allow or deny traffic using request headers, geographic location, ip addresses, cookies, etc.

- Examples:

```
inIpRange(origin.ip, '9.9.9.0/24')
request.headers['cookie'].contains('80=BLAH')
origin.region_code == 'AU'
inIpRange(origin.ip, '1.2.3.4/32') &&
request.headers['user-agent'].contains('WordPress')
evaluatePreconfiguredExpr('xss-canary')
```

Google Cloud

These are based on the OWASP Modsecurity core rule set version 3.0.1 (https://modsecurity.org/crs/)

Google Cloud Armor provides a rules language for filtering request traffic (https://cloud.google.com/armor/docs/rules-language-reference). The rules are prioritized and the highest priority match applied. These rules allow both access and denial of requests. The rules are used to write expressions that consist of attributes that can be inspected and operations that can be performed on the attributes. As an example, consider the first expression in the slide above: inIpRange(origin.ip, '9.9.9.0/24). In this case the expression returns true if the origin IP in a request is within the 9.9.9.0/24 range.

The second line above, request.headers['cookie'].contains('80=BLAH'), returns true if the cookie 80 with value BLAH exists in the request header. The expressions can be combined logically with logical AND (&&) and OR (||).

The expressions are written and assigned to an allow or deny rule to determine which traffic is allowed.

05

Encryption

Google Cloud provides server-side encryption of data at rest by default

- Data Encryption Key (DEK) uses AES-256 symmetric key.
- Keys are encrypted by Key Encryption Keys (KEK).
- Google controls root keys in Cloud KMS.
- Keys are automatically periodically rotated.
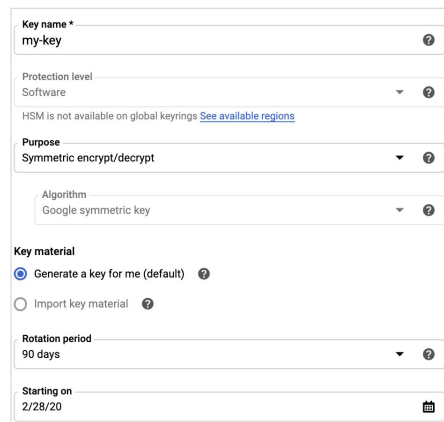- On-the-fly decryption by authorized user access with no visible performance impact.

Client

TLS / HTTPS

Cloud Network

Cloud Storage

App Engine

AES-256
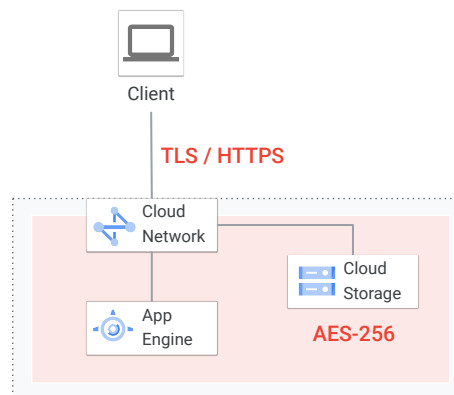
Google Cloud

Google Cloud encrypts customer data stored at rest by default, with no additional action required from users.

Data in Google Cloud is broken into subfile chunks for storage, and each chunk is encrypted at the storage level with an individual encryption key. The key used to encrypt the data in a chunk is called a data encryption key (DEK). Because of the high volume of keys at Google, and the need for low latency and high availability, these keys are stored near the data that they encrypt. The DEKs are encrypted with a key encryption key (KEK). The KEK keys are stored in Google's Key Management Service (KMS), a repository built specifically for storing keys. Keys are automatically rotated periodically by KMS. The standard rotation period is 90 days, and KMS can store up to 20 versions, meaning data requires re-encryption at least once every five years.

For compliance reasons, you may need to manage your own keys

- Customer-managed encryption keys are created in the cloud using Cloud Key Management Service (KMS).

- You create the keys and specify the rotation frequency.

- You can then select your keys when creating storage resources like bucket and disks.

Key name *
my-key

Protection level
Software

HSM is not available on global keyrings See available regions

Purpose
Symmetric encrypt/decrypt

Algorithm
Google symmetric key

Key material
◉ Generate a key for me (default)
○ Import key material

Rotation period
90 days

Starting on
2/28/20

Google Cloud

It may be a requirement that you have to manage your own encryption keys rather than use the automatically generated keys described on the previous slide.

