Audit records begin their lifecycle inside the `kube-apiserver` component. Each request on each stage of its execution generates an audit event, which is then pre-processed according to a certain policy and written to a backend. The policy determines what's recorded and the backends persist the records. The current backend implementations include log files and webhooks.

The Kubernetes documentation goes on to say that each request can be recorded with an associated stage. The defined stages are as follows:

- `RequestReceived`: The stage for events generated as soon as the audit handler receives the request, and before it is delegated down the handler chain.
- `ResponseStarted`: Once the response headers are sent, but before the response body is sent. This stage is only generated for long-running requests (for example, watch).
- `ResponseComplete`: The response body has been completed and no more bytes will be sent.
- `Panic`: Events generated when a panic occurred.

The Kubernetes audit policy defines rules about what events should be recorded and what data they should include. GKE receives the log entries from the Kubernetes API server, and it applies its own policy to determine which entries get written to the project's admin logs. For the most part, GKE applies the following rules to log entries that come from the Kubernetes API server:

- Entries that represent `create`, `delete`, and `update` requests go to the Admin Activity log
- Entries that represent `get`, `list`, and `updateStatus` requests go to the Data Access log

Admin Activity logs are turned on by default in your Google Cloud project. They can't be turned off; however, you must turn on Data Access logs. Let us move on to understanding the features of GKE logging.

Logging

By default, GKE clusters are natively integrated with Cloud Logging (and Monitoring). When you create a GKE cluster, both Monitoring and Cloud Logging are enabled by default. That means you get a monitoring dashboard specifically tailored for Kubernetes and your logs are sent to Cloud Logging's dedicated, persistent datastore and indexed for both searches and visualization in the Cloud Logs Viewer.

There are several ways to access your logs in Cloud Logging depending on your use case.

You can access your logs using the following:

- **Cloud Logging console**: You can see your logs directly from the Cloud Logging console by using the appropriate logging filters to select Kubernetes resources such as cluster, node, namespace, pod, or container logs. Here are some sample Kubernetes-related queries to help get you started.

- **GKE console**: In the **Kubernetes Engine** section of the Google Cloud console, select the Kubernetes resources listed in **Workloads**, and then the **Container** or **Audit Logs** link.
- **Monitoring console**: In the **Kubernetes Engine** section of the Monitoring console, select the appropriate cluster, nodes, pods, or containers to view the associated logs.
- **gcloud command line tool**: Using the `gcloud logging read` command, select the appropriate cluster, node, pod, and container logs.

These are all the methods of logging that you should be conversant with. Here are some best practices for containerized applications when it comes to logging:

- Use the native logging mechanisms of containers to write the logs to `stdout` and `stderr`.
- If your application cannot be easily configured to write logs to `stdout` and `stderr`, you can use a sidecar pattern for logging.
- Log directly with structured logging with different fields. You can then search your logs more effectively based on those fields.
- Use severities for better filtering and reducing noise. By default, logs written to the standard output are on the `INFO` level and logs written to the standard error are on the `ERROR` level. Structured logs with a JSON payload can include a severity field, which defines the log's severity.
- Use the links to the logs directly from the **Kubernetes Engine** section of the Cloud console for containers, which makes it quick to find the logs corresponding to the container.

Now that we have auditing and logging out of the way, let us move on to understand how GKE does network isolation.

Network Policies

You use Kubernetes NetworkPolicies for certain applications in your cluster if you wish to manage traffic flow at the IP address or port level (OSI layer 3 or 4). **Network policies** are an application-centric construct that allows you to declare how a pod can communicate across the network with various network *entities* (we use the term *entity* here to avoid overusing terms like *endpoints* and *services*, which have unique Kubernetes implications). Other connections are not affected by NetworkPolicies since they have a pod on one or both ends.

To manage communication between your cluster's pods and services, you can utilize GKE's network policy enforcement. Pod-level firewall rules are created using the Kubernetes Network Policy API to construct a network policy. These firewall rules control which pods and services in your cluster can communicate with one another.

When your cluster is hosting a multi-level application, defining a network policy allows you to enable things such as defense in depth. For example, you can set up a network policy to prevent a compromised frontend service from communicating directly with a billing or accounting server many tiers down.

A network policy can also help your application host data from several users at the same time. You can, for example, define a tenant-per-namespace paradigm to provide secure multi-tenancy. Network policy rules can ensure that pods and services in one namespace cannot access pods and services in another namespace in such a model.

Another form of network isolation is by using a feature called GKE private clusters, whereby you can host a private cluster, that is, a cluster with no public IP address. Let us understand this.

GKE private clusters

In private clusters, nodes only have internal IP addresses to isolate nodes from inbound and outbound connectivity to the internet.

Worker nodes in a private cluster only have internal IP addresses. The control plane, on the other hand, has both a private and a public destination. When creating a private cluster, you can disable access to the public endpoint. An internal load balancer provides access to the private endpoint. If you wish to reach the control plane from a GCP region other than where it is deployed, you will need to allow global access. Let us look at the architecture of private clusters now:

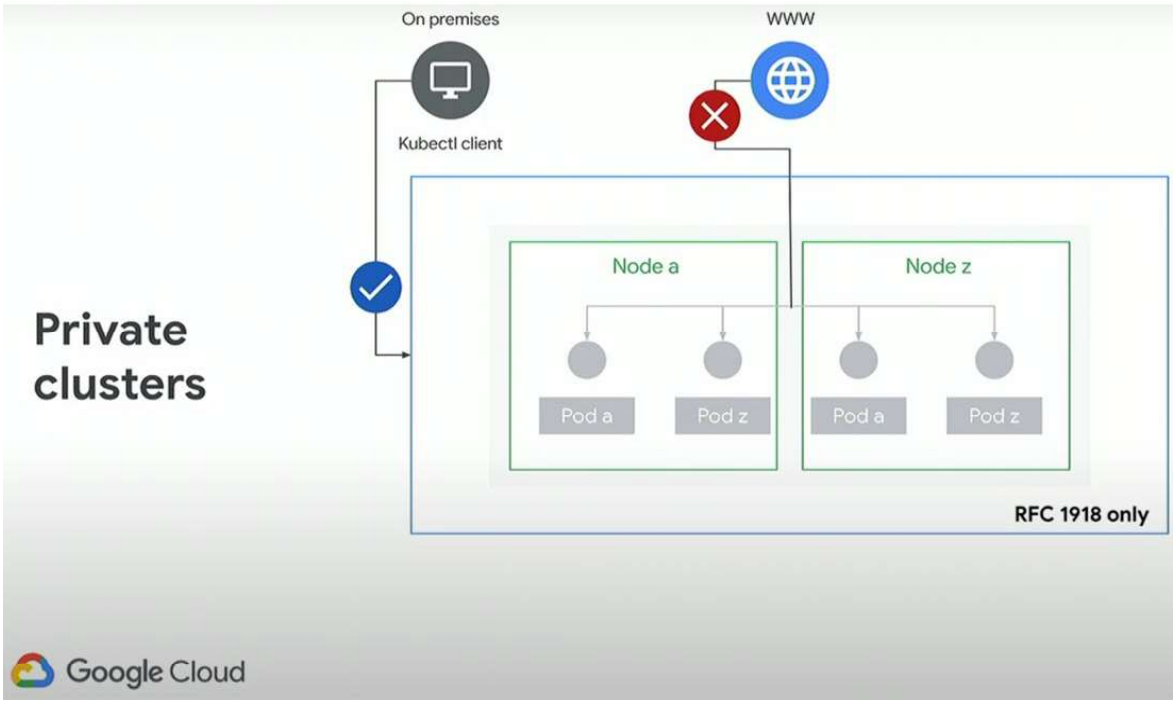

Figure 15.4 – Private clusters

As seen in *Figure 15.4*, using a private cluster has the additional security benefit that nodes are not exposed to the internet. As with public clusters, you can also use GKE's authorized networks feature with private clusters to restrict access to the master API.

A service mesh is an increasingly popular method of implementing network policies for GKE. We will understand that now.

Service mesh

A service mesh is a Layer 7 proxy. Microservices can use this service mesh to abstract the network. This effectively abstracts how inter-process and service-to-service communications are handled in the K8S cluster. In Kubernetes, the service mesh is usually implemented as a set of network proxies. These proxies, which are deployed as a *sidecar* of an application, serve as an entry point for service mesh functionality and manage communication between microservices. Istio is a popular open service mesh that connects, manages, and secures microservices in a consistent manner. It allows you to manage traffic flows between services, enforce access controls, and aggregate telemetry data without modifying the microservice code. The following are the benefits that Istio offers:

- Automatic load balancing for HTTP, gRPC, WebSocket, MongoDB, and TCP traffic
- Fine-grained control of traffic behavior with rich routing rules, retries, failovers, and fault injection
- A configurable policy layer and API that supports access controls, rate limits, and quotas
- Automatic metrics, logs, and traces for all traffic within a cluster, including cluster ingress and egress
- Secure service-to-service communication in a cluster with strong identity-based authentication and authorization
- You configure Istio access control, routing rules, and so on by using a custom Kubernetes API, either via kubectl or the Istio command-line tool istioctl, which provides extra validation

Google recommends using Anthos Service Mesh, Google's full-supported distribution of Istio.

