3
Trust and Compliance

In this chapter, we will look at the very important aspects of trust and compliance. The first part of the chapter focuses on trust, including how Google enables security and privacy and provides customers with full transparency. We will walk through examples of how you can access transparency logs and how they are used. The last part of the chapter covers the different compliance standards and programs that Google Cloud is compliant with, and how you can gain access to compliance reports. We will look at ways to access compliance reports using Compliance Reports Manager.

In this chapter, we will cover the following topics:

- Security and data privacy
- Building trust using access transparency and access approval
- Understanding compliance on Google Cloud

Establishing and maintaining trust

Data privacy and the protection of customer data are critical in establishing trust. There is no shortcut to it: time and experience are the only two factors that help in maintaining and establishing trust. Translating that to the cloud means that customers don't have the luxury of spending time testing whether a **cloud service provider** (**CSP**) can be trusted. Therefore, CSPs such as Google Cloud use *compliance* in order to demonstrate and establish *trust*.

Google creates trust by means of transparency. The Google Cloud Enterprise Privacy Commitments dictate how Google Cloud protects the privacy of its customers. Let's take a look at the privacy principles that Google defines:

- As a customer, you control and own your data. You define and control where you want to store your data (in terms of geographical location), and only you can decide where you want to move/copy your data to. Google does not have any control over copying or replicating data outside of the region that you have selected. Google only processes the data as per the various agreements.

- Customer data is not used for any ads or targeting other users for ads. Customer data does not belong to Google, which is very clearly stated by Google's policies and principles. Google Cloud will never use customer data in other Google products, including ads, or use customer data for any purposes not described in mutual agreements with customers.
- As part of the many different compliance programs that Google Cloud is compliant with, Google transparently collects data that it requires to perform necessary business functions and is aligned with compliance standards such as ISO 27001, SOC 2, GDPR, and other privacy best practices.
- None of the data hosted on Google Cloud is sold by Google to any partners or third parties for any purpose. Google's policies on data handling are very well defined and have both procedural and technical controls built around them. Google has set a very high bar for how customer data is stored, hosted, and served. Furthermore, Google's compliance reports and legal agreements with customers provide details on Google Cloud's compliance with regard to using customer data in accordance with legal, compliance, and regulatory requirements. Google never sells or uses customer data outside of the scope of the aforementioned obligations.
- Every Google product has both security and privacy built into it from the beginning. There are well-defined policies and controls that dictate how the security and privacy of each product are managed and designed. These aspects are considered of the highest importance.

As we can now see, building a trusted cloud environment that customers rely on for their business-critical information requires security, transparency, and privacy to be built into it. Google Cloud's principles ensure that these are addressed. It's difficult to establish trust in the absence of these principles and independent audits to demonstrate adherence to the compliance standards.

In the next section, we will look at how transparency is provided by Google Cloud. We will walk through a common scenario to help you understand how transparency logs and access approval work on Google Cloud.

Access Transparency and Access Approval

Before we discuss the Access Transparency and Access Approval products, let's understand why they are so important. We discussed in the previous section how transparency plays a key role in establishing trust, and it also helps Google to differentiate its security posture. At the time of writing, other major **cloud service providers** (**CSPs**) do not offer transparency logs. Customers who are highly regulated or have compliance requirements need to view and share logs of activities performed by their CSPs. Analysts such as Gartner have highlighted transparency logs as a key feature for CSPs.

Now that we have established the importance of transparency for CSPs, we will look at the product capability that Google offers. Google has two products for transparency: **Access Transparency**, which provides near real-time logs whenever Google administrators access your Google Cloud environment, and **Access Approvals**, where you can control administrative access to your data.

Access Transparency

Access Transparency logs are a feature provided by Google Cloud that offer customers visibility into the actions performed by Google employees when they access customer data. These logs provide a detailed record of any access made by Google's support and engineering teams to customer content or data stored within Google Cloud services. Access Transparency logs help customers meet regulatory compliance requirements, security audits, and internal governance policies. By providing a comprehensive record of access activities, these logs contribute to building trust and ensuring the integrity and privacy of customer data within the Google Cloud platform. Google follows a set of best practices to control access to customer data. These best practices include the following:

- **Least privilege**: All access by Google employees is denied by default. If and when access is granted, it is both conditional and temporary and limited to only what is necessary for a user to perform their role and function.
- **Limit singular access to data**: It is extremely difficult for Google employees to singularly access customer data without another individual being involved. Therefore, quorum-based access control for data is enforced.
- **All access must be justified**: Google personnel, by default, do not have access to customer data. When Google personnel have privileged access, they must have the required permissions, business justification, and consent. Again, quorum-based controls apply to ensure that there is oversight of the task being performed.
- **Monitoring and alerting**: Monitoring and response processes exist to identify, triage, and remediate violations of these access controls.

As you can see, there are a number of procedural and technological controls and policies in place to ensure that insider risk and administrative privileges are managed and controlled. When any of these activities happen due to business requirements for a customer environment, Google logs the activities and shares them when the customer requests the transparency logs. Customers request Access Transparency logs by enabling the service through the **Identity and Access Management** section of the Google Cloud console.

Adding transparency logs to purpose-built tools has a clear intent: to establish trust and help customers in gaining visibility of their data and demonstrate adherence to compliance requirements with auditors.

A key distinction to note between Access Transparency logs and admin activity logs is that admin activity logs are limited to recording the actions of the users or administrators of your Google Cloud organization, whereas Access Transparency logs record actions taken by Google employees. Access Transparency logs include details such as the impacted resource and the relevant action taken, the time of the action, the reason, and information about who accessed it.

There are many reasons why you want to have access to transparency logs. Let's take a look at some of them:

- To verify that the Google personnel who accessed your content did so with a valid business justification, such as support requests
- To verify that Google support engineers have not made an error while performing instructions you as a customer provided them with
- To verify and track compliance with legal or regulatory requirements
- To collect and analyze transparency logs and ingest them in a security information and event management tool

Transparency logs and the products that support them are listed on Google Cloud's website: `https://packt.link/R03rx`.

In order to enable Access Transparency, you need to have either the Premium, Enterprise, Gold, or Platinum support level. If you are not sure what support level you have, you can check that by accessing the following link: `https://packt.link/m0rhf`. Once Google has validated that you have the relevant support level, they can enable transparency logs.

Enabling Access Transparency

Using these instructions, Access Transparency can be either enabled or disabled. Before proceeding, you need to check your permissions at the organization level. Navigate to the Google Cloud Console page using this URL: `https://packt.link/9iuM4`. Select your Google Cloud organization when prompted to do so and check that the correct IAM admin role related to Access Transparency has been set (`roles/axt.admin`):

1. Select the Google Cloud project that you would like to use:
    - It's important to note that although Access Transparency is configured at the project level, once it's enabled, it will be available for the entire organization.
    - If you have a requirement where you need to only enable project-level Access Transparency, you will need to contact Google Cloud Support. This function can be configured from the console.

You can only configure Access Transparency inside a project where the project is already associated with a billing account. If your project is not associated with a billing account, then you will not be able to enable the Access Transparency service.

2. Navigate to **Billing** from the console menu on the left. If you see the message **This project is not associated with a billing account**, you will need to select a different project or associate the project with a billing account. Refer to the help on the page for guidance on how to change the billing account for a project.
3. Next, navigate to the **IAM & Admin | Settings** page and then click **ENABLE ACCESS TRANSPARENCY FOR ORGANIZATION** button shown in *Figure 3.1*.

   If you are unable to see the **ENABLE ACCESS TRANSPARENCY FOR ORGANIZATION** button, it is due to one of the following reasons:

   - You do not have a billing account associated with your project
   - You do not have the IAM admin permission to perform the function
   - The required support package is missing

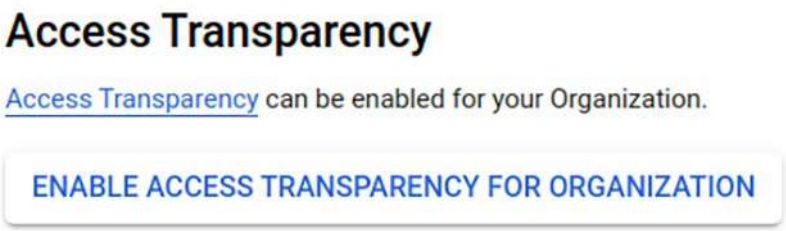

Figure 3.1 – Access Transparency enabled confirmation

Access Approval

Next, we will look at what Access Approval is and how it works. Access Approval ensures that you as a customer give your consent to Google support engineers to access the content in your environment. If you decide, for whatever reason, that you don't want to give that permission, access will be denied. Irrespective of whether you approved access or denied access, a transparency log activity entry will be created.

We will look at an example of a customer support workflow to better understand how the approval process works.

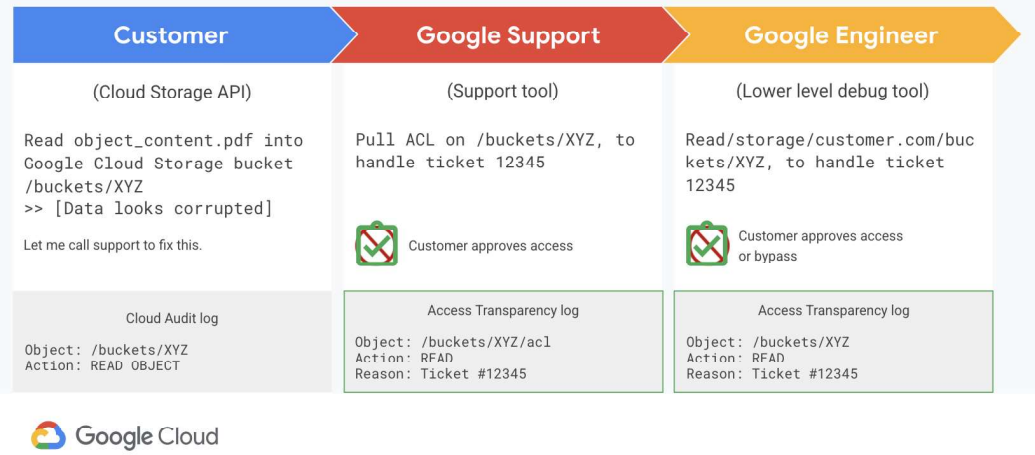

Figure 3.2 – Access Approval workflow

In this example, the customer tries to access a file stored in a Google Cloud Storage bucket and is unable to access it. The customer then opens a support ticket. At this stage, admin activity logs are created for the API call that was made to read the object. Next, the Google support engineer, using the support tool, attempts to retrieve the **access control list** (**ACL**) for the particular bucket that the customer is trying to access. The Google support engineer does not have access, so a request is sent to the customer to either approve or deny the request. Once the customer approves the request, the Google support engineer can retrieve the ACL that was inaccessible earlier. At this stage, a new transparency log entry is created that captures the **object**, which is `bucket/XYZ/acl`, the associated **action**, which is `READ`, and the **reason**, which is the ticket number, `#12345`. In the last step, the Google support engineer needs to open the file that the customer was trying to access. The reason for this is to establish that the file is now accessible, so the ticket can be resolved. Again, the support engineer will be denied access to the document and another Access Approval request will be routed to the customer to take action. At this point, the customer can either allow or deny the request, which depends on a number of factors, such as whether the document is sensitive. If the data inside the document is generic and not sensitive and there are no other concerns, the customer then approves the request. Once the request is approved, the Google support engineer can access the document and a corresponding transparency log is created: **OBJECT bucket/XYZ ACTION Read and REASON is the ticket number #12345**.