In this scenario, you can use KMS to generate what are known as Customer Managed Encryption Keys (CMEK). These keys are stored in KMS for direct use by Cloud services. The responsibility for the lifecycle is now yours: you have to take care of rotation, either manually or by selecting an automatic rotation period. You are responsible for deletion also. The keys can be used by Cloud services and for application layer encryption in any Google Cloud product. The Cloud services they can be used with are BigQuery, Cloud Build, Dataproc, Cloud Storage, and Compute Engine.

Customer-supplied encryption keys are created in your environment and provided to Google Cloud
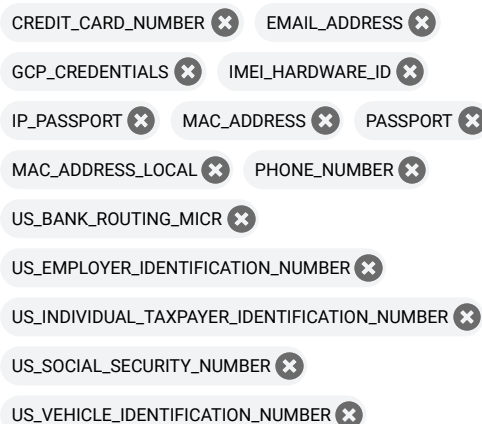
- Use your own keys with Google Cloud services.

- CSEK are supplied by the calling application per-API call.

- Only cached in RAM by Google.

- They decrypt a single payload (or column) or block of returned data.

- Supported by Compute Engine (persistent disks) and Cloud Storage.

Google Cloud

When you're required to generate your own encryption key or manage it on-premises, Google Cloud supports Customer Supplied Encryption Keys (CSEK). The keys are kept on-premises, not in the Google Cloud. The keys are provided as part of API service calls, and Google only keeps the key in memory. They can be used with Cloud Storage and Compute Engine.

For Google Storage, the CSEK is used as the Key Encryption Key to wrap the Data Encryption Key. For persistent disks on Compute Engine, the CSEK is used in combination with a persistent disk cryptographic nonce to generate a CSEK-derived key used to encrypt the data.

The Data Loss Prevention API can be used to protect sensitive data by finding it and redacting it

- Scans data in Cloud Storage, BigQuery, or Firestore.

- Can also scan images.

- Detects many different types of sensitive data, including:
  - Emails
  - Credit cards
  - Tax IDs

- You can add your own information types.

- Can delete, mask, tokenize, or just identify the location of the sensitive data.

CREDIT_CARD_NUMBER ⊗   EMAIL_ADDRESS ⊗

GCP_CREDENTIALS ⊗   IMEI_HARDWARE_ID ⊗

IP_PASSPORT ⊗   MAC_ADDRESS ⊗   PASSPORT ⊗

MAC_ADDRESS_LOCAL ⊗   PHONE_NUMBER ⊗

US_BANK_ROUTING_MICR ⊗

US_EMPLOYER_IDENTIFICATION_NUMBER ⊗

US_INDIVIDUAL_TAXPAYER_IDENTIFICATION_NUMBER ⊗

US_SOCIAL_SECURITY_NUMBER ⊗

US_VEHICLE_IDENTIFICATION_NUMBER ⊗

Google Cloud

Cloud DLP helps users better understand and manage sensitive data. It provides fast, scalable classification and redaction for sensitive data elements like credit card numbers, names, social security numbers, US and selected international identifier numbers, phone numbers, and Google Cloud credentials. Cloud DLP classifies this data using more than 90 predefined detectors to identify patterns, formats, and checksums, and even understands contextual clues. You can optionally redact data as well, using techniques like masking, secure hashing, tokenization, bucketing, and format-preserving encryption. Custom detectors can be added.

A big benefit of DLP is the ability to discover, classify and report on data from BigQuery, Cloud Storage, Firestore, and a streaming content API that enables support for additional data sources and applications.

Activity 12

🕐 15 min

Modeling Secure Google Cloud Services

Refer to your Design and Process Workbook.

- Draw a diagram that depicts your case study security requirements.

Quiz

Question #1

Question

What Google Cloud service can you use to enforce the principle of least privilege when using Google Cloud?

A. IAM members and roles

B. Firewall rules

C. Encryption keys

D. SSL certificates

Google Cloud

Question #1

Answer

What Google Cloud service can you use to enforce the principle of least privilege when using Google Cloud?