When you create or update a cluster with Istio on GKE, the following core Istio components are installed:

- The *istiod* control plane, which provides the following:
  - Traffic management
  - Security
  - Observability
- The Istio ingress gateway, which provides an ingress point for traffic from outside the cluster

The installation also lets you add the Istio sidecar proxy to your service workloads, allowing them to communicate with the control plane and join the Istio mesh.

So far, we have understood the GKE security features. Now, we will look at how to secure a container image, which can be a source of vulnerability.

Container image security

Container images are *baked* by a pipeline, a series of steps that add the required components on top of each other. The application is deployed on the very top, as a last step. The following figure shows the process of a container pipeline published by NIST.

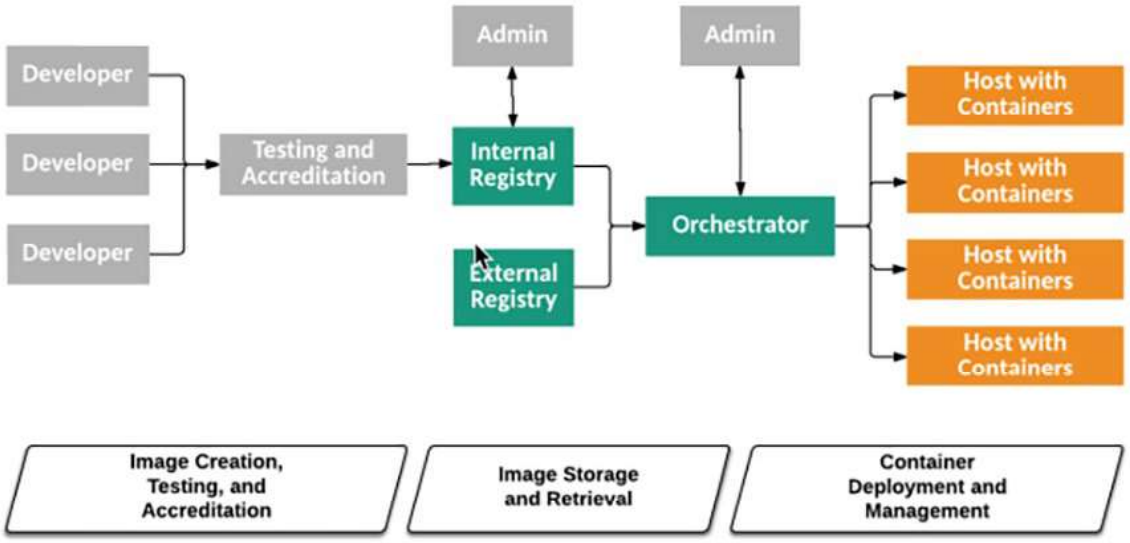

Figure 15.5 – Container pipeline

In *Figure 15.5*, as recommended by NIST (`https://packt.link/luNhB`), the container pipeline follows a controlled process for image generation:

1. Developer systems generate images and send them for testing and accreditation.
2. Testing and accreditation systems validate and verify the contents of images, sign images, and send images to the registry.
3. Registries store images and distribute images to the orchestrator upon request.
4. Orchestrators convert images into containers and deploy containers to hosts.
5. Hosts run and stop containers as directed by the orchestrator.

Now that you understand what the pipeline does, here are some best practices for container image security as recommended by Google:

- Use Google-maintained base images
- Do not rely on packages in third-party repositories (if possible, host an internal repository of the packages that are scanned for vulnerability)
- Do not include secrets in images

- Scan images for vulnerabilities
- Stop the deployment of the container if the images do not pass attestations (see the *Binary Authorization* section later in the chapter)

Now that you understand the best practices for image security, let us see how to scan an image for vulnerabilities on Google Cloud.

Container vulnerability scanning on Google Cloud

Container scanning is a service on Google Cloud that identifies known vulnerabilities in Debian, Ubuntu, Alpine, Red Hat, and CentOS packages inside the container on GCP. This feature can be enabled in Artifact Registry.

There are two types of scans supported by container scanning:

- **On-demand scanning**: Using the gcloud CLI, you may scan container images locally on your PC or in your registry on demand. This gives you the freedom to tailor your CI/CD pipeline to your specific needs, for example, when you need to obtain vulnerability results.
- **Automated scanning**: Container Analysis searches container images in Artifact Registry and Container Registry for vulnerabilities and keeps the vulnerability information up to date. Scanning and continuous analysis are the two major jobs in this technique.

When new images are uploaded to Artifact Registry or Container Registry, Container Analysis examines them. This scan extracts information on the container's system packages. Based on the image's digest, the images are only scanned once. This means that adding or changing tags will not result in new scans; only changing the image's content would. Here is what happens when you enable scanning:

- When you upload an image, Container Analysis creates occurrences for vulnerabilities found. It continually checks the information for scanned images in Artifact Registry and Container Registry for additional vulnerabilities after the initial scan.
- Container Analysis changes the metadata of the scanned images as new and updated vulnerability information is received from vulnerability sources, establishing new vulnerability occurrences for fresh images, and eliminating vulnerability occurrences that are no longer valid.
- Container Analysis only updates vulnerability metadata for images pushed or pulled in the previous 30 days. Vulnerability metadata older than 30 days is archived by Container Analysis. To re-scan an image with archived vulnerability metadata, push or pull that image.

You can also utilize **manifest lists** to scan for vulnerabilities. A manifest list is a collection of links to manifest files for various platforms. They allow a single image to work with numerous operating system architectures or variations. Only Linux amd64 images are supported by Container Analysis vulnerability scanning. Please note that if your manifest list points to more than one Linux amd64 image, only the first one will be scanned.

Container scanning uses the following severity levels:

- Critical
- High
- Medium
- Low
- Minimal

There are two types of *severity* associated with a vulnerability:

- **Effective severity**: The Linux distribution's designated severity level. Container scanning employs the severity level supplied by the provider if distribution-specific severity levels are not available.
- **CVSS score**: The **Common Vulnerability Scoring System** (**CVSS**) score and associated severity level, with two scoring versions:
  - **CVSS 2.0**: Available when using the API, the Google Cloud CLI, and the GUI
  - **CVSS 3.1**: Available when using the API and the gcloud CLI

Now that you understand container vulnerability scanning, let us see how to gate your deployment in case there are vulnerabilities or a critical step in the container pipeline is missing.

Binary Authorization

You can utilize Binary Authorization to construct a vulnerability allowlist as part of your Cloud Build workflow based on the vulnerability information provided by Container Analysis. The build will fail if the vulnerabilities contradict the allowlist's policy.

Container Analysis and Binary Authorization can be combined to provide attestations, which can prohibit known security vulnerabilities in container images from running in your production environment. Let us look at a process flow of how Binary Authorization can be incorporated into your pipeline:

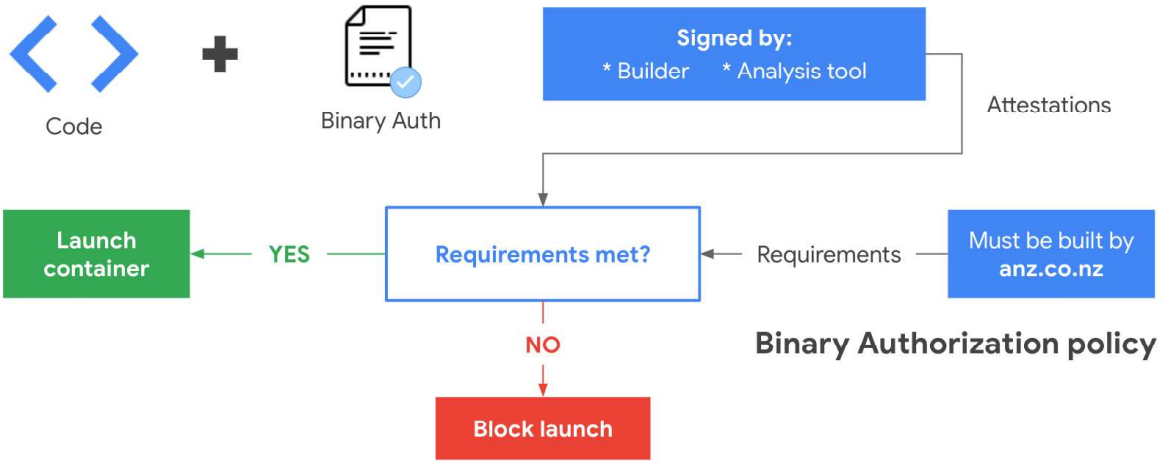

Figure 15.6 – Binary Authorization

As in *Figure 15.6*, Binary Authorization works by doing the following:

- Ensuring that only trusted code is deployed to your environment.
- Enforcing that certain signatures are on images deployed to production.
- Signing the build using Private PGP. **PGP** stands for **Pretty Good Privacy**. The public key is uploaded to the GKE admission controller for verification.

