CIS Benchmarks

Google Cloud Platform Foundation Benchmark - DRAFT

v1.0.0 - 08-21-2018

Terms of Use

Please see the below link for our current terms of use:

https://www.cisecurity.org/cis-securesuite/cis-securesuite-membership-terms-of-use/

Table of Contents

Overview

This security configuration benchmark covers foundational elements of Google Cloud Platform. The recommendations detailed here are important security considerations when designing your infrastructure on Google Cloud Platform. Most of the recommendations provided with this release of the benchmark covers security considerations only at individual Project level and not at the organization level.

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, platform deployment, and/or DevOps personnel who plan to develop, deploy, assess, or secure solutions on Google Cloud Platform.

Consensus Guidance

This benchmark was created using a consensus review process comprised of subject matter experts. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Each CIS benchmark undergoes two phases of consensus review. The first phase occurs during initial benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the benchmark. This discussion occurs until consensus has been reached on benchmark recommendations. The second phase begins after the benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the benchmark. If you are interested in participating in the consensus process, please visit https://workbench.cisecurity.org/.

Typographical Conventions

The following typographical conventions are used throughout this guide:

Convention	Meaning
`Stylized Monospace font`	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
Monospace font	Used for inline code, commands, or examples. Text should be interpreted exactly as presented.
<italic font in brackets>	Italic texts set in angle brackets denote a variable requiring substitution for a real value.
Italic font	Used to denote the title of a book, article, or other publication.
Note	Additional information or caveats

Scoring Information

A scoring status indicates whether compliance with the given recommendation impacts the assessed target's benchmark score. The following scoring statuses are used in this benchmark:

Scored

Failure to comply with "Scored" recommendations will decrease the final benchmark score. Compliance with "Scored" recommendations will increase the final benchmark score.

Not Scored

Failure to comply with "Not Scored" recommendations will not decrease the final benchmark score. Compliance with "Not Scored" recommendations will not increase the final benchmark score.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1**

  Items in this profile intend to:

  - be practical and prudent;
  - provide a clear security benefit; and
  - not inhibit the utility of the technology beyond acceptable means.

- **Level 2**

  This profile extends the "Level 1" profile. Items in this profile exhibit one or more of the following characteristics:

  - are intended for environments or use cases where security is paramount
  - acts as a defense in depth measure
  - may negatively inhibit the utility or performance of the technology.

Acknowledgements

This benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

Contributor
Shobha H D Information security engineer
Pravin Goyal , Pravin Goyal
Aditi Sahasrabudhe
Mike Wicks GCIH, GSEC, GSLC, GCFE, ECSA

Editor
Prabhu Angadi Security Content Author (Compliance | Configuration | Checklist)
Parag Patil CISSP, ISO27001LA, ECSA, CEH
Pradeep R B

Recommendations

1 Identity and Access Management

This section covers recommendations addressing Identity and Access Management on Google Cloud Platform.

1.1 Ensure that corporate login credentials are used instead of Gmail accounts (Scored)

Profile Applicability:

- Level 1

Description:

Use corporate login credentials instead of Gmail accounts.

Rationale:

Gmail accounts are personally created and controllable accounts. Organizations seldom have any control over them. Thus, it is recommended that you use fully managed corporate Google accounts for increased visibility, auditing, and control over access to Cloud Platform resources.

Audit:

For each Google Cloud Platform project, list the accounts configured in that project:

```
gcloud projects get-iam-policy <Project-ID> | grep gmail.com
```

No Gmail accounts should be listed.

Remediation:

Follow the documentation and setup corporate login accounts.

Impact:

None.

Default Value:

By default, any Gmail account can be associated with a Google Cloud Platform Project.

References:

1. https://cloud.google.com/docs/enterprise/best-practices-for-enterprise-organizations#use_corporate_login_credentials
2. https://support.google.com/work/android/answer/6371476

CIS Controls:

Version 7

16.2 Configure Centralized Point of Authentication

Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.

1.2 Ensure that multi-factor authentication is enabled for all non-service accounts (Not Scored)

Profile Applicability:

- Level 1

Description:

Setup multi-factor authentication for Google Cloud Platform accounts.

Rationale:

Multi-factor authentication requires more than one mechanism to authenticate a user. This secures your logins from attackers exploiting stolen or weak credentials.

Audit:

For each Google Cloud Platform project,
Step 1: Identify the non-service accounts.
Step 2: Manually verify that multi-factor authentication for each account is set.

Remediation:

For each Google Cloud Platform project,
Step 1: Identify the non-service accounts.
Step 2: Setup multi-factor authentication for each account.

Impact:

None

Default Value:

By default, multi-factor authentication is not set.

References:

1. https://cloud.google.com/solutions/securing-gcp-account-u2f

CIS Controls:

Version 7

16.3 Require Multi-factor Authentication

Require multi-factor authentication for all user accounts, on all systems, whether managed onsite or by a third-party provider.

1.3 Ensure that there are only GCP-managed service account keys for each service account (Scored)

Profile Applicability:

- Level 1

Description:

User managed service account should not have user managed keys.

Rationale:

Anyone who has access to the keys will be able to access resources through the service account. GCP-managed keys are used by Cloud Platform services such as App Engine and Compute Engine. These keys cannot be downloaded. Google will keep the keys and automatically rotate them on an approximately weekly basis. User-managed keys are created, downloadable, and managed by users. They expire 10 years from creation.

For user-managed keys, User have to take ownership of key management activities which includes:

- Key storage
- Key distribution
- Key revocation
- Key rotation
- Protecting the keys from unauthorized users
- Key recovery Even after owners precaution, Keys can be easily leaked by common development malpractices like checking keys into the source code or leaving them in Downloads directory, antecedently leaving them on support blogs/channels.

It is recommended to prevent use of User-managed service account keys.

Audit:

From CLI:
List All the service accounts:

```
gcloud iam service-accounts list
```

Identify user managed service accounts as such account `EMAIL` ends with `iam.gserviceaccount.com`
For each user managed Service Account, list the keys managed by the user:

```
gcloud iam service-accounts keys list --iam-account=<Service Account> --
managed-by=user
```

No keys should be listed.

Remediation:

From CLI:
To delete User manages Service Account Key,

```
gcloud iam service-accounts keys delete --iam-account=<user-managed-service-
account-EMAIL> <KEY-ID>
```

Impact:

Deleting User managed Service Account Keys may break communication with the applications using the corresponding keys

Default Value:

By default, There are no user managed keys are created for user managed service accounts.

References:

1. https://cloud.google.com/iam/docs/understanding-service-accounts#managing_service_account_keys

Notes:

Unser managed key cannot be created on GCP-Managed Service Account.

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

1.4 Ensure that ServiceAccount has no Admin privileges. (Scored)

Profile Applicability:

- Level 1

Description:

A service account is a special Google account that belongs to your application or a VM, instead of to an individual end user. Your application uses the service account to call the Google API of a service, so that the users aren't directly involved, It's recommended not to use admin access for ServiceAccount.

Rationale:

Service accounts represent service-level security of the Resources (application or a VM) which can be determined by the roles assigned to it. Enrolling ServiceAccount with Admin rights gives full access to assigned application or a VM, ServiceAccount Access holder can perform critical actions like delete, update change settings etc. without the intervention of user, So It's recommended not to have Admin rights.

This recommendation is applicable only for `User-Managed user created` service account (Service account with nomenclature: `SERVICE_ACCOUNT_NAME@PROJECT_ID.iam.gserviceaccount.com`).

Audit:

From Console

1. Go to `IAM & admin/IAM` using `https://console.cloud.google.com/iam-admin/iam`
2. Go to the `Members`
3. Ensure that there are no `User-Managed user created service account(s)` with roles containing `*Admin` or role matching `Editor` or role matching `Owner`

Via CLI gcloud :

1. Get the policy that you want to modify, and write it to a JSON file:
   gcloud projects get-iam-policy PROJECT_ID --format json > iam.json
2. The contents of the JSON file will look similar to the following. Note that `role` of members group associated with each `serviceaccount` does not contains `*Admin` or does not matches `roles/editor` or does not matches `roles/owner`

Sample Json output:

```
{
"bindings": [
{
   "members": [
      "serviceAccount:our-project-123@appspot.gserviceaccount.com",
    ],
    "role": "roles/appengine.appAdmin"
},
{
    "members": [
      "user:email1@gmail.com"
    ],
    "role": "roles/owner"
},
{
   "members": [
      "serviceAccount:our-project-123@appspot.gserviceaccount.com",
      "serviceAccount:123456789012-compute@developer.gserviceaccount.com"
    ],
    "role": "roles/editor"
}
],
"etag": "BwUjMhCsNvY=",
"version": 1
}
```

Remediation:

From Console

1. Go to `IAM & admin/IAM` using `https://console.cloud.google.com/iam-admin/iam`
2. Go to the `Members`
3. Identify `User-Managed user created` service account with roles containing `*Admin` or role matching `Editor` or role matching `Owner`
4. Click `Delete bin` icon to remove role from member (service account in this case)

Via CLI gcloud :

1. Using a text editor, Remove `Role` which contains `roles/*Admin` or matched `roles/editor` or matches 'roles/owner`. Add a role to the bindings array that defines the group members and the role for those members.