The point of explaining the entire process of how Access Approval works is to demonstrate how customers can have control over their data, which is one of the key principles we discussed in the previous section, that is, how Google ensures that trust is established.

You have the ability to enable Access Approval for the services you want to use it for. A list of supported Google Cloud products can be found here: `https://packt.link/f5MDC`.

Access Approval requires your consent only for requests for the content stored in the services you select. When you enroll for Access Approval, the following options are available to you:

- You can automatically enable Access Approval for all supported services. This is the default option.
- You can selectively enable Access Approval for services with GA-level support. When you select this option, it automatically enrolls all the services that Access Approval will support in the future with GA-level support.
- Finally, you can also choose the specific services you want to enroll in Access Approval.

Using Access Approval gives you increased control over the implementation and usage of this feature, allowing you to determine how and for which services you enable it. It provides flexibility in tailoring access approval processes according to your specific requirements.

However, it's crucial to note that there are certain exclusions that apply to Access Approval. These exclusions refer to specific scenarios or services where the feature may not be applicable or available. It's important to familiarize yourself with these exclusions to ensure you have a clear understanding of the limitations and can make informed decisions regarding the use of Access Approval in your organization.

By understanding and considering these exclusions, you can effectively utilize Access Approval in a manner that aligns with your security and compliance needs while adhering to any applicable restrictions or exceptions. Here are a few scenarios in Access Approval that are exceptions:

- Non-humans have programmatic access that is allowed and reviewed by Google processes. A compression task that runs on content is one example, as is disk destruction during the content deletion process. When binary authorization checks for accesses, it ensures that the job came from code that was checked into production.
- Manual access, such as legal access, is used when Google accesses customer content to meet legally binding requirements; in these instances, the Access Approval process is bypassed.
- When Google looks at customer content to fix a service outage, this is called a *service outage*.
- An Access Approval request will not be generated if any other Access Transparency exception is logged and if any activity fails to generate an Access Transparency log.

Similar to Access Transparency, for Access Approval you need to have either the Premium, Enterprise, Gold, or Platinum support level. If you are not sure what support level you have, you can check by accessing the following link: `https://packt.link/II1xE`.

Once you have validated that you have a required support level, you can enable Access Approval by following the steps in the next section.

Configuring Access Approval

To enroll in Access Approval, follow these steps:

1. In the Google Cloud console, navigate to the project, folder, or organization for which you want to enable Access Approval.
2. Go to the **Access Approval** page: `https://packt.link/o84Ct`.
3. To enroll in Access Approval, click **ENROLL**.

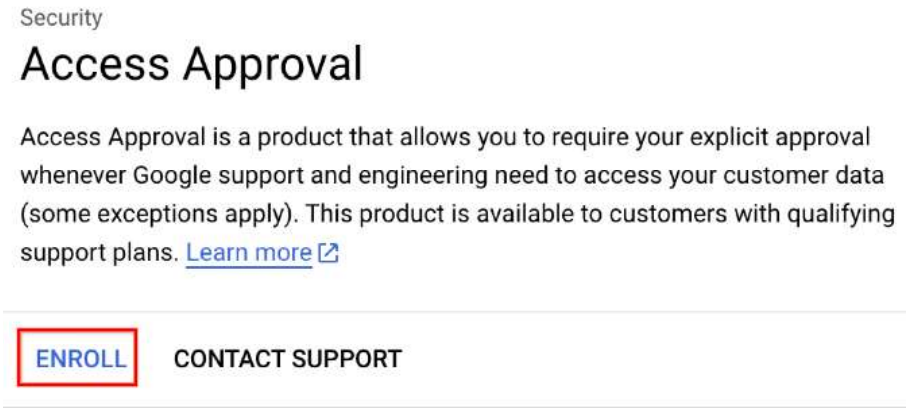

Figure 3.3 – Access Approval enrollment

4. In the dialog box that opens, click **ENROLL**.

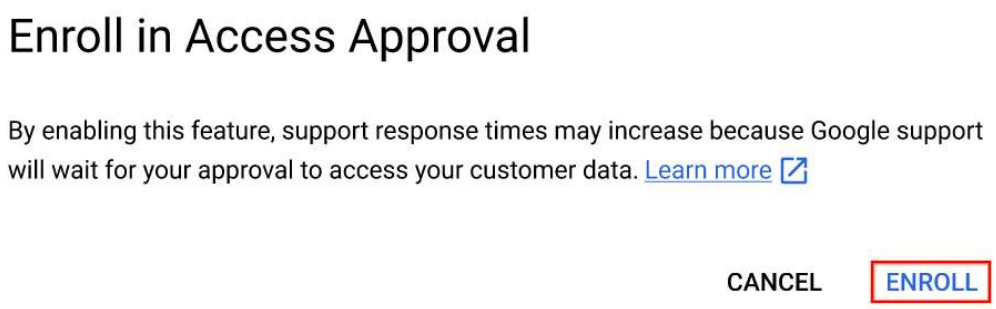

Figure 3.4 – Access Approval enrollment success

Understanding Access Transparency and Access Approval helps you understand how trust is established using transparency. If you need to build an environment on Google Cloud that is regulated and has access to transparency logs, these products will help you achieve the goal. Google has also built its Assured Workloads product, which utilizes transparency logs. Assured Workloads is out of scope for the Google Cloud Professional Security Engineer exam, but if you wish to read more about it, you can refer to the links in the *Further reading* section at the end of this chapter.

Security and privacy of data

Earlier in the chapter, we covered the data privacy principles adhered to by Google. In the previous chapter, we also covered how Google Cloud enforces the security of data by default, such as by encrypting data in transit by default. When you store your data on Google Cloud, you will read and write data; thus, there will be times when data will be out of Google-enforced security boundaries. Enforcing the encryption of data in transit ensures that data is secure.

When data is stored in any Google Cloud Storage products, the encryption of data at rest is enforced by default. This improves the security posture for any organization adopting Google Cloud, as you don't have to manage the underlying encryption technology to manage the key lifecycle or encrypt data because these controls are fully managed by Google Cloud.

Google Cloud provides flexible options to help you align with Google privacy principles and control your data. You can choose the level of control you want for your keys. For example, you can use Google Cloud **Key Management Service** (**KMS**) to create and manage your own keys, or import your own keys and store them in software, which would be KMS, or in Cloud **Hardware Security Module** (**HSM**), which is a FIPS 140-2 Level 3 certified appliance.

Many customers who have stringent compliance and regulatory requirements have the option of using an **external key management** (**EKM**) system. EKM systems are hosted by Google partners and are integrated with external HSMs. Google has built partnerships with selected EKM vendors, who integrate their EKM solutions with Google Cloud KMS. With this option, customers can use the existing key store, which can be HSM-based, and use the EKM solution to point to the keys. This gives customers the greatest control over their keys as Google does not persist them. At the time of writing, only Google Cloud provides such a capability to customers via its partners: Equinix, Fortanix, Ionic, Thales, and Unbound.

On the topic of data privacy and security, another key area that is a requirement for many customers and is also a growing regulatory requirement is **data residency**. Google is committed to meeting customer needs with regard to data residency. There are several capabilities and assurances that Google provides to help customers meet this requirement. Data residency can only be provided for the services listed here: `https://packt.link/ENTe8`. Let's look at some of the capabilities that Google Cloud offers for data residency:

- You can control where you store your data. This applies to ensuring that your data stays in a physical geographical location, also known as a Google region. Google provides assurance, in the form of compliance reports, that Google will under no circumstances move or copy your data to another region. This can only happen if you decide to move your data. You can also apply an organization policy at the folder or project level, whereby the resource location constraint is applied to enforce that only approved regions can be used to launch new services. This can be further combined with Cloud IAM, where you can enable or disable certain services for users so that they do not move data to locations outside of the approved geographical regions.

- While restricting data to a certain location is important, another control that is available to further enforce data localization is Cloud KMS. Cloud KMS has data regionalization assurances in place. Therefore, as a customer, in order to maintain the data residency of your keys and data, you need both your keys and your data to be within the boundaries of a specific Google Cloud location. This can be achieved by creating a key ring in Cloud KMS that is bound to a specific location.

Figure 3.5 – Google Cloud KMS geo-location configuration for key ring

Customer-Managed Encryption Keys (**CMEKs**) provide additional data security controls. They also ensure that resource creators can choose the appropriate locations for their resources.

- Another Google Cloud product that can help ensure that only users from a certain location can access data is VPC Service Controls. This product allows you to enforce egress and ingress policies as well as to define IP address ranges and geo-locations from which you want users to access services and data. Combined with a Cloud IAM policy, you can allow only authorized users from locations that you define. All Google Cloud services inside the *service perimeter*, a virtual boundary inside the Virtual Private Cloud, can be accessed based on a defined policy. This ensures that data is not moved to locations that are not allowed.

Google recently also announced a new feature, **Key Access Justifications** (**KAJ**). Every time there is a request to decrypt data using Cloud KMS or an external KMS, a detailed justification is provided including details such as which key was requested to decrypt the data. This is combined with a mechanism where you can approve or deny key access using an automated policy that you can configure. Using these controls, you can also limit Google's ability to decrypt your data. As the owner of the data, you are the arbitrator of who has access to your data and not the cloud provider.