> **Note**
> PGP is a widely used encryption software program that allows users to encrypt and decrypt electronic messages, files, and data. PGP uses a combination of symmetric-key and public-key cryptography to provide strong encryption and digital signatures.

Here are the components that make up Binary Authorization:

- **Policies** govern the deployment of container images to GKE through a set of rules:
  - **Rules** are part of a policy that defines constraints that container images must pass before they can be deployed. Most often, a rule will require one or more digitally signed attestations. An **attestation** is a digitally signed record that signifies that the associated container image was built by the successful execution of a specific, required process. This is a *Continuous Integration* stage artifact.

- A **verified attestation** is a statement by an attestor that a container image is ready for deployment. This is a *Continuous Deployment* stage artifact.
- A **signer** is a person or an automated process that creates an attestation by signing a unique container image descriptor with a private key.
- Binary Authorization uses an **attestor** to verify the attestation at container deploy time with a public key.

Under the hood, Binary Authorization acts as a **Kubernetes admission controller**.

In GKE, this is implemented with the pod creation API, using the `ImagePolicy` admission controller. Binary Authorization supports using breakglass to override an authorization policy.

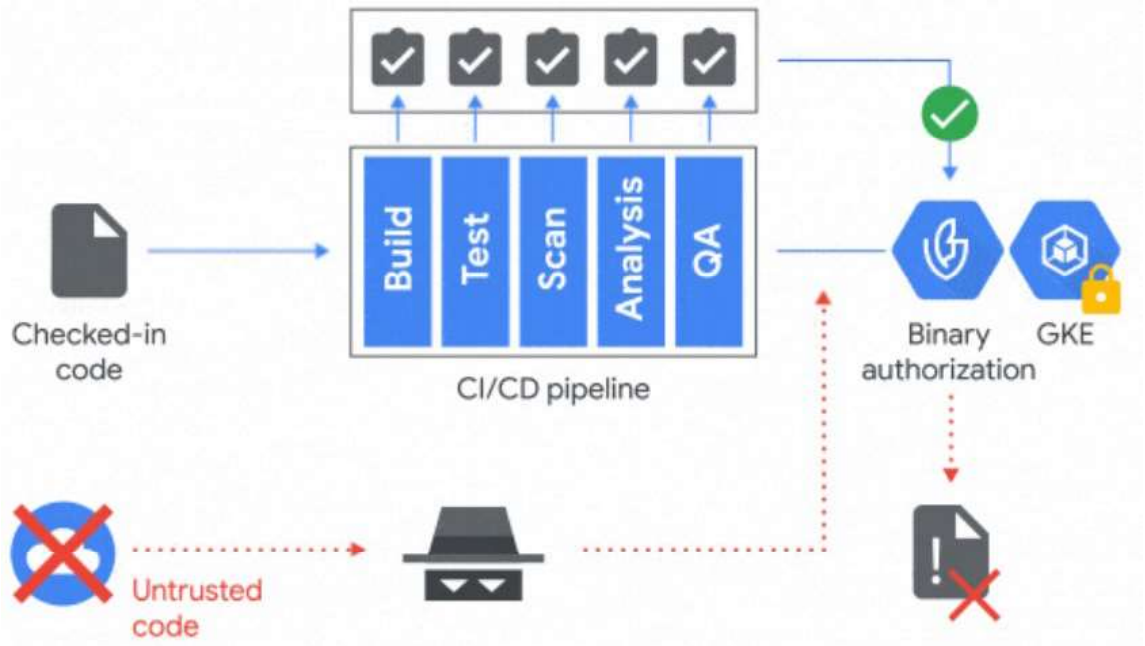

Figure 15.7 – Binary Authorization in the CI/CD pipeline

In *Figure 15.7*, you can see how Binary Authorization can stop untrusted code from deploying in the production GKE cluster.

Now that you understand the basics of the container pipeline and what Google Cloud products you can use for image vulnerability and to stop the deployment of untrusted code, let us move on to understand the certificate services of GKE.

Cluster Certificate Authority (CA)

A GKE cluster communicates with various components to keep the cluster healthy. The intracluster communication happens over **mutual TLS** (**mTLS**). Here are various components of the cluster that uses mTLS:

- Control plane to node
- Node to node
- Pod to pod
- etcd to etcd (multiple instances of etcd)
- Control plane to etcd

Now let us understand the various components of the GKE **Public Key Infrastructure** (**PKI**) system that help with mTLS. We will start with the cluster root CA.

The cluster root CA

In GKE, the cluster root CA is used by the API server and `kubelet` to establish trust. Each cluster has its own CA, so if one CA is compromised, it doesn't affect other clusters. The API server and kubelet's client certificates are verified by the cluster root CA, ensuring a shared trust between control planes and nodes. A non-exportable root key managed by an internal Google service handles the CA and signs certificate requests from `kubelet`. Even if the API server is compromised, the CA remains secure, safeguarding other clusters. Additionally, a separate per-cluster etcd CA is used to validate etcd's certificates.

In Kubernetes, there is an API called `certificates.k8s.io` that allows you to create TLS certificates that are signed by your own CA (if required). You can then use these certificates and CA to establish trust in your workloads.

Let us understand the role of the API server and kubelet now.

The API server and kubelet

The API server and kubelet rely on the cluster root CA for trust. In GKE, the control plane API certificate is signed by the cluster root CA. Each cluster runs its own CA so that if one cluster's CA is compromised, no other cluster's CA is affected. Let us see some more details here:

- An internal Google service manages root keys for this CA, which are non-exportable. This service accepts certificate signing requests, including those from the kubelets in each GKE cluster. Even if the API server in a cluster were compromised, the CA would not be compromised, so no other clusters would be affected.

- Each node in the cluster is injected with a shared secret at creation, which it can use to submit certificate signing requests to the cluster root CA and obtain kubelet client certificates. These certificates are then used by the kubelet to authenticate its requests to the API server. This shared secret is reachable by pods on the node, unless you enable Shielded GKE nodes, Workload Identity, or metadata concealment.
- The API server and kubelet certificates are valid for five years but can be manually rotated sooner by performing credential rotation.

Let us understand etcd certificate exchange now.

etcd

In GKE, etcd relies on a separate per-cluster etcd certifying authority for trust. Let us see how etcd makes a cluster safe:

- Root keys for the etcd CA are distributed to the metadata of each VM on which the control plane runs.
- Any code executing on control plane VMs, or with access to compute metadata for these VMs, can sign certificates as this CA. Even if etcd in a cluster were compromised, the CA would not be shared between clusters, so no other clusters would be affected.
- The etcd certificates are valid for five years.

It is recommended that you perform certificate rotation to rotate all your cluster's API server and kubelet certificates. There is no need for you to initiate certificate rotation in etcd; this is handled by GKE.

Now that you understand GKE requirements and methods for the certificates, let us move on to understand the Workload Identity feature of GKE. This is different from the service account Workload Identity you studied earlier.

GKE Workload Identity

Workload Identity is an identity federation mechanism that allows K8s workloads to securely access Google Cloud services. On Autopilot clusters, Workload Identity is enabled by default. Workload Identity maps a pod to a **K8s Service account** (**KSA**) and KSA to an IAM service account.

Without using Workload Identity, this is what you need to do to access a Google Cloud service:

1. Create an IAM service account.
2. Create keys for a service account.
3. Import service account keys as a K8s secret.

On a K8s workload, you would usually follow the following steps for your application (running on a pod) to communicate with Google Cloud services, for example, Cloud Storage:

1. Define a volume with the secret.
2. Mount the volume inside the container.
3. Point `$GOOGLE_APPLICATION_CREDENTIALS` at the key file.
4. The workload can now authenticate to Google Cloud APIs as that service account.

This is toilsome to set up and hard to secure. A better option is to use Workload Identity. With Workload Identity, you can do the following:

1. Enable Workload Identity for a GKE cluster.
2. Run a workload using a dedicated KSA.
3. Grant KSA access to desired Google Cloud resources using IAM roles.
4. The workload can now access Google Cloud Storage APIs by presenting (short-lived, auto-rotated) KSA tokens.

You can see Workload Identity removes a lot of steps and makes your GKE workloads more secure than when using service account keys.

So far, we have seen various security features of GKE. Now let us look at the best practices recommended by CIS.