For example, to grant the role roles/appengine.appViewer to the `ServiceAccount` which is roles/editor, you would change the example shown below as follows:

```
{
"bindings": [
{
   "members": [
     "serviceAccount:our-project-123@appspot.gserviceaccount.com",
    ],
```

```
    "role": "roles/appengine.appViewer"
},
{
    "members": [
     "user:email1@gmail.com"
    ],
    "role": "roles/owner"
   },
{
    "members": [
      "serviceAccount:our-project-123@appspot.gserviceaccount.com",
      "serviceAccount:123456789012-compute@developer.gserviceaccount.com"
    ],
    "role": "roles/editor"
}
],
"etag": "BwUjMhCsNvY="
}
```

2. Update the project's IAM policy:
   gcloud projects set-iam-policy PROJECT_ID iam.json

Impact:

After removing `*Admin or Editor or Owner` role assignments from service accounts, may break functionality that uses impacted service accounts. Required role(s) should be assigned to impacted service accounts in order to restore broken functionalities.

Default Value:

User Managed (and not user created) default service accounts have `Editor (roles/editor)` role assigned to them to support GCP services they offer.

By Default there are no roles assigned to `User Managed User created` service accounts.

References:

1. https://cloud.google.com/sdk/gcloud/reference/iam/service-accounts/
2. https://cloud.google.com/iam/docs/understanding-roles
3. https://cloud.google.com/iam/docs/understanding-service-accounts

Notes:

Default (User managed but not User created) service accounts have `Editor (roles/editor)` role assigned to them to support GCP services they offer. Such Service accounts are: `PROJECT_NUMBER-compute@developer.gserviceaccount.com`, `PROJECT_ID@appspot.gserviceaccount.com`.

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

1.5 Ensure that IAM users are not assigned Service Account User role at project level (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended to assign `Service Account User (iam.serviceAccountUser)` role to a user for a specific service account rather than assigning the role to a user at project level.

Rationale:

A service account is a special Google account that belongs to application or a virtual machine (VM), instead of to an individual end user. Application/VM-Instance uses the service account to call the Google API of a service, so that the users aren't directly involved. In addition to being an identity, a service account is a resource which has IAM policies attached to it. These policies determine who can use the service account.

Users with IAM roles to update the App Engine and Compute Engine instances (such as App Engine Deployer or Compute Instance Admin) can effectively run code as the service accounts used to run these instances, and indirectly gain access to all the resources for which the service accounts has access. Similarly, SSH access to a Compute Engine instance may also provide the ability to execute code as that instance/Service account.

As per business needs, there could be multiple user-managed service accounts configured for a project. Granting the `iam.serviceAccountUser` role to a user for a project gives the user access to all service accounts in the project, including service accounts that may be created in the future. This can result into elevation of privileges by using service accounts and corresponding `Compute Engine instances`.

In order to implement `least privileges` best practices, IAM users should not be assigned `Service Account User` role at project level. Instead `iam.serviceAccountUser` role should be assigned to a user for a specific service account giving a user access to the service account.

Audit:

From Console:

1. Go to the IAM page in the GCP Console using `https://console.cloud.google.com/iam-admin/iam`

2. Click on filter table text bar, Type `Role: Service Account User`
3. Ensure no user is listed as a result of filter.

Via CLI gcloud:

To ensure IAM users are not assigned Service Account User role at project level

```
gcloud projects get-iam-policy zeta-environs-192610 --format json | jq
'.bindings[].role' | grep "roles/iam.serviceAccountUser"
```

Command should not return any output.

Remediation:

From Console:

1. Go to the IAM page in the GCP Console using `https://console.cloud.google.com/iam-admin/iam`
2. Click on filter table text bar, Type `Role: Service Account User`
3. Click `Delete Bin` icon in front of role `Service Account User` for every user listed as a result of a filter.

Via CLI gcloud :

1. Using a text editor, Remove Role which contains roles/iam.serviceAccountUser. Add a role to the bindings array that defines the group members and the role for those members.

For example, you can use the iam.json file shown below as follows:

```
{
"bindings": [
{
   "members": [
     "serviceAccount:our-project-123@appspot.gserviceaccount.com",
    ],
    "role": "roles/appengine.appViewer"
},
{
    "members": [
     "user:email1@gmail.com"
    ],
    "role": "roles/owner"
   },
{
    "members": [
      "serviceAccount:our-project-123@appspot.gserviceaccount.com",
      "serviceAccount:123456789012-compute@developer.gserviceaccount.com"
    ],
    "role": "roles/editor"
}
],
```

```
"etag": "BwUjMhCsNvY="
    }
```

2. Update the project's IAM policy:
   gcloud projects set-iam-policy PROJECT_ID iam.json

Impact:

After revoking `Service Account User` role at project level from all impacted user account(s),`Service Account User` role should be assigned to a user(s) for specific service account(s) as per business needs.

Default Value:

By Default, users do not have Service Account User role assigned at project level.

References:

1. https://cloud.google.com/iam/docs/service-accounts
2. https://cloud.google.com/iam/docs/granting-roles-to-service-accounts
3. https://cloud.google.com/iam/docs/understanding-roles
4. https://cloud.google.com/iam/docs/granting-changing-revoking-access

Notes:

To assign `roles/iam.serviceAccountUser` to a user role on a service account instead of a project:

1. go to `https://console.cloud.google.com/projectselector/iam-admin/serviceaccounts`

2. Select `Target Project`

3. Select `target service account` and Click `Permissions` on top bar. It will open permission pane on right side of the page

4. Add desired members with `Service Account User` role

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the

principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

16 Account Monitoring and Control
Account Monitoring and Control

1.6 Ensure user-managed/external keys for service accounts are rotated every 90 days or less (Scored)

Profile Applicability:

- Level 1

Description:

Service Account keys consist of a key ID (Private_key_Id) and Private key, which are used to sign programmatic requests that you make to Google cloud services accessible to that particular Service account. It is recommended that all Service Account keys are regularly rotated.

Rationale:

Rotating Service Account keys will reduce the window of opportunity for an access key that is associated with a compromised or terminated account to be used. Service Account keys should be rotated to ensure that data cannot be accessed with an old key which might have been lost, cracked, or stolen.

Each service account is associated with a key pair, which is managed by Google Cloud Platform (GCP). It is used for service-to-service authentication within GCP. Google rotates the keys daily.

GCP provides option to create one or more user-managed (also called as external key pairs) key pairs for use from outside GCP (for example, for use with Application Default Credentials). When a new key pair is created, user is enforced download the private key (which is not retained by Google). With external keys, users are responsible for security of the private key and other management operations such as key rotation. External keys can be managed by the IAM API, gcloud command-line tool, or the Service Accounts page in the Google Cloud Platform Console. GCP facilitates up to 10 external service account keys per service account to facilitate key rotation.

Audit:

From Console:

1. Go to `APIs & Services\Credentials` using `https://console.cloud.google.com/apis/credentials`
2. In Section `Service Account Keys`, for every External (user-managed) Service account key listed ensure `creation date` is less than past 90 days

via CLI gcloud:

1. List all Service accounts from a project

```
gcloud iam service-accounts list
```

2. For every service account list service account keys

```
gcloud iam service-accounts keys list --iam-account
[Service_Account_Email_Id] --format=json
```

3. Ensure every service account key for a service account has `"validAfterTime"` less than past 90 days

Remediation:

From Console:
Delete external (user managed) Service Account Key older than 90 days:

1. Go to `APIs & Services\Credentials` using `https://console.cloud.google.com/apis/credentials`
2. In Section `Service Account Keys`, for every external (user managed) Service account key with `creation date` is more than or equal to past 90 days click `Delete Bin Icon` to `Delete Service Account key`

Create New external (user managed) Service Account Key for a Service Account:

1. Go to `APIs & Services\Credentials` using `https://console.cloud.google.com/apis/credentials`
2. Click `Create Credentials` and Select `Service Account Key`
3. Choose Service account in drop-down list for which External (user managed) Service Account key needs to be created
4. Select desired key type format among `JSON` or `P12`
5. Click `Create`. It will download `private key`. Keep it safe.
6. Click `Close` if prompted
7. It will redirect to `APIs & Services\Credentials` page. Make a note of New `ID` displayed in section `Service account keys`

Impact:

Rotating service account key will break communication for depending applications. Dependent applications needs to configured manually with new key `id` displayed in section `Service account keys` and `private key` downloaded by user.