Third-party risk assessments

A third-party risk assessment or vendor risk assessment is often a requirement for many regulated customers who want assurance from Google Cloud about specific controls. An example of this would be a financial institution such as a bank that wants to host workloads on Google Cloud and needs Google Cloud to complete a vendor questionnaire. Google Cloud provides self-assessment

questionnaires. These are complimentary documents that cover Google Cloud's security controls and can help customers assess the security of their service. These self-assessments are available via Google Compliance Manager, which can be accessed here: `https://packt.link/B15d7`.

Some of the available assessments are as follows:

- Google Cloud's **Cloud Security Alliance** (**CSA**) **STAR** self-assessment is available here: `https://packt.link/rnqoe`.
- The **Standardized Information Gathering** (**SIG**) core questionnaire can be accessed by customers to perform an initial assessment of third-party vendors to help determine how security risks are managed across 18 different risk domains. This report is accessible here: `https://packt.link/6fTSQ`.
- The IHS Markit KY3P due diligence questionnaire can be downloaded from `https://packt.link/syPOz`.
- Google Cloud's data processing and security terms (`https://packt.link/qQMu6`) are also available to customers, covering how Google processes customer data.

Besides these resources, Google Cloud also provides security whitepapers on topics ranging from data encryption and privacy to incident response and compliance, which can be used to understand Google Cloud-specific security controls. The whitepapers can be accessed here: `https://packt.link/c0aIn`.

Compliance in the cloud

In this section, we will cover two topics: how you can access the compliance reports that are made available by Google Cloud, and some of the tools and capabilities that are available to achieve continuous compliance in the cloud.

Google products undergo compliance reviews by independent third parties, and the relevant compliance reports are made available to customers. There are two sets of compliance reports: one set can be downloaded from the Google Cloud website and is generally available to anyone; the other set can be requested by a Google **Technical Account Manager** (**TAM**) if the Google customer has a TAM assigned to their organization. You can find Google Cloud's compliance reports here: `https://packt.link/eaDuj`. New Google Cloud products often have a delay of a few months before they are added to the compliance scope. Google has a scheduled review cycle, and new products are added based on that. Each product undergoes SOC 2 and ISO audits first, followed by market-specific certifications.

Google is compliant with international standards such as ISO 27001, ISO 27017, ISO 27018, SOC 2, and SOC 3. Besides these, there are some country-specific compliance frameworks, such as FedRAMP for the US and MTCS for Singapore. Similarly, for other countries, such as Germany and Australia, Google Cloud is compliant with their respective in-country privacy and compliance requirements. It is important to note that every Google Cloud region is compliant with a consistent set of compliance standards. Even though FedRAMP may not be relevant to customers in some parts of the world, Google Cloud regions are compliant with its standards. This is one of the key advantages to customers having all these compliance standards built into Google Cloud. While Google Cloud is committed to compliance with standards across all its regions, certain regions may not be compliant with certain standards due to local laws and regulations. This means that specific regulations that are applicable in the region must be taken into consideration when assessing compliance with any particular standard. For example, some regions may not be compliant with certain security standards, or a region may have specific data protection laws that must be adhered to. In these cases, it is important to ensure that the region where you are using Google Cloud's services is compliant with the applicable standards before committing to any particular service. The following is a diagram of the compliance standards and frameworks that Google Cloud is compliant with. The full list of compliance standards can be found here: `https://packt.link/fJr39`.

Figure 3.6 – Third-party certifications and regulatory compliance

We'll now take a brief look at some of the key international standards that Google Cloud is compliant with. The intent is not to cover them in depth but at a high level for your understanding of the scope of these compliance standards. From an exam perspective, you will not be tested on your knowledge of compliance standards:

- **ISO 27001**: This is an **Information Security Management System** (**ISMS**) framework, consisting of 14 groups, 35 control objectives, and 114 controls. ISMSs are documented based on ISO 27001 standards, which provide guidelines for setting up, implementing, operating, monitoring, reviewing, maintaining, and enhancing them. As one of the core compliance programs, ISO 27001 encompasses all types of organizations (commercial businesses, government agencies, and non-profit organizations). Google Cloud undergoes a yearly third-party audit to certify individual products against this standard.
- **ISO 27002**: The ISMS controls the implementation details. This is not a certification standard but rather an advisory document containing a list of best practices and guidelines that organizations can use to meet the controls listed in the ISO 27001 standard. It recommends information security controls to mitigate risks to the confidentiality, integrity, and availability of information. Since this is an advisory document, organizations can choose to use these controls, a subset of them, or a completely different set to certify against the ISO 27001 standard.
- **ISO 27017**: This framework is an extension of ISO 27001 and focuses specifically on information security relating to cloud computing beyond the general guidelines in the preceding document. This standard advises both cloud service providers and cloud service customers to create a matrix of individual and shared responsibilities.
- **ISO 27018:** This is a set of security guidelines that specifically focus on the processing and protection of **Personally Identifiable Information** (**PII**). It extends the control objectives of ISO 27002, which is a broader standard for information security management systems. Security safeguards for public cloud service providers operating as PII processors are the primary emphasis of this standard.

In the next section, we will take a look at how you can download these compliance reports using Compliance Reports Manager.

Compliance reports

Compliance Reports Manager provides Google Cloud customers with the ability to download selected compliance reports directly from `https://packt.link/C6JRN`. New reports are launched at the end of every month.

You can access Compliance Reports Manager via this link: `https://packt.link/yBhnU`.

Let's do a quick walk-through of how you can download these reports:

1. Using the filter, you can select what report you want to download. The options in the filter include **Industry**, **Region**, **Report Type**, and **Product Area**. **Industry** has two options: **Industry-agnostic**, such as the ISO, SOC, and CSA reports, and **Government/Public** sector, which refers to reports relevant to government agencies such as IRAP for Australia. Please work with your Google Cloud representative to help you locate those reports.

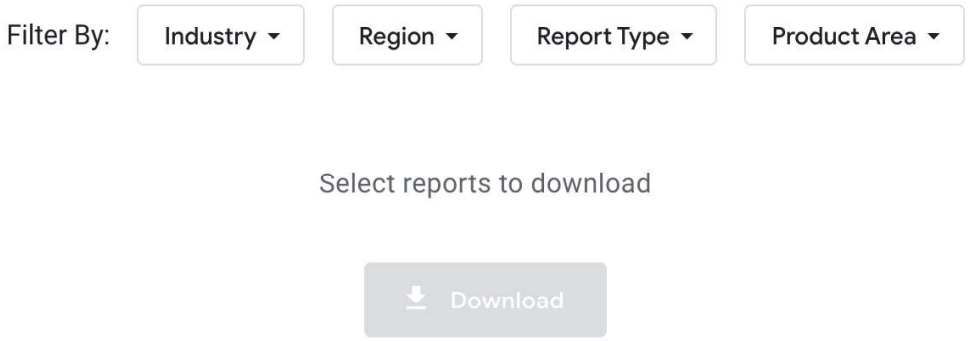

Figure 3.7 – Compliance Reports Manager

2. Next, you specify using the filter what report you want, and you will see the option to select and then download the report from Compliance Report Manager. In this example, we have left **Industry** as the default, **Region** is **Global**, **Report Type** is **Audit Report**, and **Product Area** is **Google Cloud**.

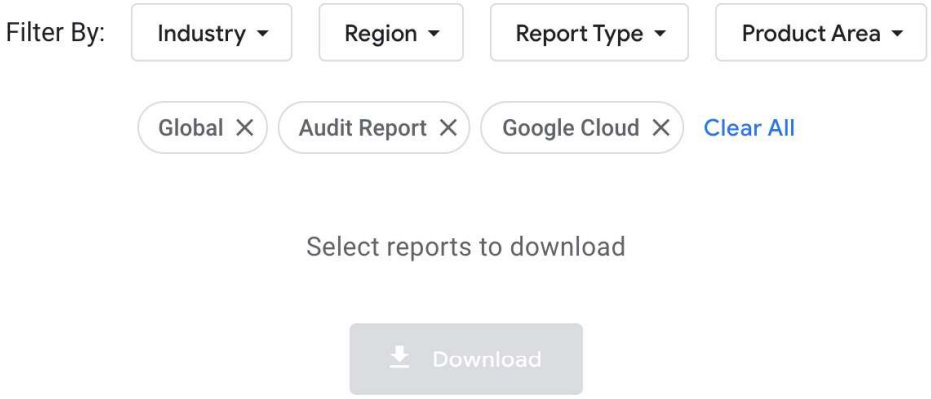

Figure 3.8 – Google Cloud Compliance Report Manager filter

3. Next, you can select the report you want and click **Download**. In our example, we have a PCI-DSS report selected to download.

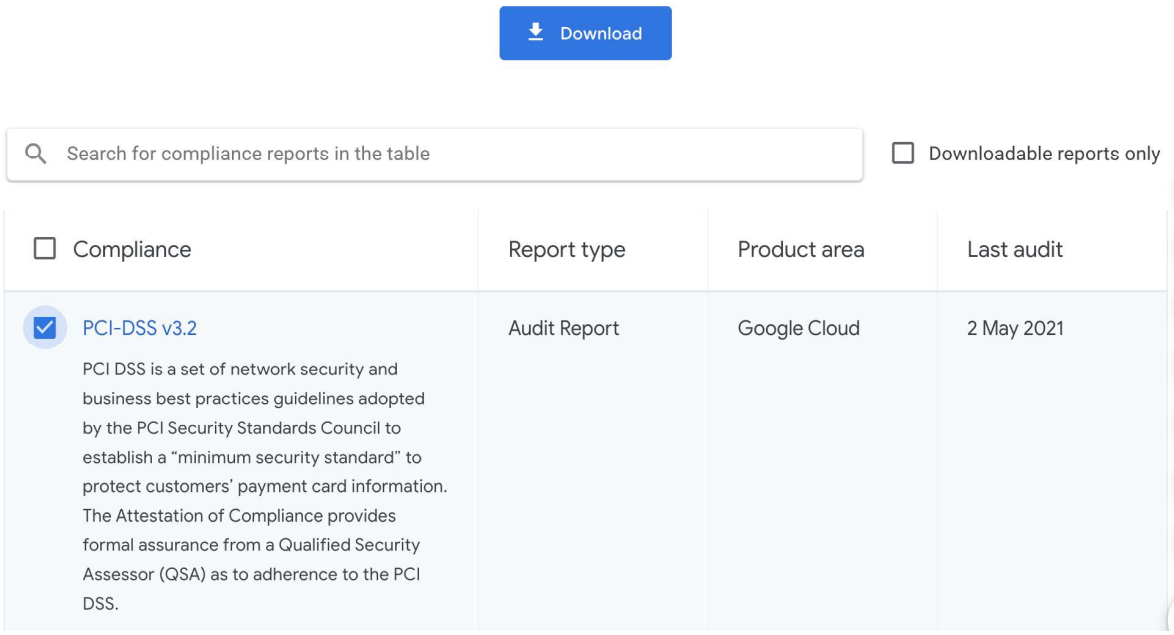

Figure 3.9 – Compliance manager: PCI-DSS report download

After this, you can download and view the PCI-DSS audit report. In a similar way, you can download other reports.