Center for Internet Security (CIS) best practices

Here are some security best practices to harden your GKE cluster security. These are presented in detail in the Google Cloud documentation as well as CIS GKE benchmarks, so make sure to understand them:

- Upgrade your GKE infrastructure in a timely fashion
- Restrict network access to the control plane and nodes
- Consider managing Kubernetes RBAC users with Google Groups for RBAC
- Enable Shielded GKE nodes
- Choose a hardened node image with the containerd runtime
- Enable Workload Identity
- Harden workload isolation with GKE Sandbox
- Enable security bulletin notifications
- Use least-privilege Google service accounts
- Restrict access to cluster API discovery

- Use namespaces and RBAC to restrict access to cluster resources
- Restrict traffic among pods with a network policy
- Consider encrypting Kubernetes secrets using keys managed in Cloud KMS
- Use admission controllers to enforce policies
- Restrict the ability of workloads to self-modify
- Audit your cluster configurations for deviations from your defined settings
- Ensure legacy Compute Engine instance metadata APIs are disabled and also ensure the GKE metadata server is enabled
- Ensure Basic Authentication using static passwords is disabled
- Ensure authentication using Client Certificates is disabled
- Leave Cloud Logging enabled
- Leave the Kubernetes web UI (dashboard) disabled
- Disable **Attribute-Based Access Control** (**ABAC**), and instead use **RBAC** in GKE
- Do not disable the `DenyServiceExternalIPs` admission controller

Now let us see some best practices for containers in general.

Container security best practices

Here are a few general security best practices when using containers for your application development process. We will look at these in three parts: the build phase, the distribution/deployment phase, and the production/run phase. These go hand in hand with your CI/CD security:

- **Security in the container build phase**: This is where the container build starts:
  - **Source image control**: In this phase, you write the code to create a container. Follow these best practices for this phase:
    i. Avoid retrieving/using source images with unknown/untrusted publishers.
    ii. If a third-party image is used, it is strongly recommended to identify and document detailed information, such as the version/build of code included and information about the creator of the image.
    iii. Never include code from unverified/untrusted sources in an image.
    iv. Use digital signature/checksum verification services on images whenever possible.
    v. Prior to completing image creation, all dependencies and libraries in source images should contain the latest security updates.

vi. Less is more—include only core, necessary software/code in a base image. This alleviates future work and maintenance load from a security perspective.

vii. Never include code from unverified/untrusted sources in an image.

- **Security/vulnerability scanning**: In this phase, you have the container image ready. Follow these best practices for scanning containers:

i. It is recommended to perform security and vulnerability scanning on images regularly. Most security scanning services can be integrated with the container environment and batched to automatically perform security scanning.

ii. It is recommended to leverage security scanning tools. Clair is an open source security scanner developed by CoreOS and is widely used for container security scanning. There are other commercial security scanners available on the market as well.

iii. Use both static scanning (for example, package integrity verification) and dynamic scanning (for example, binary-level code analysis and/or application behavior analysis) in the security scanning program.

- **Security in the container distribution phase**: In this phase, your container is scanned and ready for deployment. Follow these best practices:

- **Container registry security**:

i. Access to the registry should go through proper authentication and authorization processes.

ii. Always connect to the registry through securely encrypted channels. Images usually contain confidential business components and applications—in some cases, they even contain user data. Connecting to the registry through secure channels can ensure data confidentiality and avoid data leakage through successful man-in-the-middle attacks.

iii. Data security protection mechanisms offered by the registry provider should be closely reviewed and evaluated prior to container deployment.

- **Version control**:

iv. Proper tooling, such as change management, audit, and continuous delivery and integration tools are critical for image version control. Good version control tools are essential in urgent situations, such as when an emergency rollback is needed.

- **Security in the container run phase**: Now we are ready to deploy the container on GKE. Follow these best practices for production:
  - **Authentication/account management**: Here are the best practices when it comes to authentication:
    i. Access to the container runtime environment should go through proper authentication and authorization processes.
    ii. Many authentication issues happen when users' credentials are poorly managed (for example, username/password leakage through social engineering or phishing, or credentials are not properly terminated/transferred when a user leaves the organization). We recommend using the organization's account management system for container account management/authentication/authorization and performing regular audits and account privilege reviews.
    iii. Use RBAC to regulate access to container resources. The least privilege principle should be applied.
  - **Attack surface reduction**: Here are best practices on production clusters to minimize the risk of compromising your cluster:
    i. To minimize the risk of being targeted and attacked, critical features in container images, such as root access and kernel capabilities, should be activated/enabled only when needed under monitoring.
    ii. As part of the best practices, grant cluster-level permissions only to trusted administrators and restrict access based on the principle of least privilege.
  - **Patch management**: It's recommended to keep your images up to date with the latest version. Here are some best practices:
    i. Subscribe to relevant system/library update feeds and apply the latest patches to container images.
    ii. Use version control features from the container registry (or other tools) to maintain an audit trail of container image updates. Critical information, such as timestamps, patch details, and known issues (such as side effects) should be documented for compliance purposes.

- **Secret management**: Secret management is one of the largest attack planes. Follow these best practices to minimize the risk due to secret leaks:

  i. Secrets, or sensitive data, should not be stored within an image. They should be stored outside of the image and be called dynamically at runtime.

  ii. Major orchestration platforms, such as Kubernetes and Docker Swarm, provide native secret management features. Configure the settings properly so that the sensitive data is handled securely when it is stored in a container and when it is transmitted between different services.

  iii. If the organization has an existing secret management system, it should be integrated with the container environment by providing secrets from the non-container environment, through a secure/encrypted channel.

This concludes the best practices for container pipelines, from building to running.

Summary

We covered a lot of ground in this and the previous chapter when it comes to container security. We went over the basics of CI/CD security and then we understood what containers are. Kubernetes paved the way for modernizing applications. The deployments that used to take hours are now deployed in minutes and it's also done several times a day. Kubernetes also makes it easy to scale deployments. GKE is Google's managed offering for Kubernetes, which takes away the pain of management and the complexity. GKE is headed toward more robust, self-healing features. Granted, container and Kubernetes security can be quite challenging to understand, but our hope is that these last two chapters made it easy for you to get a solid foundation.

Congratulations! You have reached the end of this book. We hope you've enjoyed reading it as much as we enjoyed putting it together. We have covered a lot of ground on the different security offerings in Google Cloud, and we hope you feel confident in your knowledge of them. Our ultimate goal is to help you become a certified Google Cloud security engineer and a trusted practitioner in your everyday work. So go out there and ace those exams! Best of luck to you!

Further reading

For more information on container security, refer to the following links:

- A simple IAP proxy for accessing a private GKE master control plane: `https://packt.link/lXbwP`
- Best practices for building containers: `https://packt.link/UUz4G`

Google Professional Cloud Security Engineer Exam – Mock Exam I

1. In the context of Google Cloud's shared responsibility model, which of the following is the responsibility of the customer?

   A. Implementing network security measures
   B. Ensuring the physical security of data centers
   C. Managing Google hypervisor security
   D. Maintaining the server hardware

2. Which compliance standard does Google Cloud adhere to in its commitment to protecting customer data privacy and security?

   A. Health Insurance Portability and Accountability Act (HIPAA)
   B. ISO/IEC 27001
   C. Payment Card Industry Data Security Standard (PCI DSS)
   D. All of the above

3. What is the purpose of Google Cloud Access Transparency?

   A. To provide customers with real-time visibility into the status of their cloud resources
   B. To monitor and track user access and activity within Google Cloud environments
   C. To ensure transparency and accountability by providing customers with logs of Google Cloud's access to their data
   D. To enable seamless integration between Google Cloud and third-party access management systems

4. Which of the following services is supported by Access Approval?

   A. Secret Manager
   B. Identity and Access Management (IAM)
   C. Cloud Key Management Service (KMS)
   D. All of the above

5. Which of the following is a feature of Google Cloud Virtual Private Cloud (VPC)?

   A. Automatic scaling of compute resources
   B. High-performance global load balancing
   C. Managed database services
   D. Serverless function execution

6. Which of the following statements is true regarding Google Cloud VPC auto mode?

   A. In auto mode, you can specify a custom IP address range for your VPC
   B. Subnets in auto mode are manually created by the user
   C. Routing tables in auto mode are managed by the user
   D. Google automatically configures default firewall rules in auto mode

7. Which Google Cloud IAM permissions are required to create a Shared VPC network?

   A. roles/compute.networkAdmin
   B. roles/compute.instanceAdmin
   C. roles/compute.sharedVpcAdmin
   D. roles/iam.serviceAccountAdmin

8. Which of the following statements is true regarding custom route export in Google Cloud VPC?

   A. Custom route export is only available in VPC auto mode
   B. Custom route export allows routes to be advertised from a VPC to other networks
   C. Custom route export requires the use of Cloud Router
   D. Custom route export is limited to a single region within a VPC network