Default Value:

GCP does not provides automation option for External (user managed) Service key rotation.

References:

1. https://cloud.google.com/iam/docs/understanding-service-accounts#managing_service_account_keys
2. https://cloud.google.com/sdk/gcloud/reference/iam/service-accounts/keys/list
3. https://cloud.google.com/iam/docs/service-accounts

Notes:

For user-managed Service Account key(s), key management is entirely users responsibility.

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

1.7 Ensure that Separation of duties is enforced while assigning service account related roles to users (Not Scored)

Profile Applicability:

- Level 2

Description:

It is recommended that principle of Separation of duties is enforced while assigning service account related roles to users.

Rationale:

Built-in/Predefined IAM role `Service Account admin` allows user/identity to create, delete, manage service account(s). Built-in/Predefined IAM role `Service Account User` allows user/identity (with adequate privileges on Compute and App Engine) to assign service account(s) to Apps/Compute Instances.

Separation of duties is the concept of ensuring that one individual does not have all necessary permissions to be able to complete a malicious action. In Cloud IAM - service accounts, this could be an action such as using a service account to access resources that user should not normally have access to. Separation of duties is a business control typically used in larger organizations, meant to help avoid security or privacy incidents and errors. It is considered best practice.

Any user(s) should not have `Service Account Admin` and `Service Account User`, both roles assigned at a time.

Audit:

From Console:

1. Go to `IAM & Admin/IAM` using `https://console.cloud.google.com/iam-admin/iam`
2. Ensure no member has roles `Service Account Admin` and `Service account User` assigned.

Via CLI gcloud:

1. List all users and role assignments:

```
gcloud projects get-iam-policy [Project_ID]
```

2. Ensure that there are no common users found in member section for roles `roles/iam.serviceAccountAdmin` and `roles/iam.serviceAccountUser`

Remediation:

From Console:

1. Go to `IAM & Admin/IAM` using `https://console.cloud.google.com/iam-admin/iam`
2. For member having `Service Account Admin` and `Service account User` both roles granted/assigned, click on the `Delete Bin` icon to remove any one role from member.
   Removal of a role should be done as per the business requirement.

Impact:

Removed role should be assigned to some other user, as per business needs.

References:

1. https://cloud.google.com/iam/docs/service-accounts
2. https://cloud.google.com/iam/docs/understanding-roles
3. https://cloud.google.com/iam/docs/granting-roles-to-service-accounts

Notes:

Users granted with Owner (roles/owner) and Editor (roles/editor) have privileges equivalent to `Service Account Admin` and `Service Account User`. To avoid the misuse, Owner and Editor roles should be granted to very limited users and Use of these primitive privileges should be minimal. These requirements are addressed in separate recommendations.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

16 Account Monitoring and Control
Account Monitoring and Control

1.8 Ensure Encryption keys are rotated within a period of 365 days (Scored)

Profile Applicability:

- Level 1

Description:

Google Cloud Key Management Service stores cryptographic keys in a hierarchical structure designed for useful and elegant access control management. Access to resources.

The format for the rotation schedule depends on the client library that is used. For the gcloud command-line tool, the next rotation time must be in `ISO` or `RFC3339` format, and the rotation period must be in the form `INTEGER[UNIT]`, where units can be one of seconds (s), minutes (m), hours (h) or days (d).

Rationale:

Set a key rotation period and starting time, A key can be created with a specified `rotation period`, which is the time between when new key versions are generated automatically. A key can also be created with a specified next rotation time. A key is a named object representing a `cryptographic key` used for a specific purpose. The key material, the actual bits used for `encryption`, can change over time as new key versions are created.

A key is used to protect some `corpus of data`. You could encrypt a collection of files with the same key, and people with `decrypt` permissions on that key would be able to decrypt those files. Hence it's necessary to make sure `rotation period` is set to specific time.

Audit:

From Console

1. Go to `IAM & admin`.
2. Go to the `Encryption keys`.
3. Click on `key name` Ensure Key Rotation is less than 365 days from the current date.

Via CLI gcloud :

1. Ensure rotation is scheduled by `ROTATION_PERIOD` and `NEXT_ROTATION_TIME` for each key :
   gcloud kms keys list --keyring=<KEY_RING> --location= --
   format=json'(rotationPeriod)'

Ensure outcome values for `rotationPeriod` and `nextRotationTime` satisfies the below criteria.

`rotationPeriod is <= 525600m`

`rotationPeriod is <= 31536000s`

`rotationPeriod is <= 8760h`

`rotationPeriod is <= 365d`

`nextRotationTime is <= 365days` from current DATE

Remediation:

From Console

1. Go to `IAM & admin`.
2. Go to the `Encryption keys`.
3. Click on `Right side pop up blade` (3 dots).
4. Click on `Edit rotation period`.
5. Select new `rotationPeriod` and `nextRotationTime` from drop down.

Via CLI gcloud :

1. Update and schedule rotation by `ROTATION_PERIOD` and `NEXT_ROTATION_TIME` for each key :
   gcloud kms keys update new --keyring=<KEY_RING> --location= --next-rotation-time=<NEXT_ROTATION_TIME> --rotation-period=<ROTATION_PERIOD>

Impact:

After successful key rotation, Older key version is required in order to decrypt the data encrypted by that specific key version.

References:

1. https://cloud.google.com/kms/docs/key-rotation#frequency_of_key_rotationhttps://cloud.google.com/kms/docs/re-encrypt-data

Notes:

- Key rotation does NOT re-encrypt already encrypted data with the newly generated key version. If you suspect unauthorized use of a key, you should re-encrypt the data

protected by that key and then disable or schedule destruction of the prior key version.

- It is not recommended to rely solely on irregular rotation, but rather to use irregular rotation if needed in conjunction with a regular rotation schedule.

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

1.9 Ensure that Separation of duties is enforced while assigning KMS related roles to users (Scored)

Profile Applicability:

- Level 2

Description:

It is recommended that principle of Separation of duties is enforced while assigning KMS related roles to users.

Rationale:

Built-in/Predefined IAM role `Cloud KMS Admin` allows user/identity to create, delete, and manage service account(s). Built-in/Predefined IAM role `Cloud KMS CryptoKey Encrypter/Decrypter` allows user/identity (with adequate privileges on concerned resources) to encrypt and decrypt data at rest using encryption key(s). Built-in/Predefined IAM role `Cloud KMS CryptoKey Encrypter` allows user/identity (with adequate privileges on concerned resources) to encrypt data at rest using encryption key(s). Built-in/Predefined IAM role `Cloud KMS CryptoKey Decrypter` allows user/identity (with adequate privileges on concerned resources) to decrypt data at rest using encryption key(s).

Separation of duties is the concept of ensuring that one individual does not have all necessary permissions to be able to complete a malicious action. In Cloud KMS, this could be an action such as using a key to access and decrypt data that that user should not normally have access to. Separation of duties is a business control typically used in larger organizations, meant to help avoid security or privacy incidents and errors. It is considered best practice.

Any user(s) should not have `Cloud KMS Admin` and any of the `Cloud KMS CryptoKey Encrypter/Decrypter`, `Cloud KMS CryptoKey Encrypter`, `Cloud KMS CryptoKey Decrypter` roles assigned at a time.

Audit:

From Console:

1. Go to `IAM & Admin/IAM` using `https://console.cloud.google.com/iam-admin/iam`

2. Ensure no member has roles `Cloud KMS Admin` and any of the `Cloud KMS CryptoKey Encrypter/Decrypter`, `Cloud KMS CryptoKey Encrypter`, `Cloud KMS CryptoKey Decrypter` assigned.

Via CLI gcloud:

1. List all users and role assignments:

```
gcloud projects get-iam-policy [Project_ID]
```

2. Ensure that there are no common users found in member section for roles `cloudkms.admin` and any one of `Cloud KMS CryptoKey Encrypter/Decrypter`, `Cloud KMS CryptoKey Encrypter`, `Cloud KMS CryptoKey Decrypter`

Remediation:

From Console:

1. Go to `IAM & Admin/IAM` using `https://console.cloud.google.com/iam-admin/iam`
2. For member having `Cloud KMS Admin` and any of the `Cloud KMS CryptoKey Encrypter/Decrypter`, `Cloud KMS CryptoKey Encrypter`, `Cloud KMS CryptoKey Decrypter` roles granted/assigned, click on the `Delete Bin` icon to remove role from member.

Note: Removal of a roles should be done as per the business requirement.

Impact:

Removed roles should be assigned to some other user, as per business needs.

References:

1. https://cloud.google.com/kms/docs/separation-of-duties

Notes:

Users granted with Owner (roles/owner) and Editor (roles/editor) have privileges equivalent to `Cloud KMS Admin` and `Cloud KMS CryptoKey Encrypter/Decrypter`. To avoid the misuse, Owner and Editor roles should be granted to very limited users and Use of these primitive privileges should be minimal. These requirements are addressed in separate recommendations.