A. **IAM members and roles** ✅

B. Firewall rules

C. Encryption keys

D. SSL certificates

Google Cloud

A. This is the correct answer. The principle of least privilege requires user permissions that are just enough to do what they need, and no more. IAM provides this level of control.

B. This is not correct. Firewall rules allow or deny traffic rather than users.

C. This is not correct. Encryption keys are used for integrity, not access.

D. This is not correct. SSL certificates are used for authentication and encryption, not permission.

Question #2

Question

You don't want programmers to have access to production resources. What's the easiest way to do this in Google Cloud?

A. Create a firewall rule that blocks developer access to production servers and databases.

B. Create development and production projects, and don't give developers access to production.

C. Use different service accounts for production and development resources with your project.

D. Set up private access and Identity-Aware Proxy.

Question #2

Answer

You don't want programmers to have access to production resources. What's the easiest way to do this in Google Cloud?

A. Create a firewall rule that blocks developer access to production servers and databases.

B. Create development and production projects, and don't give developers access to production. ✅

C. Use different service accounts for production and development resources with your project.

D. Set up private access and Identity-Aware Proxy.

A. This is not correct. Firewalls allow or deny traffic, not users or roles, so it would not be possible to restrict developers via a firewall, or at least not in an easy and maintainable way.

B. This is the correct answer. The simplest way is to have separate projects and not give developers access to the production project.

C. This is not correct. There would be many challenges with using this solution so as with answer A, while it's possible, it is far from simple or sensible.

D. This is not correct. IAP is primarily for enabling access to Google Cloud from untrusted networks for applications and SSH/RDP access, but not all infrastructure.

Question #3

Question

Which Google Cloud features could help reduce the risk of DDoS attacks?

A. Global external Application Load Balancer

B. Cloud CDN

C. Google Cloud Armor

D. All of the above

Question #3

Answer

Which Google Cloud features could help reduce the risk of DDoS attacks?

A. Global external Application Load Balancer

B. Cloud CDN

C. Google Cloud Armor

D. **All of the above** ✅

The correct answer is D. A global external Application Load Balancer mitigates and absorbs many Layer 4 and below attacks such as SYN flood, IP fragment floods, and port exhaustion. CDN caches cacheable content at points of presence close to users.

In the event of a DDoS attack for cacheable content, the requests are sent to points of presence, not to your servers/infrastructure, thus increasing the likelihood of the attack being absorbed. Google Cloud Armor is built for DDoS mitigation, working with Cloud Load Balancing to detect DDoS attacks.

Question #4

Question

What do you have to do to enable encryption when using Cloud Storage?

A. Simply enable encryption when configuring a bucket.

B. Enable encryption and upload a key.

C. Create an encryption key using Cloud Key Management Service, and select it when creating a Cloud Storage bucket.

D. Nothing: encryption is enabled by default.
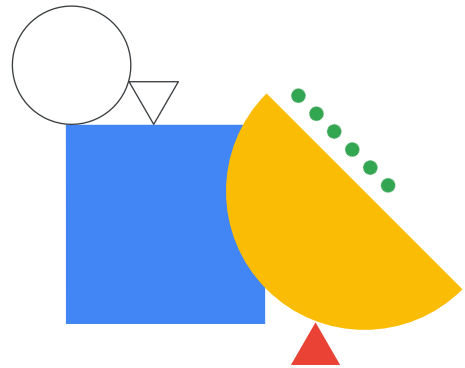
Google Cloud

Question #4

Answer

What do you have to do to enable encryption when using Cloud Storage?

A. Simply enable encryption when configuring a bucket.

B. Enable encryption and upload a key.

C. Create an encryption key using Cloud Key Management Service, and select it when creating a Cloud Storage bucket.

D. **Nothing: encryption is enabled by default.** ✅

The correct answer is D. Cloud Storage always encrypts data on the server side before it is written to disk. For server side encryption, there are options of customer supplied or customer managed encryption keys, but these are only usually used for compliance reasons and are not necessary.

Review: Security

In this module, we covered how to secure our Google Cloud resources. This includes securing both the network and our stored data. We also covered how to secure people using IAM, Cloud Identity, and Identity Aware Proxy, and how we can secure our applications and machines using service accounts.

Remember, security should be put first; everything else will follow from this.

More resources

Google Cloud security products

https://cloud.google.com/security/products/

Encryption at rest

https://cloud.google.com/security/
encryption-at-rest/default-encryption/

Encryption in transit

https://cloud.google.com/security/
encryption-in-transit/

Google Cloud

These resources are useful to consult when considering securing your applications and data.