9. Which of the following statements is true regarding the use of service accounts in Google Cloud VPC firewall rules?

    A. Service accounts cannot be used as a source or destination in firewall rules
    B. Service accounts can be used as a source or destination in firewall rules
    C. Service accounts can only be used as a source but not as a destination in firewall rules
    D. Service accounts can only be used as a destination but not as a source in firewall rules

10. Which of the following load balancer types are available in Google Cloud (select all that apply)?

    A. Internal Load Balancer and External Load Balancer
    B. Regional Load Balancer and Global Load Balancer
    C. TCP Load Balancer and HTTP Load Balancer
    D. Network Load Balancer and Application Load Balancer

11. True or false: Private Google Access can be enabled at the subnet level within a VPC network.

    A. True
    B. False

12. Which of the following authentication methods is supported by Google Cloud Identity-Aware Proxy (IAP)?

    A. OAuth 2.0 tokens
    B. Usernames and passwords
    C. Public/private key pairs
    D. Security Assertion Markup Language (SAML)

13. Which of the following scenarios would benefit from using Google Cloud NAT with multiple NAT IP configurations?

    A. A high-traffic web application with distributed users worldwide
    B. A small-scale development environment with limited outgoing traffic
    C. A local network connecting to a single Google Cloud region
    D. A scenario that requires complex load balancing configurations

14. Which of the following statements is true about Google Cloud NAT logs?

    A. NAT logs cannot be stored in Cloud Storage for long-term retention
    B. NAT logs provide detailed information about inbound traffic flows
    C. NAT logs can be exported to third-party logging and monitoring systems
    D. NAT logs are automatically enabled and cannot be disabled

15. Which of the following types of attacks can Google Cloud Armor help protect against?

    A. Denial-of-Service (DoS) attacks
    B. SQL injection attacks
    C. Cross-Site Scripting (XSS) attacks
    D. All of the above

16. Which of the following types of logs are available in Google Cloud Logging?

    A. Performance logs
    B. Network logs
    C. Security logs
    D. All of the above

17. What is the role of a log router in Google Cloud Logging?

    A. To aggregate logs from multiple projects into a single view
    B. To apply filters and transformations to log entries before storage
    C. To manage log retention policies and the archiving of log data
    D. To provide real-time streaming of logs for near-instant analysis

18. Which of the following statements is true regarding custom images in Google Compute Engine?

    A. Custom images can only be created from scratch using the Compute Engine API
    B. Custom images are automatically backed up and replicated across multiple regions
    C. Custom images can be created from existing boot disks or other custom images
    D. Custom images are limited to a maximum size of 10 GB per image

19. Which of the following methods can be used to control access to compute images on Google Cloud?

    A. Setting up IAM roles and permissions for specific users or service accounts
    B. Configuring firewall rules to restrict inbound and outbound traffic to the compute images
    C. Enforcing network tags and labels to regulate access to compute images
    D. Implementing Identity-Aware Proxy (IAP) to authenticate and authorize users accessing the compute images

20. Which of the following is considered a best practice for securing a CI/CD pipeline on Google Cloud?

    A. Regularly updating and patching the CI/CD tools and dependencies
    B. Disabling Multi-Factor Authentication (MFA) for pipeline service accounts to streamline the deployment process
    C. Storing sensitive credentials and secrets in plain text within the pipeline configuration files
    D. Allowing unrestricted access to the CI/CD pipeline from any external IP address

21. Which security concept is relevant for Static Code Analysis (SCA) in a CI/CD pipeline on Google Cloud?

    A. Analyzing code for vulnerabilities and security flaws
    B. Encrypting sensitive data during the code deployment process
    C. Enforcing strong IAM roles for pipeline administrators
    D. Disabling automated testing and quality checks

22. Which security feature helps enhance container runtime security on Google Cloud?

    A. Enforcing container runtime isolation through the use of lightweight Virtual Machines (VMs)
    B. Disabling container sandboxing to improve application performance
    C. Running containers with elevated privileges to ensure system compatibility
    D. Ignoring container image scanning and vulnerability assessment

23. Which statement accurately describes the concept of a container in Google Cloud?

    A. Containers are virtual machines that provide isolated environments for running applications
    B. Containers are lightweight, standalone executable packages that include everything needed to run an application
    C. Containers are graphical user interfaces used to manage virtual machine instances in Google Cloud
    D. Containers are physical servers that are partitioned into multiple isolated environments

24. Which security measure can help mitigate the risk of a container breakout attack in a Kubernetes environment?

    A. Regularly updating the host operating system of the Kubernetes cluster
    B. Disabling network policies to allow unrestricted communication between containers
    C. Running containers with privileged access and unrestricted capabilities
    D. Ignoring container image vulnerabilities and not performing vulnerability scans

25. True or false: Containers running in Kubernetes clusters are automatically isolated from each other by default, preventing unauthorized access between containers.

    A. True
    B. False

26. Which security measure can help mitigate the risk of a container breakout attack in a Kubernetes environment?

    A. Regularly updating the host operating system of the Kubernetes cluster
    B. Disabling network policies to allow unrestricted communication between containers
    C. Running containers with privileged access and unrestricted capabilities
    D. Ignoring container image vulnerabilities and not performing vulnerability scans

27. A customer has several business units that want to use Google Cloud products. Each has its own team that handles development, QA, and product support. The customer wants to give enough independence to the business teams but still wants to control the security aspects of Google Cloud to privilege escalation and avoid security incidents. What is the right approach for the customer to implement such controls?

    A. Set up organization policies, create a project for the business unit, provide the development and QA team owner role to the project, and grant the production support team an Organization admin role so they can manage organization policies.
    B. Set up organization policies, create a folder for the business unit, and provide the ability to create a new project under that folder for the production support team.
    C. Set up organization policies, create a few projects for the business unit, and provide owner permission to all projects to the production support team.
    D. Set up organization policies, create a folder for the business unit, and provision projects for each environment. Only enable services that the team needs in that project. Provide a limited role based on the job responsibilities.

28. A customer needs to implement workloads that comply with PCI and HIPAA regulations. What is the best approach to implement such controls with Google Cloud, provide strict requirements for data segregation, and implement access control among multiple teams while reducing the toil of management?

    A. Implement organization policies and provide access to the team using centralized SSO, create folders and provision projects underneath to segregate the development, QA, and production environments, and grant the development team access to the projects.

    B. Create two Organization Units (OUs) in Cloud Identity, provision users to those OUs based on the requirements, and create multiple SAML profiles.

    C. Create three Google Cloud organizations. One organization will host projects for HIPAA, the second will host projects for PCI, and the third will host all other projects. Provision users and groups for each organization and set up a separate SSO for all three.

    D. Create two Google Cloud organizations. One organization will host projects that do not need strict data and access segregation. Provision users from the same source to both organizations and separate SSO.

29. You are implementing Google Cloud. Your identities are synchronized from your on-premises Active Directory to Azure Active Directory. Your business rules are set in Azure. What is the best approach to provide access to Google Cloud?

    A. Use Google Cloud Identity Sync to provision users and groups from on-premises Active Directory and create SSO using Azure.

    B. Use Azure to provision users and groups into Cloud identity. Create another application in Azure to provide federated access to Google Cloud.

    C. Use Google Cloud Identity Sync to provision users and groups from on-premises Active Directory and use Active Directory Federation Services (ADFS) to provide SSO.

    D. Use Google Cloud Identity Sync to provision users and groups from on-premises Active Directory, use Azure to provision partners to Cloud Identity, and use Azure to provide SSO.

30. A customer needs to rely on their existing user directory and have native authentication when developing solutions in Google Cloud. They want to leverage their existing tooling and functionality to gather insights into user activity using a familiar interface. Which action should you take to meet the customer's requirements?

    A. Provision users to Cloud Identity using Just-in-Time SAML 2.0 user provisioning with the customer user directory as the source

    B. Configure Cloud Identity as a SAML 2.0 service provider, using the customer's user directory as the identity provider

    C. Configure and enforce 2-Step Verification (2SV) in Cloud Identity for all super admins

    D. Configure a third-party identity provider (Okta or PingFederate) to manage authentication

31. A customer wants to use Cloud Identity as their primary identity provider. The customer wants to use other non-Google Cloud SaaS products for CRM, messaging, and customer ticketing management. The customer also wants to improve employee experience with Single Sign-On (SSO) capabilities to securely access Google Cloud and non-Google Cloud applications. Only authorized individuals should be able to access these third-party applications. What action should the customer take to meet these requirements?