CIS Controls:

Version 7

4 Controlled Use of Administrative Privileges
Controlled Use of Administrative Privileges

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

16 Account Monitoring and Control
Account Monitoring and Control

1.10 Ensure API keys are not created for a project (Not Scored)

Profile Applicability:

- Level 2

Description:

Keys are insecure because they can be viewed publicly, such as from within a browser, or they can be accessed on a device where the key resides. It is recommended to use standard authentication flow instead.

Rationale:

Security risks involved in using API-Keys are below:

- API keys are a simple encrypted strings
- API keys do not identify the user or the application making the API request
- API keys are typically accessible to clients, making it easy to discover and steal an API key

To avoid security risk by using API keys, it is recommended to use standard authentication flow instead.

Audit:

From Console:

1. Go to `APIs & Services\Credentials` using `https://console.cloud.google.com/apis/credentials`
2. In Section `API Keys`, No API key should be listed

Remediation:

From Console:

1. Go to `APIs & Services\Credentials` using `https://console.cloud.google.com/apis/credentials`
2. In Section `API Keys`, to delete API Keys Click on the `Delete Bin Icon` in front of every `API Key Name`

Impact:

Deleting API key will break dependent applications (if any).

References:

1. https://cloud.google.com/docs/authentication/api-keys

Notes:

Google recommend using the standard authentication flow instead of using API keys. However, there are limited cases where API keys are more appropriate. For example, if there is a mobile application that needs to use the Google Cloud Translation API, but doesn't otherwise need a back-end server, API keys are the simplest way to authenticate to that API.

If use of API keys is required for your business then API keys should be secured properly.

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

1.11 Ensure API keys are restricted to use by only specified Hosts and Apps (Not Scored)

Profile Applicability:

- Level 1

Description:

Unrestricted keys are insecure because they can be viewed publicly, such as from within a browser, or they can be accessed on a device where the key resides. It is recommended to restrict API key usage only from trusted hosts, HTTP referrers and apps.

Rationale:

Security risks involved in using API-Keys are below:

- API keys are a simple encrypted strings
- API keys do not identify the user or the application making the API request
- API keys are typically accessible to clients, making it easy to discover and steal an API key

Because of this Google recommend using the standard authentication flow instead. However, there are limited cases where API keys are more appropriate. For example, if there is a mobile application that needs to use the Google Cloud Translation API, but doesn't otherwise need a back-end server, API keys are the simplest way to authenticate to that API.

In order to reduce attack vector, API-Keys can be restricted only to the trusted hosts, HTTP referrers and applications.

Audit:

From Console:

1. Go to `APIs & Services\Credentials` using `https://console.cloud.google.com/apis/credentials`
2. In Section `API Keys`, Click on the `API Key Name`. it will display API Key properties on new page.
3. For every API Key, ensure section `Key restrictions` parameter `Application restrictions` is not set to `None`

Or

ensure `Application restrictions` is set to `HTTP referrers` and referrer is not set to wild-

cards `(* or *.[TLD] or *.[TLD]/*) allowing access to any/wide HTTP referrer(s)`
Or
ensure `Application restrictions` is set to `IP addresses` and referrer is not set to `any host (0.0.0.0 or 0.0.0.0/0 or ::0)`

Remediation:

From Console:

1. Go to `APIs & Services\Credentials` using `https://console.cloud.google.com/apis/credentials`
2. In Section `API Keys`, Click on the `API Key Name`. It will display API Key properties on new page
3. In `Key restrictions` section set application restrictions to either of `HTTP referrers, IP Adresses, Android Apps, iOs Apps`.
4. Click `Save`
5. Repeat steps 2,3,4 for every unrestricted API key
   Note: Do not set `HTTP referrers` to wild-cards (* or *.[TLD] or *.[TLD]/*) allowing access to any/wide HTTP referrer(s)
   Do not set `IP addresses` and referrer to `any host (0.0.0.0 or 0.0.0.0/0 or ::0)`

Impact:

Setting `Application Restrictions` may break existing application functioning, if not done carefully.

Default Value:

By default, `Application Restrictions` are set to `None`.

References:

1. https://cloud.google.com/docs/authentication/api-keys

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

1.12 Ensure API keys are restricted to only APIs that application needs access (Not Scored)

Profile Applicability:

- Level 1

Description:

API keys are insecure because they can be viewed publicly, such as from within a browser, or they can be accessed on a device where the key resides. It is recommended to restrict API keys to use (call) only APIs required by an application.

Rationale:

Security risks involved in using API-Keys are below:

- API keys are a simple encrypted strings
- API keys do not identify the user or the application making the API request
- API keys are typically accessible to clients, making it easy to discover and steal an API key

Because of this Google recommend using the standard authentication flow instead. However, there are limited cases where API keys are more appropriate. For example, if there is a mobile application that needs to use the Google Cloud Translation API, but doesn't otherwise need a back-end server, API keys are the simplest way to authenticate to that API.

In order to reduce attack surface by providing `least privileges`, API-Keys can be restricted to use (call) only APIs required by an application.

Audit:

From Console:

1. Go to `APIs & Services\Credentials` using `https://console.cloud.google.com/apis/credentials`
2. In Section `API Keys`, Click on the `API Key Name`. it will display API Key properties on new page.
3. For every API Key, ensure section `Key restrictions` parameter `API restrictions` is not set to `None`

Or

`API restrictions` is not set to `Google Cloud APIs`

Note: `Google Cloud APIs` represents API collection of all cloud services/APIs offered by Google cloud.

Remediation:

From Console:

1. Go to `APIs & Services\Credentials` using `https://console.cloud.google.com/apis/credentials`
2. In Section `API Keys`, Click on the `API Key Name`. it will display API Key properties on new page
3. In `Key restrictions` section go to `API restrictions`
4. Click `Select API` drop-down to choose API
5. Click `Save`
6. Repeat steps 2,3,4,5 for every unrestricted API key

Note: Do not set `API restrictions` is `Google Cloud APIs`, as It allows access to all services offered by Google cloud.

Impact:

Setting `API restrictions` may break existing application functioning, if not done carefully.

Default Value:

By default, `API restrictions` are set to `None`.

References:

1. https://cloud.google.com/docs/authentication/api-keys
2. https://cloud.google.com/apis/docs/overview

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

1.13 Ensure API keys are rotated every 90 days (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended to rotate API keys every 90 days.

Rationale:

Security risks involved in using API-Keys are below:

- API keys are a simple encrypted strings
- API keys do not identify the user or the application making the API request
- API keys are typically accessible to clients, making it easy to discover and steal an API key

Because of this Google recommend using the standard authentication flow instead. However, there are limited cases where API keys are more appropriate. For example, if there is a mobile application that needs to use the Google Cloud Translation API, but doesn't otherwise need a backend server, API keys are the simplest way to authenticate to that API.

Once the key is stolen, it has no expiration, so it may be used indefinitely, unless the project owner revokes or regenerates the key. Rotating API keys will reduce the window of opportunity for an access key that is associated with a compromised or terminated account to be used. API keys should be rotated to ensure that data cannot be accessed with an old key which might have been lost, cracked, or stolen.

Audit:

From Console:

1. Go to `APIs & Services\Credentials` using `https://console.cloud.google.com/apis/credentials`
2. In Section `API Keys`, for every key ensure `creation date` is less than 90 days.

Remediation:

From Console:

1. Go to APIs & Services\Credentials using Https://console.cloud.google.com/apis/credentials

2. In Section API Keys, Click on the API Key Name. It will display API Key properties on new page
3. Click REGENERATE KEY to rotate API key
4. Click Save
5. Repeat steps 2,3,4 for every API key that is has not been rotated in last 90 days

Note: Do not set `HTTP referrers` to wild-cards (* or *.[TLD] or *.[TLD]/*) allowing access to any/wide HTTP referrer(s)
Do not set `IP addresses` and referrer to `any host (0.0.0.0 or 0.0.0.0/0 or ::0)`

Impact:

`Regenerating Key` may break existing client connectivity as client will try to connect with older API keys they have stored on devices.

References:

1. There is no option to automatically regenerate (rotate) API keys periodically.

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

2 Logging and Monitoring

This section covers recommendations addressing Logging and Monitoring on Google Cloud Platform.

2.1 Ensure that Cloud Audit Logging is configured properly across all services and all users from a project (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended that Cloud Audit Logging is configured to track all Admin activities and read, write access to user data.

Rationale:

Cloud Audit Logging maintains two audit logs for each project and organization: Admin Activity and Data Access.

1. Admin Activity logs contain log entries for API calls or other administrative actions that modify the configuration or metadata of resources. Admin Activity audit logs are enabled for all services and cannot be configured.
2. Data Access audit logs record API calls that create, modify, or read user-provided data. These are disabled by default and should be enabled.