Continuous compliance

All the benefits that the cloud has to offer, specifically, capabilities such as automation and orchestration, make compliance in the cloud much easier to achieve. One of the key advantages that many customers gain when moving from traditional on-premises data centers to the cloud is the ability to achieve compliance in a more automated way. The cloud helps you achieve compliance and audit objectives in an automated and on-demand fashion rather than waiting on quarterly manual audits. This reduces the time needed to resolve security-related issues, thus improving your overall security posture. Google Cloud offers an elegant solution to meet compliance objectives.

In this section, we will look at cloud-native capabilities that are available on Google Cloud that can help you to achieve continuous compliance. Continuous compliance means that you can run your compliance checks on your cloud assets against the policy that is being implemented. Whenever an asset changes its state, such as a running Google Compute Engine resource with new known vulnerabilities, the compliance-as-code policy executes to check the state and returns a result to indicate compliance or non-compliance. If compliant, no action is required; however, depending on your organizational maturity, you can take different actions. For example, you may just want a notification to be sent to your security and compliance teams to take action and review the finding, or you can build an automated process that means that once an asset is found to be non-compliant, a Cloud Functions script will execute and take corrective action to make the asset compliant again.

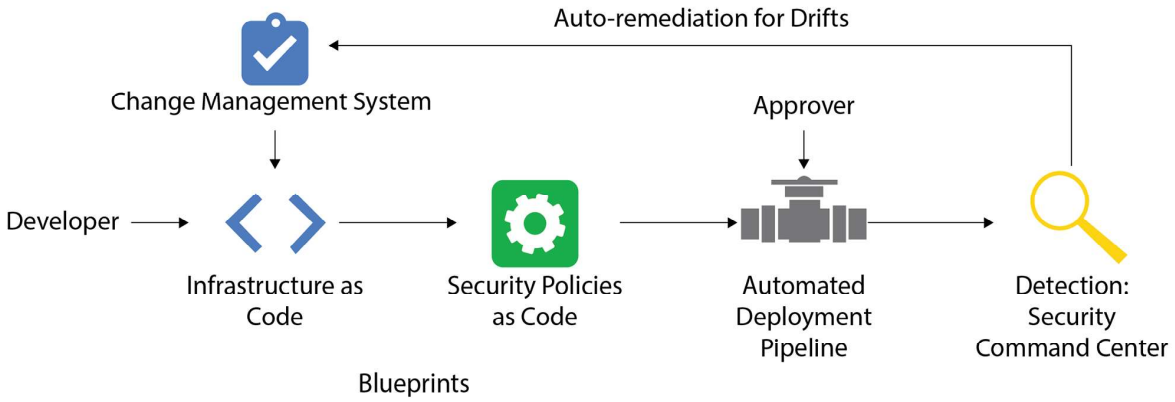

Figure 3.10 – Continuous compliance pipeline

Google Cloud offers a comprehensive set of tools and services that help organizations ensure that their systems and applications remain in compliance with laws and regulations. This includes automated compliance checks, continuous monitoring, and alerting capabilities to help organizations respond quickly to potential issues.

Summary

In this chapter, we looked at how Google Cloud establishes trust using transparency and applies privacy principles backed up by independent audits and compliance. We also covered two key products that help customers establish trust: Access Transparency and Access Approval. In terms of ensuring the security and privacy of data, we covered the controls available, such as encryption and KMS. Furthermore, we looked at the importance of data residency and how Google Cloud provides capabilities, features, and products to support the localization of data. We covered third-party and vendor risk assessments and the support that Google Cloud extends in helping customers to be compliant with regulatory requirements. Finally, we covered aspects of compliance in the cloud such as how you can download Google Cloud compliance reports, standards, and an overview of achieving continuous compliance in the cloud.

In the next chapter, we will cover the resource hierarchy in Google Cloud.

Further reading

For more information on Google Cloud security, trust, and compliance, refer to the following links:

- Trusting your data with Google Cloud: `https://packt.link/mEnMK`
- Compliance offerings: `https://packt.link/k7PMI`
- Resource on risk and compliance vendor due diligence: `https://packt.link/ugIhD`
- Data residency, operational transparency, and privacy for European customers on Google Cloud: `https://packt.link/I9lgg`
- Overview of Assured Workloads: `https://packt.link/BgkjA`

4
Resource Management

In this chapter, we will look at resource management and understand some of the key components, such as projects, folders, and organizations. When building your environment on Google Cloud, these components can help you do segmentation at a macro level. We will also look at organizational policy constraints, some of the pre-built constraints that are available, and how the inheritance of policy works for **Identity and Access Management** (**IAM**) and firewall rules. We will also cover Cloud Asset Inventory, which is an essential part of resource management, and its role from a security perspective. We will end the chapter with some best practices and design considerations for resource management.

In this chapter, we will cover the following topics:

- Overview of Google Cloud Resource Manager
- The resource hierarchy
- The Organization Policy Service
- Organization Policy constraints
- Policy inheritance
- Hierarchical firewall policies
- Cloud Asset Inventory
- Design considerations and best practices

Overview of Google Cloud Resource Manager

Google Cloud Resource Manager acts like a container for your cloud resources, allowing you to group your resources in a hierarchical way within the project, folder, or organization. Think of Resource Manager as a high-level way to perform macro-level segmentation. This not only helps you define the entire organization's structure but also the implementation of security guardrails that can be inherited. More on this in the *Policy inheritance* section.

Figure 4.1 is an example of how you can structure your organization on Google Cloud. The top-level organization is where all your other components such as folders and projects are created. Organizing your resources in a hierarchical way lets you manage aspects such as access control and other configuration settings. The same applies to IAM policies, which can be applied at different levels and are then inherited top-down.

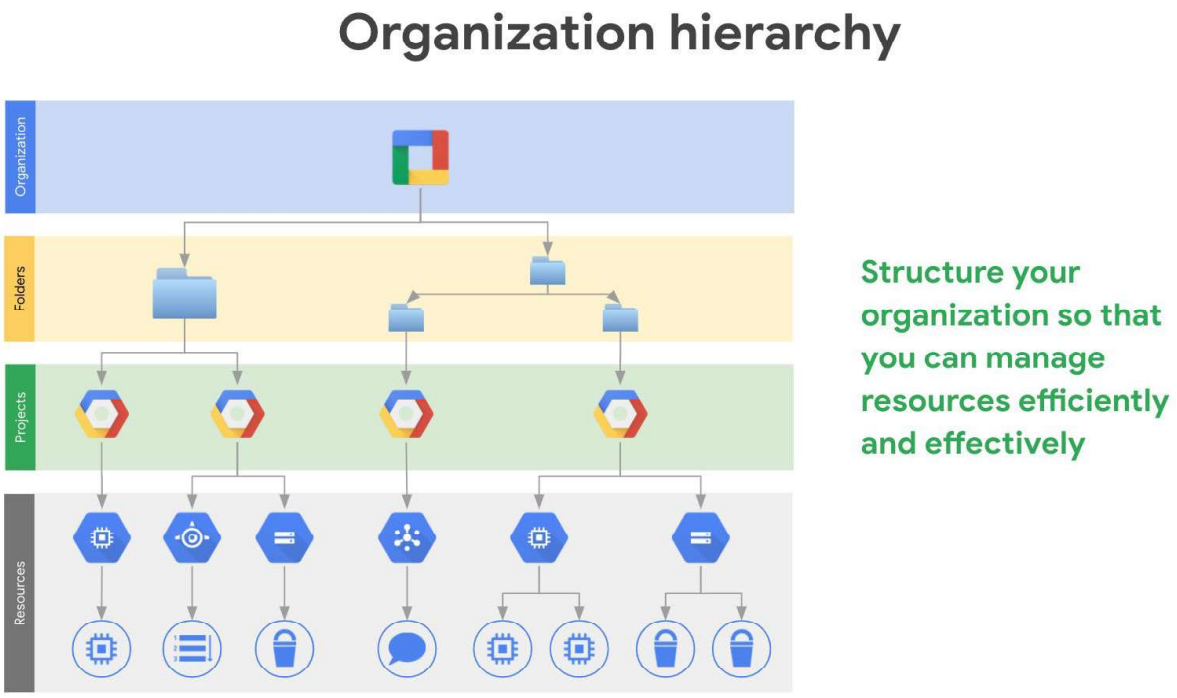

Figure 4.1 – Organization hierarchy

Let's take a look at the key components that make up the organization hierarchy and further explore how policies are applied and inherited and also some best practice examples for organization structure.

Understanding resource hierarchy

The key components that make up the resource hierarchy are as follows:

- **Organization**: The top-level component—all other components are linked to this.
- **Folders**: Used to group similar projects to consistently apply policies. These are optional but highly recommended.
- **Projects**: Where all the resources, such as your compute instances, databases, and others, exist.

Now, let's look at each of these components in detail. From a certification standpoint, you will be tested on resource hierarchy and the topics covered in this chapter. It's important to understand how to manage and create resources and apply access policies and organizational constraints. We will cover all these topics in this chapter.

Organization

This is the root node and hierarchical super-node of the project. It is closely associated with the Workspace / Cloud Identity account. It's a single directory containing the organization's users and groups. The organization is created from the Cloud Identity account. After creating a Cloud Identity account, you need to manually create an organization through the console to provision the GCP organization. The organization doesn't contain users or groups but rather holds GCP resources. Users and groups are part of Cloud Identity, but they receive IAM permissions in GCP. The organization at the top-level node represents, for example, a company in the resource hierarchy. Google Cloud Identity and Workspace are provisioned within one organization. When a user creates a project in Google Cloud, only then is an organization automatically created. Organizations are also needed to manage the resource lifecycle. They act as central resource management for all projects that are created. All members of the organization belong to the organization by default. With an organization, you can also apply policies across all resources, such as IAM policies and organization policies. We will cover this in the *Policy inheritance* section later in this chapter.