    A. Remove the employee from Cloud Identity, set the correct license for the individuals, and resync them to Cloud Identity for the changes to take effect

    B. Configure third-party applications to federate authentication and authorization to the Google Cloud identity provider

    C. Remove the individuals from the third-party applications, add the license to Cloud Identity, and resync the individuals back to the third-party applications

    D. Copy user personas from Cloud Identity to all third-party applications for the domain

32. All of the following are best practices for super admin accounts except:

    A. Create a dedicated super admin account and store it away in vault systems. Use this account only when break-glass access is required.

    B. Delegate the ability to manage Google Cloud organization resources to other users in the organization and assign fundamental Cloud IAM roles to ensure the separation of duties.

    C. Make sure you require Google Cloud super admins to use your organization's SSO system so their activities can be monitored for auditing.

    D. Make sure you have multiple super admins in your organization.

33. All of the following are best practices while setting up Multi-Factor Authentication (MFA) in a Google Cloud organization except (select all that apply):

    A. All users are to turn on 2-Step Verification (2SV)

    B. Do not turn on enforcement of 2SV since users may lock themselves out

    C. Allow new users enough time to enroll

    D. Allow users to trust the device

    E. Turn on any 2SV method

    F. Allow security codes with remote access

34. The data team needs to use service accounts to interact with BigQuery. Their project requires a Cloud Storage bucket that holds data feeds. The project transforms the data in the data feeds to BigQuery. What is the right approach for the data team to implement this?

    A. Grant each member of the data team access to Cloud Storage buckets and BigQuery datasets.

    B. Create a Google group with all developers of the data team and grant the Cloud Storage bucket and BigQuery Dataset access to the Google group.

    C. Create a Google group with all data team developers. Assign the group the IAM role of Service Account User.

    D. Create a Google group with all developers. Assign the group the IAM role of Service Account Admin and have developers generate and download their own keys.

35. A three-tier application is running on Google Cloud. The application's middle tier is running on Compute Engine and needs read access to a dataset in Google BigQuery. How should you grant access?

    A. Create a service account for the application and grant the BigQuery User role at the project level

    B. Create a service account for the application and grant the BigQuery User role at the dataset level

    C. Create a service account for the application and grant the BigQuery Admin role at the dataset level

36. A cloud development team needs to use service accounts extensively in their local development. You need to provide the team with the keys for these service accounts. You want to follow Google-recommended practices. What should you do?

    A. Implement a daily key rotation process that generates a new key and commits it to the source code repository every day.

    B. Implement a daily key rotation process and provide developers with a Cloud Storage bucket from which they can download the new key every day.

    C. Create a Google group with all developers. Assign the group the IAM role of Service Account User and have developers generate and download their own keys.

    D. Create a Google group with all developers. Assign the group the IAM role of Service Account Admin and have developers generate and download their own keys.

37. A hybrid application is running on AWS and Google Cloud. The application part on AWS needs to query Google BigQuery. How do you implement this approach securely?

    A. Create a service account for the application, and grant the BigQuery User role at the dataset level. The team managing the AWS side will have ownership of the keys that they will provision with the application.

    B. Create a workforce pool and an AWS workforce provider, and grant impersonation access to the AWS account to the service account at the project level.

    C. Create a workload pool and an AWS workload provider, and grant impersonation access to the AWS account to the service account for the BigQuery dataset.

    D. Create a workload pool and an AWS workload provider, and grant impersonation access to the AWS role to the service account for the BigQuery dataset.

38. You are implementing a control whereby the SRE team needs to get access to Google Cloud only when the condition arises. How do you implement this approach with the least privilege?

    A. Create a Google group for the SRE team, and grant access to the SRE team at the organization level with a condition that allows access only between 8:00 AM and 5:00 PM and with an on-premises IP range.

    B. Create a group for the SRE team, and grant access to the SRE team at the project level with a condition that allows access only between 8:00 AM and 5:00 PM.

    C. Create a group for the SRE team; pre-approve them for access to this group. Ask them to request access to this group, stating the amount of time in minutes that they need access to the Google Cloud project.

    D. Create a group for the SRE team and add the users to this group. Ask them to request access for the amount of time in minutes that they need access to the Google Cloud project.

39. A Google Cloud customer wants to implement segregation of duties. The SRE team should have the ability to generate the keys but shouldn't have the ability to use the key to encrypt or decrypt the data. What is the right method to achieve this?

    A. The SRE team and the service account used by the application should be granted a Cloud KMS Admin role in the project keys that are generated

    B. The SRE team and the service account used by the application should be granted a Cloud KMS Admin role in the project keys that are used

    C. The SRE team should be granted a KMS Admin role for the project and the service account should be granted the role of Cloud KMS CryptoKey Encrypter/Decrypter in the project keys that are generated

D. The SRE team should be granted a KMS Admin role for the project and the service account should be granted the role of Cloud KMS CryptoKey Encrypter/Decrypter in the project keys that are used

E. None of the above

40. What is the primary purpose of Google Cloud Key Management Service (KMS)?

A. Encrypting data at rest

B. Managing access control for Google Cloud resources

C. Generating and managing cryptographic keys

D. Monitoring and auditing cloud security events

41. Which of the following statements about Google Cloud KMS keys is true?

A. Google Cloud KMS generates and stores Customer-Managed Keys (CMKs) for encryption

B. Google Cloud KMS uses only pre-defined keys provided by Google for encryption

C. Google Cloud KMS keys can be used for authentication purposes

D. Google Cloud KMS keys are automatically rotated by the service

42. How does Google Cloud KMS handle key versioning?

A. It automatically generates and manages multiple versions of a key

B. It allows users to manually create and manage multiple versions of a key

C. It only supports a single version of a key at a time

D. It doesn't support key versioning

43. What is the role of Identity and Access Management (IAM) in Google Cloud KMS?

A. IAM is used to encrypt and decrypt data using KMS keys

B. IAM provides access control for managing KMS resources and operations

C. IAM is responsible for automatically rotating KMS keys

D. IAM enforces encryption policies for data stored in Google Cloud Storage

44. How can you list all keys within a key ring in Google Cloud KMS using the gcloud command-line tool?

    A. gcloud kms keys list
    B. gcloud kms keyrings list-keys
    C. gcloud kms keyrings keys list
    D. gcloud kms list-keys

45. A company wants to ensure compliance with data protection regulations by identifying and redacting sensitive information in its cloud storage. Which feature of Google Cloud DLP can help achieve this?

    A. Data classification
    B. Data de-identification
    C. Data discovery
    D. Data masking

46. An organization needs to analyze large volumes of data for patterns and trends without exposing any Personally Identifiable Information (PII). Which Google Cloud DLP technique can be applied to achieve this?

    A. Tokenization
    B. Redaction
    C. Masking
    D. Anonymization

47. A development team wants to securely store API keys and database credentials for their microservices deployed on Google Kubernetes Engine (GKE). They also need to rotate these secrets regularly. Which Google Cloud Secret Manager feature can help achieve this?

    A. Secret versioning
    B. Secret replication
    C. Secret auditing
    D. Secret scanning

48. A financial institution wants to grant granular access control to different teams within their organization for managing secrets. They want to ensure that only authorized individuals can create, access, and update specific secrets. Which Google Cloud Secret Manager feature provides this capability?

    A. IAM integration
    B. Secret rotation
    C. Secret monitoring
    D. Secret access controls

49. A large organization wants to have centralized visibility and monitoring of security risks and vulnerabilities across its Google Cloud infrastructure. Which feature of Google Cloud Security Command Center can help achieve this?

    A. Asset Inventory
    B. Security findings
    C. Security Health Analytics
    D. Security policies

50. A company wants to assess the compliance of its Google Cloud resources with industry-specific security standards and regulations. Which feature of Google Cloud Security Command Center can assist in this evaluation?

    A. Security Health Analytics
    B. Security policies
    C. Security findings
    D. Asset Inventory

Answer Key

1. A
2. D
3. C
4. D
5. B
6. D
7. C
8. B
9. B
10. A and B
11. A
12. A
13. A
14. C
15. D
16. D
17. B
18. C
19. D
20. A
21. A
22. A
23. B
24. A
25. B
26. A
27. D
28. D
29. B
30. B

31. B
32. C
33. B, E, and F
34. C
35. B
36. B
37. D
38. D
39. C
40. C
41. A
42. A
43. B
44. B
45. A
46. C
47. A
48. D
49. B
50. B

Google Professional Cloud Security Engineer Exam – Mock Exam II

1. Which of the following options can be configured for Google Cloud Access Transparency?

   A. Granular access control policies for individual users

   B. Real-time monitoring and alerting of access activities

   C. Logging and retention settings for access logs

   D. Integration with third-party identity providers

2. Which of the following statements is true regarding Google Cloud Access Approval?

   A. Access Approval allows administrators to grant automatic access to all users within a Google Cloud project

   B. Access Approval provides real-time monitoring and tracking of user access and activity within Google Cloud environments

   C. Access Approval streamlines and controls Google user access to sensitive resources and data within Google Cloud

   D. Access Approval is used to authenticate user identities for secure access to Google Cloud services

3. Which of the following options allows you to create hybrid connectivity between your on-premises network and Google Cloud?