   There are three kinds of Data Access audit log information:

   - Admin read: Records operations that read metadata or configuration information. Admin Activity audit logs record writes of metadata and configuration information which cannot be disabled.
   - Data read: Records operations that read user-provided data.
   - Data write: Records operations that write user-provided data.

It is recommended to have effective default audit config configured in such a way that:

1. logtype is set to DATA_READ (to logs user activity tracking) and DATA_WRITES (to log changes/tampering to user data)
2. audit config is enabled for all the services supported by Data Access audit logs feature
3. Logs should be captured for all users i.e.. there are no exempted users in any of the audit config section. This will ensure overriding audit config will not contradict the requirement.

Audit:

Using Command line:

1. Run command:

```
gcloud projects get-iam-policy [PROJECT_ID]
```

2. Policy should have default auditConfigs section which should have logtype set to DATA_WRITES and DATA_READ for all services.

Sample output for default audit configs may looks like this:

```
        auditConfigs:
        - auditLogConfigs:
        - logType: ADMIN_READ
        - logType: DATA_WRITE
        - logType: DATA_READ
         service: allServices
```

3. Any of the auditConfigs sections should not have parameter "exemptedMembers:" set which will ensure that Logging is enabled for all users and no user is exempted.

Remediation:

Using Command Line:

1. To Read project's IAM policy and store it in a file run a command:

```
gcloud projects get-iam-policy [PROJECT_ID] > /tmp/policy.yaml
```

2. Edit policy in /tmp/policy.yaml, adding or changing only the audit logs configuration to:

```
auditConfigs:
- auditLogConfigs:
  - logType: DATA_WRITE
  - logType: DATA_READ
  service: allServices
```

Note: `exemptedMembers:` is not set as audit logging should be enabled for all the users

3. To write new IAM policy run command:

```
gcloud projects set-iam-policy [PROJECT_ID] /tmp/policy.yaml
```

If the preceding command reports a conflict with another change, then repeat these steps, starting with the first step.

Impact:

There is no charge for Admin Activity audit logs. Enabling the Data Access audit logs might result in your project being charged for the additional logs usage.

Default Value:

Admin Activity logs are always enabled. They cannot be disabled. Data Access audit logs are disabled by default because they can be quite large.

References:

1. https://cloud.google.com/logging/docs/audit/
2. https://cloud.google.com/logging/docs/audit/https://cloud.google.com/logging/docs/audit/configure-data-access

Notes:

- Log type `DATA_READ` is equally important to that of `DATA_WRITE` to track detailed user activities.
- BigQuery Data Access logs are handled differently from other Data Access logs. BigQuery logs are enabled by default and cannot be disabled. They do not count against logs allotment and cannot result in extra logs charges.

CIS Controls:

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.7 Regularly Review Logs
On a regular basis, review logs to identify anomalies or abnormal events.

2.2 Ensure that sinks are configured for all Log entries (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended to create sink which will export copies of all the log entries.

Rationale:

Log entries are held in Stackdriver Logging for a limited time known as the retention period. After that, the entries are deleted. To keep log entries longer, sink can export them outside of Stackdriver Logging. Exporting involves writing a filter that selects the log entries to export, and choosing a destination in Cloud Storage, BigQuery, or Cloud Pub/Sub. The filter and destination are held in an object called a sink. To ensure all log entries are exported using sink ensure that there is no filter configured for a sink. Sinks can be created in projects, organizations, folders, and billing accounts.

Audit:

From Console:

1. Go to `Logging/Exports` by visiting `https://console.cloud.google.com/logs/exports?`
2. For every sing, click 3 dot button for Menu option and select `View Filter`.
3. Ensure that there is at least one sink with `empty` Sink Filter
4. Additionally for Sink with `empty` sink filter, Ensure resource mentioned in `Destination` Exists.

Via CLI gcloud:

1. Ensure sink with `empty filter` exists:

```
gcloud logging sinks list
```

Output should list at least one sink with `empty filter`.

2. Additionally Ensure that `Destination` for a sink with `Empty filter` exists.

If sink is destined to a `Specific Cloud Storage Bucket` ensure `Destination` Storage bucket exists using CLI gsutils:

```
gsutil list | grep <destination_Bucket_name>
```

should not return empty Output.

Remediation:

From Console:

1. Go to `Logging/Logs` by visiting `https://console.cloud.google.com/logs/viewer?`
2. Click down arrow symbol on `Filter Bar` at rightmost corner and select `Convert to Advanced Filter`
3. This will convert `Filter Bar` to `Advanced Filter Bar`
4. Clear any text from `Advanced Filter` - This will ensure that `log-filter` is set to empty and hence it captures all the logs
5. Click on `Submit Filter` and should display all logs
6. Click `Create Export` it will open `Edit Export` Menu on right
7. Configure `Sink Name`. In `sink service` click drop down `select destination` and select desired service to store exports, preferably `Cloud Storage`
8. In `Select Destination` click drop down and select desired destination resource or If required create new.
9. Click `Create Sink`

Using Command-Line:

To create a sink to export all log entries in google cloud storage bucket:

```
gcloud logging sinks create <sink-name>
storage.googleapis.com/<destination_bucket-name>
```

Note:

1. Created sink by command-line above will export logs in storage bucket however alternately sink can be configured to export logs into BigQuery, or Cloud Pub/Sub or `Custom Destination`
2. While creating sink option `--log-filter` is not used to ensure sink exporting all log entries.

Impact:

There are no costs or limitations in Stackdriver Logging for exporting logs, but the export destinations charge for storing or transmitting the log data.

Default Value:

By default, there are no sinks configured.

References:

1. https://cloud.google.com/logging/docs/reference/tools/gcloud-logginghttps://cloud.google.com/logging/quotas
2. https://cloud.google.com/logging/quotas
3. https://cloud.google.com/logging/docs/export/
4. https://cloud.google.com/logging/docs/export/using_exported_logs?hl=en_US&_ga=2.8327026.-1306762988.1522230337#destinations
5. https://cloud.google.com/logging/docs/export/configure_export_v2

Notes:

For Command-Line Audit and Remediation, Sink destination of type `Cloud Storage Bucket` is considered. However destination could be configured to `Cloud Storage Bucket` or `BigQuery` or `Cloud Pub\Sub` or `Custom Destination`. CLI commands would change accordingly.

CIS Controls:

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.4 Ensure adequate storage for logs
Ensure that all systems that store logs have adequate storage space for the logs generated.

2.3 Ensure that object versioning is enabled on log-buckets (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended to enable object versioning on log-buckets.

Rationale:

Logs can be exported by creating one or more sinks that include a logs filter and a destination. As Stackdriver Logging receives new log entries, they are compared against each sink. If a log entry matches a sink's filter, then a copy of the log entry is written to the destination.

Sinks can be configured to export logs in Storage buckets. To support the retrieval of objects that are deleted or overwritten, Object Versioning feature should be enabled on all such storage buckets where sinks are configured.

Audit:

Using Command-Line:

1. To list all sinks destined to storage buckets:

```
gcloud logging sinks list | grep storage.googleapis.com
```

2. For every storage bucket listed above, verify that Object Versioning is Enabled:

```
gsutil versioning get gs://<Bucket>
```

Output for command should return `Enabled`

Remediation:

Using Command-Line:

1. To list all sinks destined to storage buckets:

```
gcloud logging sinks list | grep storage.googleapis.com
```

2. For every storage bucket listed above, verify that Object Versioning is Enabled:

```
gsutil versioning set on gs://<Bucket>
```

Impact:

None

Default Value:

By Default Object Versioning is in `Suspended` state for a storage bucket.

References:

1. https://cloud.google.com/storage/docs/object-versioning

CIS Controls:

Version 7

6 Maintenance, Monitoring and Analysis of Audit Logs
Maintenance, Monitoring and Analysis of Audit Logs

2.4 Ensure log metric filter and alerts exists for Project Ownership assignments/changes (Scored)

Profile Applicability:

- Level 1

Description:

In order to prevent unnecessarily project ownership assignments to users/service-accounts and further misuses of project and resources, all `roles/Owner` assignments should be monitored.

Members (users/Service-Accounts) with role assignment to primitive role `roles/owner` are Project Owners.

Project Owner has all the privileges on a project it belongs to. These can be summarized as below:

```
- All viewer permissions on All GCP Services part within the project

- Permissions for actions that modify state of All GCP Services within the
project

- Manage roles and permissions for a project and all resources within the
project

- Set up billing for a project
```

Granting owner role to a member (user/Service-Account) will allow members to modify the IAM policy. Therefore grant the owner role only if the member has a legitimate purpose to manage the IAM policy. This is because as project IAM policy contains sensitive access control data and having a minimal set of users manage it will simplify any auditing that you may have to do.