In the following figure, let's understand the link between Cloud Identity and Workspace. The link between Google Cloud Identity and Google Workspace is that Google Cloud Identity provides the foundation for user authentication and access management, which is used by Google Workspace to manage and secure user accounts and access to its suite of productivity tools. You cannot have an organization without Cloud Identity (which is a prerequisite). It's important to note that the Cloud Identity / Workspace super admin is the overall owner of the domain verification and also has the right to create any IAM role in Google Cloud.

> **Note**
>
> You can read more about Cloud Identity in *Chapter 5*, *Understanding Google Cloud Identity*, and *Chapter 6*, *Google Cloud Identity and Access Management*.

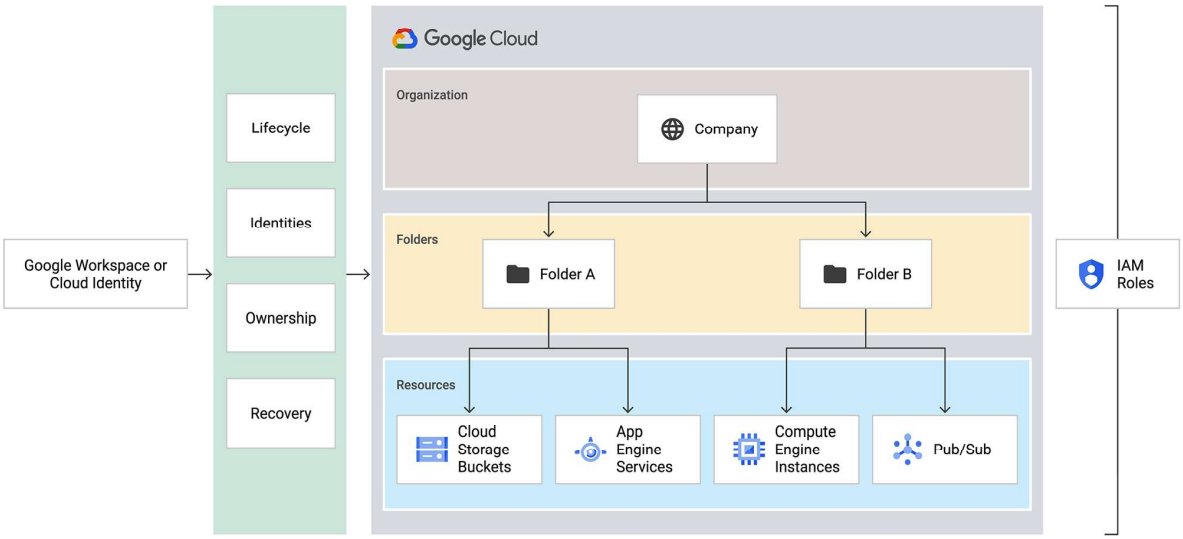

Figure 4.2 – Organizations

The organization resource that is exposed via the Resource Manager API has the following construct:

```
{
  "creationTime": "2020-01-07T21:59:43.314Z",
  "displayName": "my-organization",
  "lifecycleState": "ACTIVE",
  "name": "organizations/34739118321",
  "owner": {
    "directoryCustomerId": "C012ba234"
  }
}
```

In the preceding snippet, the number `34739118321` represents the unique identifier, which is the organization ID; `my-organization` is the display name that is generated by the Cloud Identity

domain; the creation time and the last modified time are also recorded; and the `owner` field is set by Workspace and cannot be changed—it is the customer ID that is specified in the Director API.

Folders

The main function of the folders is to group and segment your projects. You can define boundaries when designing your organization structure. Similar to organizations, the folders have certain parameters and principles to follow, which will be discussed in this section. Folders are the nodes in the organization hierarchy and are used to model a workflow and access pattern. Let's take a look at *Figure 4.3* to understand how you can use folders in your GCP organization.

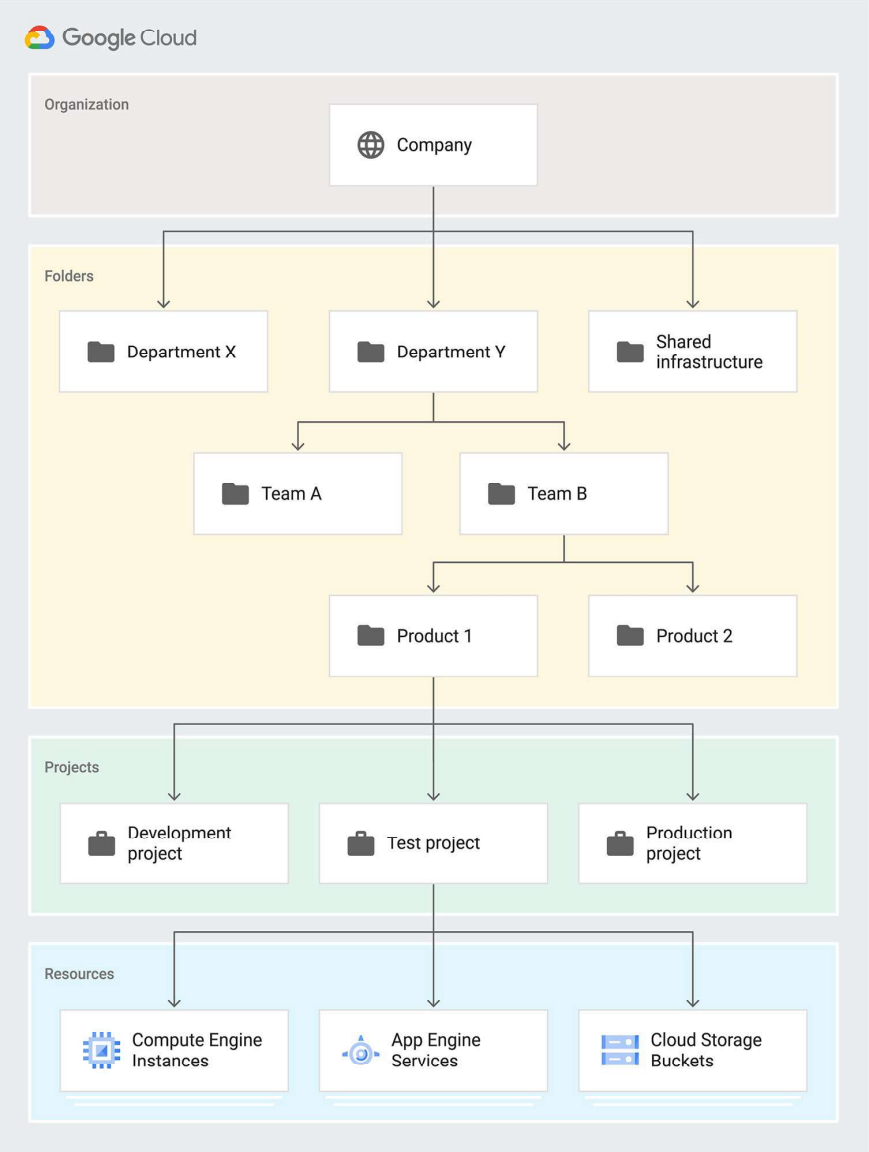

Figure 4.3 – Example enterprise organization

In this example enterprise organization structure, the first level of folders represents a company's departments, such as **X** and **Y**. As folders have the capability to go up to 10 levels deep (a hard limit), you can create more folders, which could represent different teams, and then further segment them into different products. This gives you the ability to define access control and permissions at the department level, team level, or product level. Note the policy inheritance applied is top-down. There is flexibility in modifying the folder tree and it's also possible to move projects, but you have to be mindful of policy inheritance, as that could potentially impact you moving one project from, say, **Team A** to **Team B**. An important design criterion is to decide on a folder structure that is in line with how your policies should be applied.

Projects

A project is a base-level object that is required for using Google Cloud. Your organizations can contain projects and folders. A project is where all your cloud resources are created. Each project is completely separate from other projects and so are the resources inside a project. Projects are useful to group specific resources from a functional and access standpoint. There is no cost associated with creating projects and you can create as many as you want (it's important to note that there are limits that may apply based on the account type, such as Enterprise versus Developer). Projects are not bound to geographical location and serve as an IAM enforcement point.

Let's take a look at the construct of a project and what it consists of using the following example snippet:

```
{
  "createTime": "2020-01-07T21:59:43.314Z",
  "lifecycleState": "ACTIVE",
  "name": "my-project",
  "parent": {
    "id": "634792535758",
    "type": "folder"
  },
  "projectId": "my-project",
  "labels": {
     "my-label": "prod"
  },
  "projectNumber": "464036093014"
}
```

Each project has two identifiers: one is a project ID, which is a customizable name that you can create, and the other is a project number, which is automatically assigned by Google when a project is created and is unique. This project number is read-only. You can also add labels to projects such as if a project is used for production (represented as `prod`) or for filtering projects. The lifecycle state is displayed as `ACTIVE` or `DELETE_REQUESTED`.

The project ID and project name are used when you are interacting with cloud resources. For every request to manage your cloud resources, you need to specify the project information.

Understanding how IAM policy inheritance works

We will cover cloud IAM in depth in *Chapter 6*, *Google Cloud Identity and Access Management*. In this section, we will understand how hierarchical IAM policy inheritance works in organizations. Cloud IAM allows you to configure fine-grained access to your Google Cloud resources. It lets you configure who (users) have access (permissions/roles) to what (resources) by means of an IAM policy.

The following figure depicts how IAM policy inheritance works, which is top-down, meaning you can configure an IAM policy at the organization level, folder level, or project level, and even at the resource level. If you recall from previous sections, we looked at how an organization hierarchy can be created based on your access boundaries.

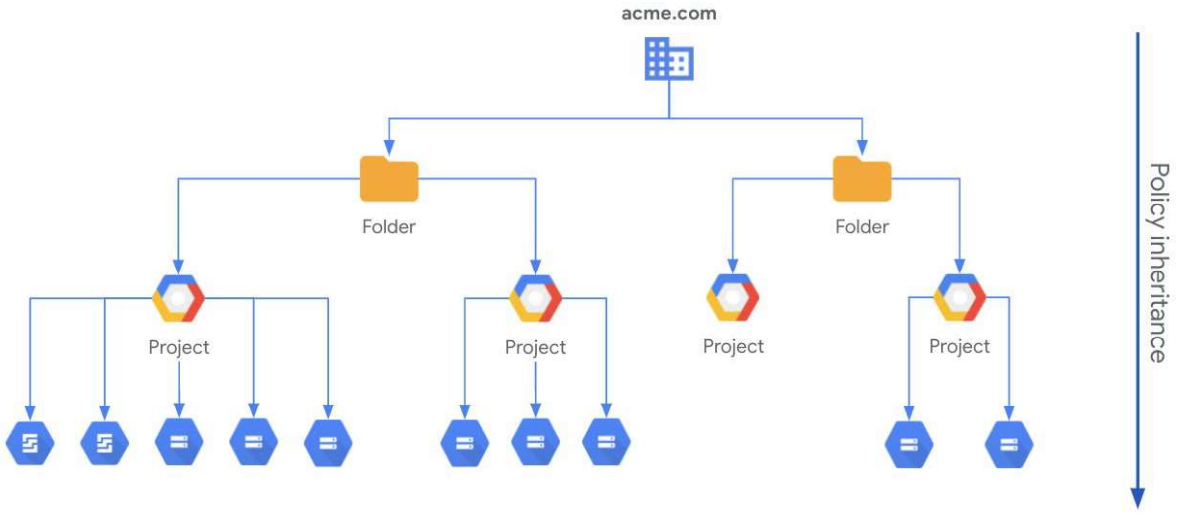

Figure 4.4 – IAM policy inheritance

> **Note**
>
> The effective policy for a resource is the union of the policy set at that resource and the policy inherited from its parent.