   A. Google Cloud VPN

   B. Cloud Dataflow

   C. Google Cloud Storage

   D. Google Cloud Spanner

4. Which of the following statements accurately describes the default firewall rules in Google Cloud VPC auto mode?

   A. The default firewall rules allow incoming connections from any source to the instances within the VPC network

   B. The default firewall rules block all incoming and outgoing connections for instances within the VPC network

   C. The default firewall rules allow incoming connections from the internet to instances within the VPC network while blocking outgoing connections

   D. The default firewall rules allow outgoing connections from instances within the VPC network while blocking incoming connections

5. Which of the following requirements apply when establishing VPC peering in Google Cloud, depending on the VPC mode?

   A. In VPC auto mode, VPC networks must be located in the same region

   B. In VPC custom mode, VPC networks must have non-overlapping IP address ranges

   C. In both VPC auto mode and VPC custom mode, VPC peering can only be established within the same project

   D. In both VPC auto mode and VPC custom mode, VPC peering requires specific IAM roles to be assigned to the user

6. How are Google Cloud VPC firewall rules evaluated for incoming traffic?

   A. Firewall rules are evaluated in ascending order based on their creation timestamp

   B. Firewall rules are evaluated in descending order based on their priority value

   C. Firewall rules are evaluated randomly to ensure fair traffic distribution

   D. Firewall rules are evaluated based on their assigned project ID

7. Which of the following describes the different types of tags that can be used in Google Cloud VPC firewall rules?

   A. Network tags and service account tags

   B. Ingress tags and egress tags

   C. Source tags and destination tags

   D. Priority tags and action tags

8. Which of the following statements is true regarding Google Cloud Private Google Access?

   A. It provides access to Google APIs and services through the public internet
   B. It allows instances in a VPC network to access Google services using internal IP addresses
   C. It is only available for VPC networks located in the same region as the Google service
   D. It requires the creation of VPN tunnels for secure communication with Google services

9. What is the primary purpose of Google Cloud Identity-Aware Proxy (IAP)?

   A. To provide secure access to Google Cloud Virtual Machines (VMs) and applications
   B. To manage user identities and access permissions within Google Cloud
   C. To encrypt and secure data during transit between Google Cloud services
   D. To monitor and detect potential security threats within Google Cloud

10. What is the role of Cloud Identity-Aware Proxy (IAP) in protecting applications deployed on Google Cloud?

    A. It allows easy access management and authorization of users
    B. It provides encryption of data at rest and in transit
    C. It monitors application performance and uptime
    D. It automatically scales applications based on traffic demand

11. In Google Cloud NAT, what is the purpose of using the Manual NAT mapping option?

    A. To allow the automatic scaling of NAT configurations based on traffic demands
    B. To assign specific IP addresses for source and destination NAT mapping
    C. To enable fine-grained control over port allocation for NAT translations
    D. To integrate NAT configurations with Cloud Router for advanced routing capabilities

12. What is the primary purpose of Google Cloud Armor?

    A. To provide load balancing capabilities for Google Cloud resources
    B. To secure and protect applications against web-based attacks
    C. To monitor and analyze network traffic for performance optimization
    D. To manage user identities and access permissions within Google Cloud

13. What is the role of security policies in Google Cloud Armor?

    A. Security policies define rules for access control to Google Cloud resources
    B. Security policies enforce encryption of data at rest and in transit
    C. Security policies enable the configuration of firewall rules and network ACLs
    D. Security policies specify the behavior of Google Cloud Armor for incoming requests

14. What is the purpose of log sinks in Google Cloud Logging?

    A. To collect logs from various Google Cloud services and store them in Cloud Storage
    B. To analyze log data and generate insights using machine learning algorithms
    C. To export logs to third-party logging and monitoring systems
    D. To manage access controls and permissions for log data

15. What is the benefit of enabling log-based metrics in Google Cloud Logging?

    A. It allows you to visualize log data using custom dashboards and charts
    B. It enables you to monitor specific events and trigger alerts based on log entries
    C. It provides real-time analysis of log data to detect security threats
    D. It helps you automatically manage log storage and retention policies

16. What is the purpose of instance templates in Google Cloud Compute Engine?

    A. Instance templates are used to create snapshots of Compute Engine instances
    B. Instance templates allow for the management of machine types and sizes
    C. Instance templates enable the creation of custom images from scratch
    D. Instance templates define the configuration settings for creating VM instances

17. Which statement accurately describes the image lifecycle management for Compute Engine on Google Cloud?

    A. Compute Engine provides automatic image lifecycle management, including regular backups and versioning for all compute images
    B. Users have full control over the image lifecycle, including the ability to create and manage custom image snapshots and metadata
    C. Compute Engine automatically deletes all unused images after a specific period of inactivity to optimize storage usage
    D. Image lifecycle management is only available for Google-managed images and cannot be applied to custom images

18. Which Google Cloud feature provides enhanced security for Virtual Machines (VMs) by enabling Secure Boot, virtual Trusted Platform Module (vTPM), and integrity monitoring?

    A. Shielded VM
    B. Virtual Private Cloud (VPC)
    C. Google Cloud Armor
    D. Cloud Security Scanner

19. Which security measure should be implemented to secure the artifacts produced by a Google Cloud CI/CD pipeline?

    A. Storing artifacts in a publicly accessible storage bucket
    B. Encrypting the artifacts at rest and in transit
    C. Granting broad read access to all pipeline service accounts
    D. Disabling the logging and auditing of artifact access

20. Which Google Cloud feature helps enforce a secure software supply chain by allowing organizations to define and enforce custom policies for container image deployment based on metadata and attributes?

    A. Binary Authorization
    B. Container Registry
    C. Cloud Build
    D. Google Kubernetes Engine (GKE)

21. Which security feature in Google Kubernetes Engine (GKE) helps protect container workloads by automatically managing and rotating encryption keys for data at rest?

    A. Google Cloud Armor
    B. Cloud Identity-Aware Proxy (IAP)
    C. Binary Authorization
    D. GKE Data Encryption using Cloud Key Management Service (KMS)

22. Which statement accurately describes the purpose of namespaces in Google Kubernetes Engine (GKE)?

    A. Namespaces are used to control network traffic between different GKE clusters
    B. Namespaces provide logical isolation within a single GKE cluster by partitioning resources
    C. Namespaces are responsible for automatically scaling the number of nodes in a GKE cluster
    D. Namespaces ensure secure access to the GKE API server through token-based authentication

23. What is the primary purpose of Role-Based Access Control (RBAC) in Google Kubernetes Engine (GKE)?

    A. RBAC enables fine-grained control over permissions and access to Kubernetes resources
    B. RBAC automatically scales the number of GKE nodes based on the workload demand
    C. RBAC manages containerized applications' network traffic within a GKE cluster
    D. RBAC ensures secure communication between GKE clusters using a service mesh

24. Which statement accurately describes the role of a service mesh in Google Kubernetes Engine (GKE)?

    A. A service mesh provides additional encryption layers for securing data at rest within a GKE cluster
    B. A service mesh is responsible for managing network policies and ingress control for GKE services
    C. A service mesh enhances observability, traffic management, and security for microservices in a GKE cluster
    D. A service mesh automatically scans container images for vulnerabilities and security risks in GKE

25. Which security measure helps ensure the integrity and security of container images in Google Cloud?

    A. Container Registry vulnerability scanning
    B. Disabling image scanning for faster image deployment
    C. Running containers with elevated privileges for system compatibility
    D. Sharing container images publicly without any access controls

26. True or false: Container vulnerability scanning in Google Cloud automatically identifies and alerts users about security risks and vulnerabilities in container images.

    A. True
    B. False

27. Your Google Cloud organization policies provide guardrails for security controls for all organizations. Which of these are not best practices to follow when setting up organization policies (select all that apply)?

    A. Organization policies should be implemented over a period of time as your requirements become clear
    B. Delegate control of organization policies to the SRE teams

C. Prior to enforcing policies, thoroughly test and validate their impact on your resources and applications

D. It is usually not required to monitor and audit policies as they provide a robust framework to implement security controls right from the start

E. Since organization policies are set at the organizational level, it is often not required to implement them at the folder or project level

F. You should not version control policies via infrastructure as code

28. When creating projects in Google, which of the following are not best practices (select all that apply)?

A. Implement consistent naming conventions for your projects to ensure clarity and ease of identification

B. Use the default project controls since it allows you to get up and running quickly

C. Enable audit logs and monitoring for your projects to track and analyze activities within your environment

D. Google doesn't place a cap on the number of projects in a Google Cloud organization so it is safe to just leave the project and create a new one

E. Use IAM roles that are broad enough to grant more access than is needed so development is not hindered

F. Utilize Google Cloud's resource hierarchy, including folders and projects, to organize and manage resources effectively

29. A customer has a team that exclusively manages user identity and access. A cloud engineering team manages Google Cloud and has an intimate knowledge of organization policies. What is the right permission model for this company to manage Google Cloud securely?