Rationale:

Project Ownership Having highest level of privileges on a project, to avoid misuse of project resources project ownership assignment/change actions mentioned should be monitored and alerted to concerned recipients.

```
- Sending project ownership Invites

- Acceptance/Rejection of project ownership invite by user

- Adding `role\owner` to a user/service-account

- Removing a user/Service account from `role\owner`
```

Audit:

From GCP Console, Ensure prescribed Log metric present:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`https://console.cloud.google.com/logs/metrics?
2. In `User-defined Metrics` ensure at least one metric `<Log_Metric_Name>` present with filter text:

```
(protoPayload.serviceName="cloudresourcemanager.googleapis.com") AND
(ProjectOwnership OR projectOwnerInvitee) OR
(protoPayload.serviceData.policyDelta.bindingDeltas.action="REMOVE" AND
protoPayload.serviceData.policyDelta.bindingDeltas.role="roles/owner") OR
(protoPayload.serviceData.policyDelta.bindingDeltas.action="ADD" AND
protoPayload.serviceData.policyDelta.bindingDeltas.role="roles/owner")
```

From Stackdriver Console, Ensure prescribed Alerting Policy present:

1. Go to stackdriver account at `https://app.google.stackdriver.com/` and select target GCP project on top bar by clicking drop-down arrow symbol.
2. Go to in Left column click `Alerting` select `Policies Overview`
3. on `POLICIES WITH BASIC CONDITIONS` page ensure at least one of the Policies with condition `Violates when: Any logging.googleapis.com/user/<Log Metric Name> stream is above a threshold of .001 for greater than 1 minute` present and state is `ON`

Ensure Alerting Policy Notifications are configured to appropriate subscribers/recipients:

on `POLICIES WITH BASIC CONDITIONS` page click target `Policy Name` to open policy configuration. Check Notifications section for appropriate subscribers/Recipients.

Using CLI, Ensure prescribed Log metric present:

```
gcloud beta logging metrics list --format json
```

Output should contain a metric with filter set to

```
(protoPayload.serviceName="cloudresourcemanager.googleapis.com") AND
(ProjectOwnership OR projectOwnerInvitee) OR
(protoPayload.serviceData.policyDelta.bindingDeltas.action="REMOVE" AND
protoPayload.serviceData.policyDelta.bindingDeltas.role="roles/owner") OR
(protoPayload.serviceData.policyDelta.bindingDeltas.action="ADD" AND
protoPayload.serviceData.policyDelta.bindingDeltas.role="roles/owner")
```

property `metricDescriptor.name` for the identified metric that will be used in next step.

Using CLI, Ensure prescribed Alerting Policy present:

```
gcloud alpha monitoring policies list --format json
```

Output should contain an alert policy where:

- `conditions.filter` is set to `"project= "<ProjectID>" AND metric.type="<metricDescriptor.type from previous command output>""`
- AND `conditions.filter` does not contain any other parameter than `metric.type` and `project` which will restrict alerting to a particular resource/type e.g.. `resource.type`
- AND `conditions.thresholdValue` set to `0.001`
- AND `conditions.conditionThreshold.aggregations.alignmentPeriod` set to `60s`
- AND `conditions.conditionThreshold.aggregations.crossSeriesReducer` set to `REDUCE_COUNT`
- AND `conditions.conditionThreshold.aggregations.perSeriesAligner` set to `ALIGN_RATE`,
- AND `enabled` is set to `true`

Remediation:

From GCP Console, Create prescribed Log Metric:

1. Go to `Logging/Logs` by visiting `https://console.cloud.google.com/logs/viewer?`
2. Click down arrow symbol on `Filter Bar` at rightmost corner and select `Convert to Advanced Filter`
3. This will convert `Filter Bar` to `Advanced Filter Bar`
4. Clear any text from `Advanced Filter` and add text

```
(protoPayload.serviceName="cloudresourcemanager.googleapis.com") AND
(ProjectOwnership OR projectOwnerInvitee) OR
(protoPayload.serviceData.policyDelta.bindingDeltas.action="REMOVE" AND
protoPayload.serviceData.policyDelta.bindingDeltas.role="roles/owner") OR
(protoPayload.serviceData.policyDelta.bindingDeltas.action="ADD" AND
protoPayload.serviceData.policyDelta.bindingDeltas.role="roles/owner")
```

5. Click on `Submit Filter` and should display logs based on filter text set in step above
6. Click `Create Metric` it will open `Metric Export` Menu on right
7. Configure `Name`, `Description` to desired values
8. Set `Units` to `1` (default) and `Type` to `Counter`
9. Click `Create Metric`. This will take to `Logging/Logs` at `https://console.cloud.google.com/logs/metrics?`

From stackdriver Console, Create prescribed Alert Policy:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`

2. In section `User-defined Metrics` for target metric, click 3 dot icon in rightmost column to open menu options
3. Select `Create alert from Metric`.
4. It will take to `Stackdriver Console\Alerting\Create` and directly open `Add Metric Threshold Condition` window

```
Set `Target`: `Resource Type` to `Log Metric`

Set `Configuration`:

- IF METRIC : user/<Log Metric Name>

- Condition : above

- Threshold: .001

- For: 1 minute

Set `Resource` to `ANY`
```

Click `Save Condition`

5. It will take back to `Stackdriver Console\Alerting\Create`
6. In Section `2 Notifications` click `+ Add Notification`. Add desired channel(s) as required.
7. In section `3 Documentation` optionally `+ Add Documentation`
8. In Section `4 Name this policy` leave system provided policy name (Threshold - user/) or configure custom name
9. Click `Save Policy`. It will open `Alerting/Policies Overview` page which lists all the alert policies including this one.
   For alert policy Observe Condition which is currently set to

```
Any logging.googleapis.com/user/<Log_Metric_Name> is above a threshold of
0.001 for greater than 1 minute
```

It is been observed that without following next step, Alert policy will not generate any alerts or open any incidents.

10. For newly created policy, click `EDIT` to Open `Edit alerting Policy` pane
11. At section 1 Conditions\Basic Conditions\suggested condition click `Edit`
12. In `Target` Section, for `Aggregation` drop down select `count`
13. Click `save condition` and then click `Save Policy`. It will open `Alerting/Policies Overview`

Condition for same alert policy will be updated to:

```
Violates when: Any logging.googleapis.com/user/<Log_Metric_Name> stream is
above a threshold of 0.001 for greater than 1 minute
```

Now, newly created Alert policy will be able to generate alerts or open incidents.

Using CLI

Create prescribed Log Metric

- Use command: gcloud beta logging metrics create
- Reference for Command Usage: https://cloud.google.com/sdk/gcloud/reference/beta/logging/metrics/create

Create prescribed Alert Policy

- Use command: gcloud alpha monitoring policies create
- Reference for command Usage: https://cloud.google.com/sdk/gcloud/reference/alpha/monitoring/policies/create

Impact:

Based on Service Tiers, Stackdriver account may be charged.

References:

1. https://cloud.google.com/logging/docs/logs-based-metrics/
2. https://cloud.google.com/monitoring/custom-metrics/
3. https://cloud.google.com/monitoring/alerts/
4. https://cloud.google.com/logging/docs/reference/tools/gcloud-logging

Notes:

1. Before considering this recommendation Ensure target GCP project is configured with stackdriver account
2. Project Ownership assignments for a user cannot be done using gcloud utility as setting project ownership to a user requires sending and accepting invite
3. Project Ownership assignment to a service account does not sends any invites. SetIAMPolicy to `role/owner`is directly performed on service accounts

CIS Controls:

Version 7

6.2 Activate audit logging

Ensure that local logging has been enabled on all systems and networking devices.

2.5 Ensure log metric filter and alerts exists for Audit Configuration Changes (Scored)

Profile Applicability:

- Level 1

Description:

Google Cloud Platform services write audit log entries to Admin Activity and Data Access logs to helps answer the questions of "who did what, where, and when?" within Google Cloud Platform projects. Cloud Audit logging records information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the GCP services. Cloud Audit logging provides a history of AWS API calls for an account, including API calls made via the Console, SDKs, command line tools, and other GCP services.

Rationale:

Admin activity and Data access logs produced by Cloud audit logging enables security analysis, resource change tracking, and compliance auditing. Configuring metric filter and alerts for Audit Configuration Changes ensures recommended state of audit configuration and hence, all the activities in project are audit-able at any point in time.

Audit:

From GCP Console, Ensure prescribed Log metric present:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`https://console.cloud.google.com/logs/metrics?
2. In `User-defined Metrics` ensure at least one metric `<Log_Metric_Name>` present with filter text:

```
protoPayload.methodName="SetIamPolicy" AND
protoPayload.serviceData.policyDelta.auditConfigDeltas:*
```

From Stackdriver Console, Ensure prescribed Alerting Policy present:

1. Go to stackdriver account at `https://app.google.stackdriver.com/` and select target GCP project on top bar by clicking drop-down arrow symbol.
2. Go to in Left column click `Alerting` select `Policies Overview`

3. on `POLICIES WITH BASIC CONDITIONS` page ensure at least one of the Policies with condition `Violates when: Any logging.googleapis.com/user/<Log Metric Name> stream is above a threshold of .001 for greater than 1 minute` present and state is `ON`

Ensure Alerting Policy Notifications are configured to appropriate subscribers/recipients:

on `POLICIES WITH BASIC CONDITIONS` page click target `Policy Name` to open policy configuration. Check Notifications section for appropriate subscribers/Recipients.