Essentially, what this means is that resources inherit the policy from the project, and the project inherits the policy from the folders and the organization. From the preceding figure, if you give Alice, the user, the role of Project Editor at the folder level, then Alice will have access as Project Editor for all the projects that are under the specific folder. If you assign Bob the role of an instance admin at the resource level for a given project, then Bob will only have access as an instance admin for the compute instance in that particular project.

Another aspect to remember is while you have the ability to migrate projects from one folder to another, IAM policy inheritance will apply, meaning when you move the project, it will inherit the IAM policies for that particular folder, and permissions applied on the project will be kept. The permissions associated with the previous folder will no longer be applied and the new folder permissions will be inherited.

Let's further look at an example of how IAM policy inheritance works; we will look at policy inheritance for Compute Engine, as illustrated in *Figure 4.5*.

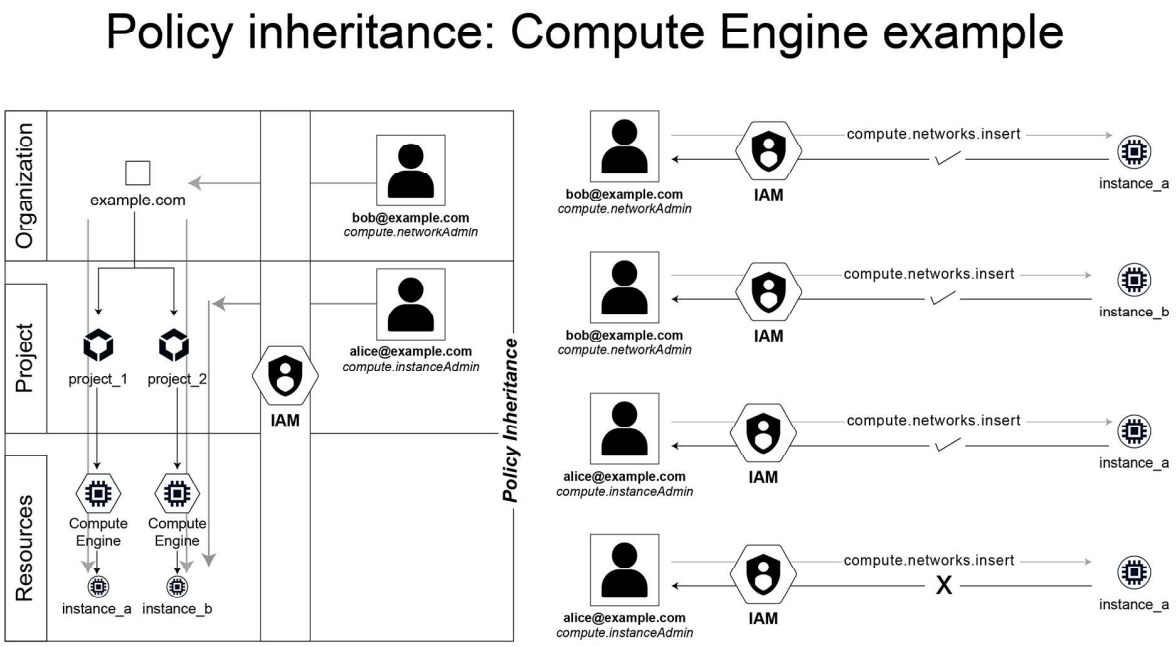

Figure 4.5 – IAM policy inheritance—Compute Engine

In *Figure 4.5*, Bob is granted access at the organization level as `compute.networkAdmin`; therefore, Bob will have access to perform IAM admin functions as governed by the policy for both `instance_a` and `instance_b`. Another user, Alice, is assigned `compute.instanceAdmin` at the `project_2` level. In this case, Alice can perform the `compute.instance.insert` function for `instance_b`, which is grouped under `project_2`, but will not have the permissions to perform the `compute.networks.insert` function for `instance_a` as Alice does not have the permissions at the project level for `project_1`.

Next, we will look at how you can apply Organization Policy Service constraints to your resources in Google Cloud.

Applying constraints using the Organization Policy Service

The Organization Policy Service provides you with a centralized and programmatic method to apply constraints across your organization and the respective cloud resources. We often dictate requirements that need to be applied organization-wide; these constraints are non-negotiable and every resource in your cloud environment has to adhere. This is where Organization Policy Service constraints come in, giving you the ability to centrally apply these policies to avoid or mitigate misconfigurations downstream in your projects and cloud resources. This helps you create guardrails that you can define, which can be based on regulatory requirements or internal compliance. The development teams who are consuming Google Cloud services do not have to worry about applying these policies and duplicating the effort. This further reduces the number of errors that may happen as you are now centrally in control of applying organization-wide constraints.

It is important to know the difference between IAM and Organization Policy Service constraints. While IAM focuses on who can take an authorized action on a particular resource, policy constraints focus on the what, meaning what restrictions are enforced on the resources dictating how they can be configured. An example would be giving access to a user called Bob to perform administrative functions on Compute Engine using an IAM policy, but by defining a constraint of not allowing external IP addresses to be assigned to the compute engine, you can restrict Bob from configuring an external IP address for any instance under the organization.

Let's further understand how an organization policy is applied, as per *Figure 4.6*.

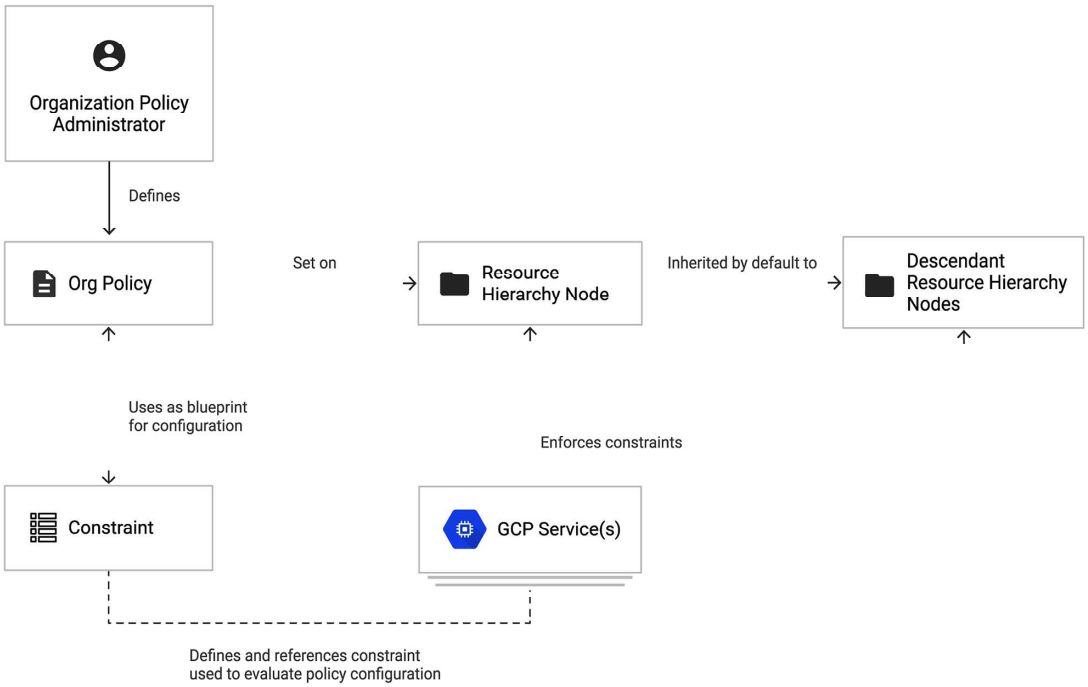

Figure 4.6 – Organization policy

As per the preceding illustration, an **Organization Policy Administrator** can apply an **Org Policy** at the organization, folder, or project level. An **Org Policy** must define a set of constraints that enforce restrictions as part of the policy. Once the constraints are defined, the policy can then be applied at the level decided by the administrator. Note that an **Org Policy** can be overridden at a lower level if needed by an **Organization Policy Administrator**. Next, we will take a look at the different types of constraints that can be applied.

Organization policy constraints

Constraints, simply put, are nothing but restrictions that can be applied to a resource. You have the ability to apply these constraints at different node levels, such as organization-wide or at the folder or project level. Google Cloud defines these constraints as a guardrail that defines what can and cannot be done for a particular resource.

Every constraint is defined by a set of attributes and these include a unique name such as `constraints/compute.disableSerialPortAccess`, a human-friendly name with a description of what the constraint is, followed by the default behavior of the constraint.

Figure 4.7 shows a list of a few constraints that are highly recommended for every GCP cloud environment to enforce.

Services	Constraints	Description	Useful for
Google Compute Engine	External IPs for VM instances	Defines a set of VM instances allowed to use external IP addresses.	Ensuring **minimal external surface**. VMs should normally get internal IPs only.
	Skip default network creation	Skips the creation of the default network and related resources during project creation.	Enforcing usage of **centrally managed and secured VPC networks**.
	Require OS Login	Enables OS Login on all newly created projects.	Ensuring SSH access to VMs is **centrally managed by IAM,** and not SSH keys stored as project/VM metadata.
Cloud IAM	Domain restricted sharing	Defines the set of members (domains) that can be added to Cloud IAM policies.	Protect against **malicious acts and human mistakes** by **ensuring access** only for users in **whitelisted domains**.
GCP	Resource location restriction (**Beta**)	Defines the set of locations where location-based GCP resources can be created.	**Compliance with regulations** that restrict resources location.
Cloud Storage	Enforce bucket policy only	Requires buckets to use Bucket Policy only where this constraint.	**Object-level access** policies don't consider bucket-level policy. They are hard to get visibility into, and can become a **security risk**.