A. Assign the super admin role to the cloud engineering team so they can manage users and groups for the entire organization.

B. Restrict the super admin role. Assign the user and group management roles to the identity access team. Assign the organization administrator role to the cloud engineering team.

C. Assign the super admin role to the identity access team and assign the organization administrator role to the cloud engineering team.

D. Assign the super admin and organization administrator roles to the identity access team so they can effectively manage access to the entire organization.

30. A Google Workspace customer is investigating the use of Google Cloud. They have called you to help with planning. They would like to set up Google Cloud in such a way that the roles and responsibilities in Cloud and Workspace are seperated while minimizing the effort required to set up Google Cloud. What approach would you recommend?

    A. Set up Google Cloud in the same organization as Google Workspace. Assign the organization administrator role to the cloud engineering team. Since identities are already provisioned, you can get up and running quickly.

    B. Create a subdomain in the Google Workspace domain and use that to set up Google Cloud.

    C. Create a new domain for Google Cloud. Provide access to users and groups from the Workspace domain to the Google Cloud domain.

    D. Create a new domain for Google Cloud, and provision users and groups to the Google Cloud domain.

31. Developers in an organization are prototyping a few applications on Google Cloud and are using their personal/consumer Gmail accounts to set up and manage their projects within Google Cloud. A security engineer raises this practice as a concern to the leadership team because of the lack of centralized project management and access to the data being stored in these accounts. Which solution should be used to resolve this concern?

    A. Enforce the setup of security keys using the 2-Step Verification (2SV) method for those Gmail accounts

    B. Set up Cloud Identity and require the developers to use those accounts for Google Cloud work

    C. Require the developers to log/store their Gmail passwords with the security team

    D. Enable logging on all Google Cloud projects to track all developer activity

32. It is a security best practice to set a session length for Google Cloud. Which of the following are not best practices when setting the session length (select all that apply)?

    A. Set the same session length for all Google applications (such as Gmail), including Google Cloud

    B. Exempt trusted apps from requiring session reauthentication

    C. Set a reauthentication policy to not require reauthentication

    D. Set a uniform session control for all users irrespective of their roles and responsibilities in your organization

    E. Set policies to limit long sessions

    F. Session length should apply to any application, including third-party applications that require user authorization for Google Cloud

33. You are setting up Google Cloud Directory Sync to provision users and groups to Cloud Identity. What should you not do (select all that apply)?

    A. Set a read-only account to your LDAP server to query users and groups
    B. Query all users and groups from your LDAP server and provision them to Cloud Identity
    C. Just select a few key attributes to sync to Cloud Identity
    D. Set a base distinguished name so Google Cloud Directory Sync (GCDS) can see all users and groups from your directory

34. In a large organization with multiple subsidiaries, each subsidiary has its own Google Cloud project. The organization wants to ensure that the IAM policies for each subsidiary's project are managed independently, but they also need a centralized view of IAM policies across all projects for auditing purposes. Which IAM feature can fulfill these requirements?

    A. IAM custom roles
    B. IAM conditions
    C. IAM service accounts
    D. IAM resource hierarchy

35. A company wants to grant access to specific resources in Google Cloud based on user attributes such as department, location, and job title. They also want to automate access provisioning and deprovisioning based on changes to user attributes. Which IAM feature can help accomplish this?

    A. IAM roles
    B. IAM Group membership
    C. IAM policies
    D. IAM service accounts
    E. IAM conditions

36. A multinational company wants to delegate administrative responsibilities for specific Google Cloud projects to regional IT teams. However, they want to ensure that the centralized IT team retains overall control and visibility of all projects. Which IAM feature can facilitate this delegation while maintaining centralized control?

    A. IAM service accounts
    B. IAM custom roles
    C. IAM policies
    D. IAM resource hierarchy

37. A company wants to grant a third-party consultant temporary access to their Google Cloud project to perform a security assessment. They want to ensure that the consultant has the necessary access rights for the assessment, and they want to make sure the consultant has access to the environment during working hours and gets a minimum privileged role. What can help achieve this (select all that apply)?

    A. Assign project viewer roles
    B. Assign IAM conditions in the IAM policies
    C. Create an IAM service account for the consultant
    D. Create a service account key for the consultant so they can execute queries

38. A company wants to implement fine-grained access control for a specific Google Cloud resource, allowing only specific actions to be performed by specific users. They want to ensure that users can perform only authorized actions and nothing beyond that. Which IAM feature can provide this level of granularity in access control?

    A. IAM custom roles
    B. IAM conditions
    C. IAM policies
    D. IAM service accounts

39. Google Cloud Key Management Service (KMS) supports all of the following purposes of encryption algorithm, except:

    A. Symmetric encryption
    B. Asymmetric signing
    C. Asymmetric encryption
    D. MAC signing
    E. Symmetric signing

40. A cloud customer has an on-premises key management system and wants to generate, protect, rotate, and audit encryption keys with it. How can the customer use Cloud Storage with their own encryption keys?

    A. Declare usage of default encryption at rest in the audit report on compliance
    B. Upload encryption keys to the same Cloud Storage bucket
    C. Use Customer-Managed Encryption Keys (CMEKs)
    D. Use Customer-Supplied Encryption Keys (CSEKs)

41. Which encryption algorithm is used with default encryption in Cloud Storage?

    A. AES-256
    B. SHA512
    C. MD5
    D. 3DES

42. A cloud customer wants to use the same encryption keys in two projects. What are the possible options for customers to do this?

    A. Call the encrypt and decrypt methods of the KMS API in the project where the key exists
    B. Export the encryption from the first project and encrypt in the other project
    C. Use Cloud HSM in both projects and import the key there
    D. Bring the key from on-premises for both workloads

43. How can you create a new key ring in Google Cloud KMS using the gcloud command-line tool?

    A. gcloud kms keyrings create
    B. gcloud kms keyrings add
    C. gcloud kms keyrings insert
    D. gcloud kms keyrings generate

44. A financial institution wants to audit and monitor the usage of encryption keys across its Google Cloud projects. They need visibility into key usage, including key creation, deletion, and key operations. Which feature of Google Cloud KMS provides this capability?

    A. Key labels
    B. Key rings
    C. Key rotation
    D. Key usage logs

45. A healthcare organization wants to detect and protect sensitive health information stored in different languages, including English, Spanish, and Mandarin. Which feature of Google Cloud DLP can be used to achieve language-agnostic data detection?

    A. Custom infoType
    B. Predefined infoType
    C. Inspect Templates
    D. Entity Extraction

46. A financial institution wants to prevent the accidental sharing of sensitive data through a data warehouse. Which Google Cloud DLP feature can help accomplish this?

    A. Data discovery
    B. Job configuration
    C. Content redaction
    D. Inspect templates

47. An organization wants to grant different teams within their company granular access control over secrets stored in Google Cloud Secret Manager. They want to ensure that each team can only access and manage secrets specific to their projects. Which feature of Google Cloud Secret Manager enables this?

    A. Secret monitoring
    B. Secret replication
    C. Secret access controls
    D. Secret auditing

48. A company wants to integrate its existing on-premises secrets management system with Google Cloud Secret Manager to have a unified solution. Which feature of Google Cloud Secret Manager allows this integration?

    A. Secret import
    B. Secret monitoring
    C. Secret auditing
    D. Sync secrets using APIs

49. An organization wants to detect and respond to potential security incidents in real time within their Google Cloud environment. Which feature of Google Cloud Security Command Center can facilitate this proactive monitoring?

    A. Security findings
    B. Security Health Analytics
    C. Security policies
    D. Asset Inventory

50. A company wants to track and manage the security posture of its Google Cloud projects individually. Which feature of Google Cloud Security Command Center provides project-specific security insights and recommendations?

    A. Security Health Analytics
    B. Security findings
    C. Security policies
    D. Asset Inventory

Answer Key

1. C
2. C
3. A
4. D
5. C
6. B
7. A
8. B
9. A
10. A
11. C
12. B
13. D
14. C
15. B
16. D
17. B
18. A
19. B
20. A
21. D
22. B
23. A

24. C
25. A
26. A
27. A, B, D, E, and F
28. B, D, and E
29. B
30. C
31. B
32. A, B, C, and D
33. B and C
34. D
35. B
36. B
37. A and B
38. B
39. C
40. D
41. A
42. A
43. A
44. D
45. A
46. C
47. C
48. A
49. A
50. A