Using CLI, Ensure prescribed Log metric present:

```
gcloud beta logging metrics list --format json
```

Output should contain a metric with filter set to

```
protoPayload.methodName="SetIamPolicy" AND
protoPayload.serviceData.policyDelta.auditConfigDeltas:*
```

property `metricDescriptor.name` for the identified metric that will be used in next step.

Using CLI, Ensure prescribed Alerting Policy present:

```
gcloud alpha monitoring policies list --format json
```

Output should contain an alert policy where:

- `conditions.filter` is set to `"project= "<ProjectID>" AND metric.type="<metricDescriptor.type from previous command output>""`
- AND `conditions.filter` does not contain any other parameter than `metric.type` and `project` which will restrict alerting to a particular resource e.g.. `resource.type`
- AND `conditions.thresholdValue` set to `0.001`
- AND `conditions.conditionThreshold.aggregations.alignmentPeriod` set to `60s`
- AND `conditions.conditionThreshold.aggregations.crossSeriesReducer` set to `REDUCE_COUNT`
- AND `conditions.conditionThreshold.aggregations.perSeriesAligner` set to `ALIGN_RATE`,
- AND `enabled` is set to `true`

Remediation:

From GCP Console, Create prescribed Log Metric:

1. Go to `Logging/Logs` by visiting `https://console.cloud.google.com/logs/viewer?`
2. Click down arrow symbol on `Filter Bar` at rightmost corner and select `Convert to Advanced Filter`
3. This will convert `Filter Bar` to `Advanced Filter Bar`
4. Clear any text from `Advanced Filter` and add text

```
protoPayload.methodName="SetIamPolicy" AND
protoPayload.serviceData.policyDelta.auditConfigDeltas:*
```

5. Click on `Submit Filter` and should display logs based on filter text set in step above
6. Click `Create Metric` it will open `Metric Export` Menu on right
7. Configure `Name`, `Description` to desired values
8. Set `Units` to `1` (default) and `Type` to `Counter`
9. Click `Create Metric`. This will take to `Logging/Logs` at `https://console.cloud.google.com/logs/metrics?`

From Stackdriver Console, Create prescribed Alert Policy:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`
2. In section `User-defined Metrics` for target metric, click 3 dot icon in rightmost column to open menu options
3. Select `Create alert from Metric`.
4. It will take to `Stackdriver Console\Alerting\Create` and directly open `Add Metric Threshold Condition` window

```
Set `Target`: `Resource Type` to `Log Metric`

Set `Configuration`:

- IF METRIC : user/<Log Metric Name>

- Condition : above

- Threshold: .001

- For: 1 minute

Set `Resource` to `ANY`
```

Click `Save Condition`

5. It will take back to `Stackdriver Console\Alerting\Create`
6. In Section `2 Notifications` click `+ Add Notification`. Add desired channel(s) as required.
7. In section `3 Documentation` optionally `+ Add Documentation`
8. In Section `4 Name this policy` leave system provided policy name (Threshold - user/) or configure custom name
9. Click `Save Policy`. It will open `Alerting/Policies Overview` page which lists all the alert policies including this one.
   For alert policy Observe Condition which is currently set to

```
Any logging.googleapis.com/user/<Log_Metric_Name> is above a threshold of
0.001 for greater than 1 minute
```

It is been observed that without following next step, alert policy will not generate any alerts or open any incidents.

10. For newly created policy, click `EDIT` to Open `Edit alerting Policy` pane
11. At section 1 Conditions\Basic Conditions\suggested condition click `Edit`
12. In `Target` Section, for `Aggregation` drop down select `count`
13. Click `save condition` and then click `Save Policy`. It will open `Alerting/Policies Overview`

Condition for same alert policy will be updated to:

```
Violates when: Any logging.googleapis.com/user/<Log_Metric_Name> stream is
above a threshold of 0.001 for greater than 1 minute
```

Now, newly created Alert policy will be able to generate alerts or open incidents.

Using CLI

Create prescribed Log Metric

- Use command: gcloud beta logging metrics create
- Reference for Command Usage: https://cloud.google.com/sdk/gcloud/reference/beta/logging/metrics/create

Create prescribed Alert Policy

- Use command: gcloud alpha monitoring policies create
- Reference for command Usage: https://cloud.google.com/sdk/gcloud/reference/alpha/monitoring/policies/create

Impact:

Based on Service Tiers, Stackdriver account may be charged.

References:

1. https://cloud.google.com/logging/docs/logs-based-metrics/
2. https://cloud.google.com/monitoring/custom-metrics/
3. https://cloud.google.com/monitoring/alerts/
4. https://cloud.google.com/logging/docs/reference/tools/gcloud-logging
5. https://cloud.google.com/logging/docs/audit/configure-data-access#getiampolicy-setiampolicy

Notes:

Before considering this recommendation Ensure target GCP project is configured with stackdriver account.

CIS Controls:

Version 7

6.2 Activate audit logging

Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging

Enable system logging to include detailed information such as a event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

2.6 Ensure log metric filter and alerts exists for Custom Role changes (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended that a metric filter and alarm be established for changes IAM Role creation, deletion and updating activities.

Rationale:

Google Cloud Identity and Access Management (Cloud IAM) provides predefined roles that give granular access to specific Google Cloud Platform resources and prevent unwanted access to other resources. However to cater organization specific needs, Cloud IAM also provides ability to create custom roles. Project Owner and administrators with Organization Role Administrator role or the IAM Role Administrator role can create custom roles. Monitoring role creation, deletion and updating activities will help in identifying over-privileged role at early stages.

Audit:

From GCP Console, Ensure prescribed Log metric present:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`https://console.cloud.google.com/logs/metrics?
2. In `User-defined Metrics` ensure at least one metric `<Log_Metric_Name>` present with filter text:

```
resource.type="iam_role" AND protoPayload.methodName =
"google.iam.admin.v1.CreateRole" OR
protoPayload.methodName="google.iam.admin.v1.DeleteRole" OR
protoPayload.methodName="google.iam.admin.v1.UpdateRole"
```

From Stackdriver Console, Ensure prescribed Alerting Policy present:

1. Go to stackdriver account at `https://app.google.stackdriver.com/` and select target GCP project on top bar by clicking drop-down arrow symbol.
2. Go to in Left column click `Alerting` select `Policies Overview`
3. on `POLICIES WITH BASIC CONDITIONS` page ensure at least one of the Policies with condition `Violates when: Any logging.googleapis.com/user/<Log Metric`

Name> stream is above a threshold of .001 for greater than 1 minute present and state is ON

Ensure Alerting Policy Notifications are configured to appropriate subscribers/recipients:

on POLICIES WITH BASIC CONDITIONS page click target Policy Name to open policy configuration. Check Notifications section for appropriate subscribers/Recipients.

Using CLI, Ensure prescribed Log metric present:

```
gcloud beta logging metrics list --format json
```

Output should contain a metric with filter set to

```
resource.type="iam_role" AND protoPayload.methodName =
"google.iam.admin.v1.CreateRole" OR
protoPayload.methodName="google.iam.admin.v1.DeleteRole" OR
protoPayload.methodName="google.iam.admin.v1.UpdateRole"
```

property metricDescriptor.name for the identified metric that will be used in next step.