Figure 4.7 – Example organization policy constraints

> **Note**
> You can find a complete list of supported constraints at `https://packt.link/Onku2`.

Policy inheritance

Organization policy inheritance works by applying the policy at the node level (organization/folder/project), where the policy will be inherited by the resources. You can also create custom policies at child nodes, and these can either merge or overwrite the inherited policy.

Let's further understand the policy inheritance rules to better understand how policies are evaluated and applied:

- The first rule applies when no policy is defined. If a resource doesn't inherit or have a policy defined anywhere in the hierarchy, then the constraint's default behavior is enforced.
- The second rule states that if a resource node has a policy set, it will override any parent policy set for that node. However, there is an exception to this rule. If the node policy has the `inheritFromParent` attribute set to `true`, the resource node will inherit the policy from the parent node. In this case, the node-level policy will be merged with the inherited policy, and the two will be reconciled and evaluated.
- The third rule is if a resource node has `inheritfromParent = false` set, the resource node will not inherit the policy from the parent node. If a policy is configured at the resource node level, it will take effect; if no policy is configured, then the default behavior of the policy will be enforced as defined in the first rule.
- The fourth rule is about managing conflicting policies. Let's consider an example where you have a policy constraint defined at the folder level that allows the value `project/123`, and you have another policy constraint at the project level under the folder that denies the value `project/123`. In such circumstances, when a parent policy allows an action and the project level denies the same action, `DENY` will always take precedence. This is how policies are merged. It is also highly recommended as a best practice to not create policy conflicts where one policy contradicts another. This results in confusion about the actual purpose of the policy.

An important note on exceptions that applies to policy merges: for all organization policies that are derived from Boolean constraints, do not merge. For example, say you have defined a policy constraint at the folder level and set enforcement to `true`. Another policy constraint for the project under the folder is set to `false`. In this case, the most specific policy action, `false`, will take effect.

Google Cloud documentation uses the illustration in *Figure 4.8* to explain how policy evaluation works. Let's summarize based on the preceding rules what policy actions are taken on each resource:

Figure 4.8 – Organization policy constraints evaluation criteria

In this example, we see at the **Organization Node** level a red square and a green circle defined. For **Resource 1**, the arrows represent `inheritfromParent = true`; therefore, the first resource will inherit a red square and green circle and add a blue diamond as a node-level resource policy.

Similarly, for **Resource 2**, `inheritfromParent = true`; therefore, it will inherit the policy from the parent but it defines another policy at the resource node level to remove the green circle. This means the reconciled policy will only include the red square.

The next two resources are set to `inheritfromParent = false`; therefore, for **Resource 3**, a yellow circle is added as defined by the node resource policy, and for **Resource 4**, `restoredefault` is defined. This means all default policy constraints will be defined.

This concludes the resource management topic. In the next section, we will cover hierarchical firewall policies and how they are applied and evaluated.

How hierarchical firewall policies work

Although hierarchical firewall policies are out of scope for the exam, they are included here for completeness and awareness. Hierarchical firewall policies are applied in a similar way to how you apply IAM policies and constraints for an organization. You can enforce these policies at the organization root node or folder level. This lets you create a consistent set of network policies across the entire organization.

Hierarchical firewall policies are similar to how VPC firewall rules work, where you can create allow or deny rules. The only difference is that hierarchical firewall rules do allow you to delegate rules by applying a `goto_next` action. Also, lower-level rules cannot override higher-level rules that are defined by the network administrators and are applied organization-wide.

There are some specifications defined for hierarchical firewall policies. Let's understand them, including the evaluation criteria:

- You can only create hierarchical firewall policies at the organization and folder level.
- Creating a policy does not enforce it, unless you apply it at the level you want to, such as at the organization or folder level.
- Rule evaluation is based on the resource hierarchy. All rules at a given level are evaluated before moving to the next level.
- The `goto_next` action lets you delegate the evaluation of the policy to a lower-level policy at the next level.
- Using resource targets such as virtual machine instances, subnets, or specific services such as **Google Kubernetes Engine** (**GKE**) clusters, you can create hierarchical policies for specific VPCs and VMs. This can help you manage exceptions for certain VM groups.
- You can include either an IPv4 or IPv6 address as part of the policy but cannot include both in the same policy.

Next, let's take a look at how policy inheritance works for hierarchical firewall policies.

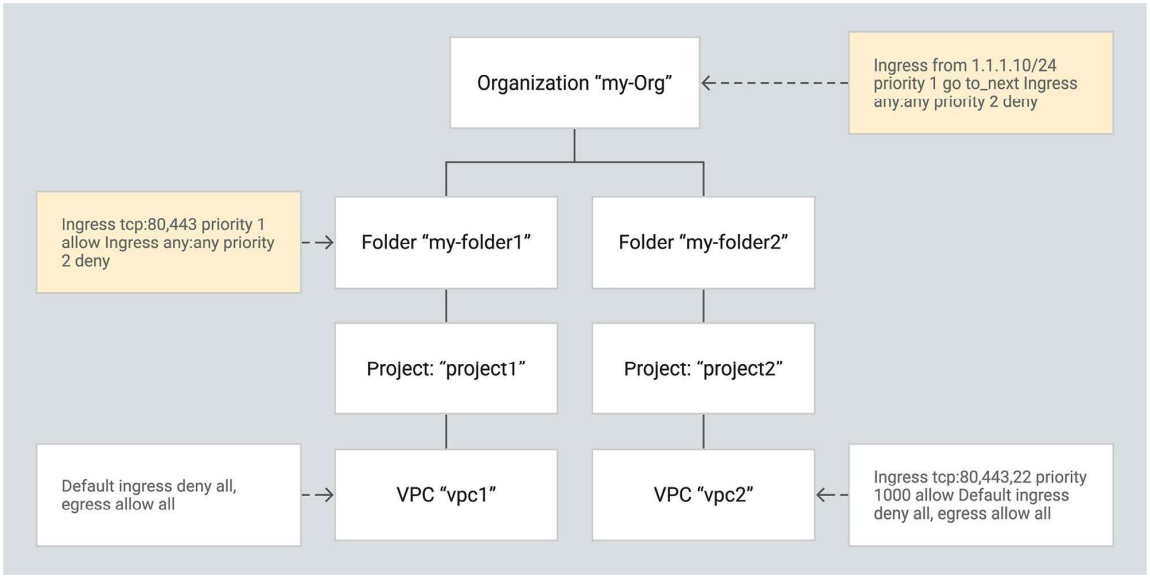

Figure 4.9 – Hierarchical firewall policy inheritance

In *Figure 4.9*, you can apply hierarchical firewall policies at the organization level or folder level. When applied at the **Organization "my-Org"** level, all policies are inherited by the folders and their respective projects and VPCs. In the case where policies are applied only at the folder level, such as the one in the preceding figure applied at the **Folder "my-folder1"** level, these policies will only apply to **project1** and **vpc1**. Also, note that lower-level policies cannot override higher-level policies, as mentioned earlier. However, you have the ability to delegate the policy decision by defining the `goto_next` action for the next lower-level policy to be evaluated. Let's look at how hierarchical firewall policy evaluation works.

The rule evaluation for hierarchical firewall policies is fairly easy and follows three rules.

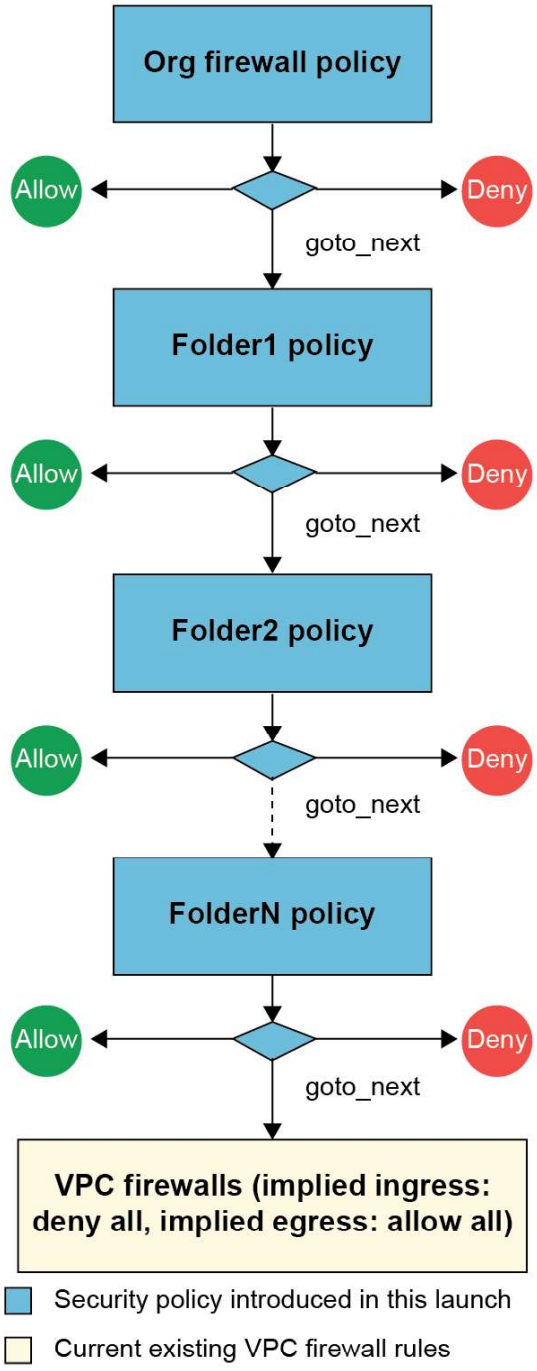

Figure 4.10 – Hierarchical firewall policy evaluation

Let's look at *Figure 4.10* and understand how the rules are applied and evaluated:

1. The first rule is that when you apply the firewall policy at the organization level, all policies are evaluated against an `allow` or `deny` action for a given virtual machine. You can further include the `goto_next` action to delegate the rule to be evaluated by the next lower-level policy.
2. The second rule is when firewall policies are applied at the folder level, the policies are evaluated and then they make their way to the child folders if they exist. Similar to organization policies, you can configure a `goto_next` action for the policy to be evaluated by a child folder.
3. The third rule is unless an `allow` or `deny` action was executed, no action is required. If the `goto_next` action was associated, then the VPC firewall rules will enforce the `allow` or `deny` action to a connection.

In this section, we looked at how hierarchical firewall policies work. Next, we will look at asset management in Google Cloud using the Cloud Asset Inventory service.

Asset management using Cloud Asset Inventory

Cloud Asset Inventory plays a key role in security as it gives you the ability to view your assets in near real time and also to detect the associated changes to the asset. Previously, Cloud Asset Inventory was accessible either via the CLI or Security Command Center, but with recent changes, you can now access Cloud Asset Inventory via the Google Cloud console.

Cloud Asset Inventory is a metadata inventory service that lets you search, export, monitor, and analyze metadata related to supported Google Cloud assets. All metadata information related to an asset is presented on a timeline where you can view historical information (the past five weeks) to get insights into changes to an asset. Before we look at some use cases that Cloud Asset Inventory can help with, let's understand some key concepts.

An asset is a Google Cloud resource or a policy object. Cloud Asset Inventory collects metadata about resources, such as for Compute Engine, storage buckets of App Engine, and so on. Policy-related metadata includes IAM policies, organization constraint policies, and Access Context Manager policies. The third type of metadata information is collected for runtime information such as OS inventory, which includes the operating system and installed and available packages for a given Compute Engine virtual machine.

> **Note**
> You can view the full list of all supported assets at `https://packt.link/GWLd3`.

Now let's look at some of the key attributes and features of Cloud Asset Inventory.

Asset search

The key advantage of Cloud Asset Inventory is its features that can help you search, export, monitor, and analyze information on different assets and provide useful insights. Let's take a look at each one of them.

With the Cloud Asset Inventory API, you can create a custom query to find IAM policies at the organization, folder, or project level. The following command is an example that you can use to list all IAM policies:

```
gcloud asset search-all-iam-policies \
  --scope=projects/12345678
```

You can detect risky policy settings using the Cloud Asset Inventory API by looking for service accounts that have an owner role. The following command is an example of such a query:

```
gcloud asset search-all-iam-policies \
  --scope=organizations/123456 \
  --query='policy roles/owner serviceAccoun')' \
  --page-size=50 \
  --flatte'='policy.bindings[].members']' \
  --forma'='table(resource.segment(3):label=RESOURCE_TYPE, resource.
basename():label=RESOURCE, policy.bindings.member')' \
  | grep serviceAccount
```

Using the Cloud Asset Inventory search function can help you answer questions such as these:

- What IAM policies contain `"foo@bar.com"`?
- Which resources are open to the world (containing `"allUsers"`)?
- Who has an owner role in my organization?
- Are there any `gmail.com` users with `*.setIamPolicy` permissions?

Besides searching policies, you can also search for resources, such as listing all the resources within your project, finding VMs located in the EU region, or finding BigQuery datasets labeled as sensitive. All these capabilities give you powerful access to reduce the time it takes and retrieve information to take corrective action. Next, let's take a look at how you can export asset-related information.

Asset export

With asset export, you can export all resources and policies using the Cloud Asset Inventory API. Using the `ExportAssets` API, you can export all assets at a given timestamp to a Cloud Storage file or BigQuery table. Further, you can use the `ListAssets` API to list all assets at a given timestamp and can use the `BatchGetAssetsHistory` API to retrieve change histories of specific assets for a given timeframe. Cloud Asset Inventory provides time series metadata on resources, IAM policies, organization policies, and access policies. A five-week history is available to export.

Asset monitoring

You can monitor your Google Cloud assets by using the real-time notification APIs, which can create feeds to receive asset changes in real time through Cloud Pub/Sub. This allows you to select interesting changes by specifying types, names, and **Common Expression Library** (**CEL**) conditions upon any data field. For example, in order to detect whether an `@gmail` account has been newly added to in IAM policy, you can use the following:

```
temporal_asset.asset.iam_policy.bindings.filter(b, b.members.exists(m,
endsWith("@gmail.com"))).size() > temporal_asset.prior_asset.iam_
policy.bindings.filter(b, b.members.exists(m, endsWith("@gmail.
com"))).size()
```

This provides a continuous monitoring capability for sensitive resources and policies such as firewalls and IAM policies.

Asset analyzer

Using a policy analyzer, you can get more details on the impact of policies on your cloud resources. You can answer questions such as *Who can access what and why?* You can find out who can read sensitive data from a BigQuery dataset or who has access to a Google Cloud Storage bucket and at what time.

This is an extremely useful and powerful capability to help you gain control and visibility of your Google Cloud environment. Another useful feature is **Asset Insights**, which can give you details on IAM policies that are applied to your resources and detect any risky policies so you can take corrective action.

Another key update to Cloud Asset Inventory, which is available now for use, is **Asset Relationships**. Some assets are related to each other; for instance, a web server may be connected via a firewall rule to access a MySQL database. In situations where a network administrator modifies a firewall rule that breaks the connectivity between the web application and the database, you can establish a relationship based on policies and resources to help you troubleshoot the issue and identify who (the user) did what (the action) to a resource(s) that led to an incident.

Cloud Asset Inventory is not part of the exam blueprint, but you may see indirect references to assets and policies, and it will benefit you as a Google Cloud Security engineer to have a foundational understanding of how asset management is done on Google Cloud and, more importantly, its relevance to security.

We will next look at some of the best practices and design considerations that you should know about and how they apply to the resources in your project.

Best practices and design considerations

Some of the design considerations are to understand how the resources will be managed inside the project. Using one project might be a good idea to keep it simple, but the isolation and separation of duties will not be achieved. On the flip side, if you use too many projects, there will be a lot of overhead to manage the projects, but you will achieve the separation of duties and the isolation required.

Some of the design considerations to follow when breaking down resources and workloads into projects are as follows. Bear in mind that all considerations are correlated:

- You don't want a misconfiguration or a compromise in one operating environment to impact the other. A key consideration is how to reduce the blast radius.
- Quotas and limits are applied at the project level. It's undesirable for a dev/test project to consume the quota required by a prod project, or that one app consumes the quota of another.
- You want to break down cloud resources in a way that will ensure ease of IAM management to achieve least-privilege access. A key consideration is reducing administrative complexity.
- It's easier to track billing at the project level.
- Ensure the separation of duties. For example, you might want to separate central networking projects from app projects to achieve the separation of duties between networking teams and development teams.
- You want to achieve all the above for different operating environments by separating accordingly (for example, prod, test, dev, staging, and user acceptance testing).

The preceding considerations help us understand how and what to consider when breaking down resources and workloads in projects. Further to this, some other key considerations for managing projects are having one project per application, using data classification as a way to keep sensitive data in separate projects and not mixing different data classifications, and finally, using consistent project ID naming conventions.

The best practice is to use workload- and environment-specific projects, and as project creation is a repetitive task, Infrastructure as Code and automation should be leveraged.

In terms of folder design considerations, try not to replicate your organization structure. Many examples cited here use that construct for the purpose of illustration. The key is to understand the workflow and access pattern. When designing your folder structure, the key questions to ask are these:

- What is the access pattern today?
- Does an entire team that works on an app need access to all of its components?
- Do you have separation of access policies between different operating environments?

This will help you understand how to organize your folder structure based on workflow and access control that fits your organization's needs.

There is almost never a need for multiple organizations. Usually, an organization can lump all of its data that's subject to compliance policies into a single folder structure to separate it from all other projects and ensure that the organization admins are trained in the handling of all types of compliant information.

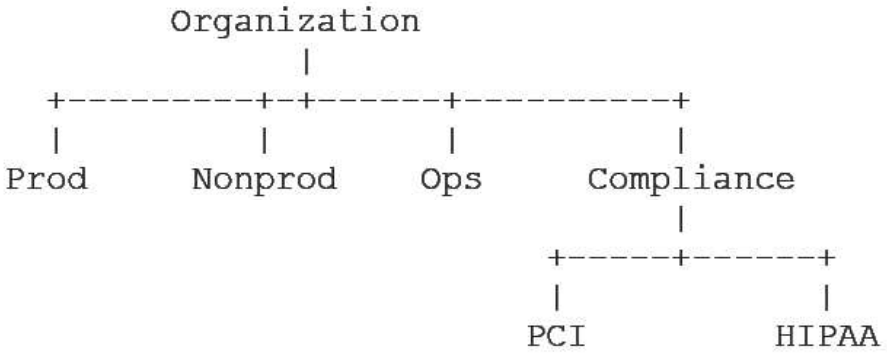

Figure 4.11 – Example organization structure

Let's consider this example: if you have to create multiple organizations, you can use folders to group projects based on business units, development environments, and teams to share common IAM permissions and organizational policies. The IAM permissions will use the inheritance model, and for organizational policies, you can make exceptions as required per project.

Use multiple organizations only if multiple teams handle data subject to mutually exclusive compliance policies (for example, **HIPAA** data needs to be separated from **PCI** data). Some key design considerations for managing multiple organizations include the following:

- Each organization should have its own Cloud Identity account and will need admin accounts provisioned and managed independently
- It requires managing multiple Cloud Identity directories (users, groups, domains)
- IAM policies should be decentralized
- Organization-level policies will need to be duplicated

- Any changes/updates to policies will need to be appropriated and applied to each organization
- Organization-level dashboards will only provide a view of their own environment
- Environments are separated by organizations
- Combining environments or restructuring is very difficult
- Projects cannot be easily moved between organizations
- Shared VPC can only be used to share network resources within the same organization
- Future org-wide GCP features may not be able to be fully leveraged or may require workarounds to apply to all environments

It's important to keep these considerations in mind when looking to deploy multiple organizations and avoid complexity. The best practice, wherever possible, is to use folders to separate business units.

Summary

In this chapter, we learned how to organize our Google Cloud infrastructure and manage our available resources. We discussed the basics of creating folders and projects, as well as the features available to you if you want to enforce organizational policy constraints on them. If you're planning on using IAM or firewall rules in your infrastructure, we also went over how policy inheritance works and how it might aid in your organizational structure. We wrapped up by reviewing Cloud Asset Inventory and some best practices for managing your resources.

In the next chapter, we will do a deep dive into Cloud Identity.

Further reading

For more information on GCP compliance, refer to the following links:

- Resource Manager how-to guides: `https://packt.link/RtxyV`
- Best Practices: GCP Resource Organization and Access Management (Cloud Next'19): `https://packt.link/Z35Ox`
- Creating and managing organization policies: `https://packt.link/OegQi`
- Cloud Asset Inventory how-to guides: `https://packt.link/MexmV`
- Understanding Cloud Asset Inventory: `https://packt.link/L91Io`