Using CLI, Ensure prescribed Alerting Policy present:

```
gcloud alpha monitoring policies list --format json
```

Output should contain an alert policy where:

- conditions.filter is set to "project= "<ProjectID>" AND metric.type="<metricDescriptor.type from previous command output>""
- AND conditions.filter does not contain any other parameter than metric.type and project which will restrict alerting to a particular resource/type e.g.. resource.type
- AND conditions.thresholdValue set to 0.001
- AND conditions.conditionThreshold.aggregations.alignmentPeriod set to 60s
- AND conditions.conditionThreshold.aggregations.crossSeriesReducer set to REDUCE_COUNT
- AND conditions.conditionThreshold.aggregations.perSeriesAligner set to ALIGN_RATE,
- AND enabled is set to true

Remediation:

From GCP Console, Create prescribed Log Metric:

1. Go to Logging/Logs by visiting https://console.cloud.google.com/logs/viewer?
2. Click down arrow symbol on Filter Bar at rightmost corner and select Convert to Advanced Filter
3. This will convert Filter Bar to Advanced Filter Bar
4. Clear any text from Advanced Filter and add text

```
resource.type="iam_role" AND protoPayload.methodName =
"google.iam.admin.v1.CreateRole" OR
protoPayload.methodName="google.iam.admin.v1.DeleteRole" OR
protoPayload.methodName="google.iam.admin.v1.UpdateRole"
```

5. Click on `Submit Filter` and should display logs based on filter text set in step above
6. Click `Create Metric` it will open `Metric Export` Menu on right
7. Configure `Name`, `Description` to desired values
8. Set `Units` to `1` (default) and `Type` to `Counter`
9. Click `Create Metric`. This will take to `Logging/Logs` at `https://console.cloud.google.com/logs/metrics?`

From Stackdriver Console, Create prescribed Alert Policy:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`
2. In section `User-defined Metrics` for target metric, click 3 dot icon in rightmost column to open menu options
3. Select `Create alert from Metric`.
4. It will take to `Stackdriver Console\Alerting\Create` and directly open `Add Metric Threshold Condition` window

```
Set `Target`: `Resource Type` to `Log Metric`

Set `Configuration`:

- IF METRIC : user/<Log Metric Name>

- Condition : above

- Threshold: .001

- For: 1 minute

Set `Resource` to `ANY`
```

Click `Save Condition`

5. It will take back to `Stackdriver Console\Alerting\Create`
6. In Section `2 Notifications` click `+ Add Notification`. Add desired channel(s) as required.
7. In section `3 Documentation` optionally `+ Add Documentation`
8. In Section `4 Name this policy` leave system provided policy name (Threshold - user/) or configure custom name
9. Click `Save Policy`. It will open `Alerting/Policies Overview` page which lists all the alert policies including this one.
   For alert policy Observe Condition which is currently set to

```
Any logging.googleapis.com/user/<Log_Metric_Name> is above a threshold of
0.001 for greater than 1 minute
```

It is been observed that without following next step, Alert policy will not generate any alerts or open any incidents.

10. For newly created policy, click `EDIT` to Open `Edit alerting Policy` pane
11. At section 1 Conditions\Basic Conditions\suggested condition click `Edit`
12. In `Target` Section, for `Aggregation` drop down select `count`
13. Click `save condition` and then click `Save Policy`. It will open `Alerting/Policies Overview`

Condition for same alert policy will be updated to:

```
Violates when: Any logging.googleapis.com/user/<Log_Metric_Name> stream is
above a threshold of 0.001 for greater than 1 minute
```

Now, newly created Alert policy will be able to generate alerts or open incidents.

Using CLI

Create prescribed Log Metric

- Use command: gcloud beta logging metrics create
- Reference for Command Usage: https://cloud.google.com/sdk/gcloud/reference/beta/logging/metrics/create

Create prescribed Alert Policy

- Use command: gcloud alpha monitoring policies create
- Reference for command Usage: https://cloud.google.com/sdk/gcloud/reference/alpha/monitoring/policies/create

Impact:

Based on Service Tiers, Stackdriver account may be charged.

References:

1. https://cloud.google.com/logging/docs/logs-based-metrics/
2. https://cloud.google.com/monitoring/custom-metrics/
3. https://cloud.google.com/monitoring/alerts/
4. https://cloud.google.com/logging/docs/reference/tools/gcloud-logging
5. https://cloud.google.com/iam/docs/understanding-custom-roles

Notes:

Before considering this recommendation Ensure target GCP project is configured with stackdriver account.

CIS Controls:

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as a event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

2.7 Ensure log metric filter and alerts exists for VPC Network Firewall rule changes (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended that a metric filter and alarm be established for VPC Network Firewall rule changes.

Rationale:

Monitoring for Create or Update firewall rule events gives insight network access changes and may reduce the time it takes to detect suspicious activity.

Audit:

From GCP Console, Ensure prescribed Log metric present:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`https://console.cloud.google.com/logs/metrics?
2. In `User-defined Metrics` ensure at least one metric `<Log_Metric_Name>` present with filter text:

```
resource.type="gce_firewall_rule" AND
jsonPayload.event_subtype="compute.firewalls.patch" OR
jsonPayload.event_subtype="compute.firewalls.insert"
```

From Stackdriver Console, Ensure prescribed Alerting Policy present:

1. Go to stackdriver account at `https://app.google.stackdriver.com/` and select target GCP project on top bar by clicking drop-down arrow symbol.
2. Go to in Left column click `Alerting` select `Policies Overview`
3. On `POLICIES WITH BASIC CONDITIONS` page ensure at least one of the Policies with condition `Violates When: Any logging.googleapis.com/user/<Log Metric Name> stream is above a threshold of .001 for greater than 1 minute` present and state is `ON`

Ensure Alerting Policy Notifications are configured to appropriate subscribers/recipients:

on `POLICIES WITH BASIC CONDITIONS` page click target `Policy Name` to open policy

configuration. Check Notifications section for appropriate subscribers/Recipients.

Using CLI, Ensure prescribed Log metric present:

```
gcloud beta logging metrics list --format json
```

Output should contain a metric with filter set to

```
resource.type="gce_firewall_rule" AND
jsonPayload.event_subtype="compute.firewalls.patch" OR
jsonPayload.event_subtype="compute.firewalls.insert"
```

property `metricDescriptor.name` for the identified metric that will be used in next step.

Using CLI, Ensure prescribed Alerting Policy present:

```
gcloud alpha monitoring policies list --format json
```

Output should contain an alert policy where:

- `conditions.filter` is set to `"project= "<ProjectID>" AND metric.type="<metricDescriptor.type from previous command output>""`
- AND `conditions.filter` does not contain any other parameter than `metric.type` and `project` which will restrict alerting to a particular resource/type e.g.. `resource.type`
- AND `conditions.thresholdValue` set to `0.001`
- AND `conditions.conditionThreshold.aggregations.alignmentPeriod` set to `60s`
- AND `conditions.conditionThreshold.aggregations.crossSeriesReducer" set to "REDUCE_COUNT"
- AND `conditions.conditionThreshold.aggregations.perSeriesAligner" set to "ALIGN_RATE",
- AND `enabled` is set to `true`

Remediation:

From GCP Console, Create prescribed Log Metric:

1. Go to `Logging/Logs` by visiting `https://console.cloud.google.com/logs/viewer?`
2. Click down arrow symbol on `Filter Bar` at rightmost corner and select `Convert to Advanced Filter`
3. This will convert `Filter Bar` to `Advanced Filter Bar`
4. Clear any text from `Advanced Filter` and add text

```
resource.type="gce_firewall_rule" AND
jsonPayload.event_subtype="compute.firewalls.patch" OR
jsonPayload.event_subtype="compute.firewalls.insert"
```

5. Click on `Submit Filter` and should display logs based on filter text set in step above
6. Click `Create Metric` it will open `Metric Export` Menu on right

7. Configure `Name`, `Description` to desired values
8. Set `Units` to `1` (default) and `Type` to `Counter`
9. Click `Create Metric`. This will take to `Logging/Logs` at `https://console.cloud.google.com/logs/metrics?`

From Stackdriver Console, Create prescribed Alert Policy:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`
2. In section `User-defined Metrics` for target metric, click 3 dot icon in rightmost column to open menu options
3. Select `Create alert from Metric`.
4. It will take to `Stackdriver Console\Alerting\Create` and directly open `Add Metric Threshold Condition` window

```
Set `Target`: `Resource Type` to `Log Metric`

Set `Configuration`:

- IF METRIC : user/<Log Metric Name>

- Condition : above

- Threshold: .001

- For: 1 minute

Set `Resource` to `ANY`
```

Click `Save Condition`

5. It will take back to `Stackdriver Console\Alerting\Create`
6. In Section `2 Notifications` click `+ Add Notification`. Add desired channel(s) as required.
7. In section `3 Documentation` optionally `+ Add Documentation`
8. In Section `4 Name this policy` leave system provided policy name (Threshold - user/) or configure custom name
9. Click `Save Policy`. It will open `Alerting/Policies Overview` page which lists all the alert policies including this one.
   For alert policy Observe Condition which is currently set to

```
Any logging.googleapis.com/user/<Log_Metric_Name> is above a threshold of
0.001 for greater than 1 minute
```

It is been observed that without following next step, alert policy will not generate any alerts or open any incidents.

10. For newly created policy, click `EDIT` to Open `Edit alerting Policy` pane
11. At section 1 Conditions\Basic Conditions\suggested condition click `Edit`
12. In `Target` Section, for `Aggregation` drop down select `count`
13. Click `save condition` and then click `Save Policy`. It will open `Alerting/Policies Overview`

Condition for same alert policy will be updated to:

```
Violates when: Any logging.googleapis.com/user/<Log_Metric_Name> stream is
above a threshold of 0.001 for greater than 1 minute
```

Now, newly created Alert policy will be able to generate alerts or open incidents.

Using CLI

Create prescribed Log Metric

- Use command: gcloud beta logging metrics create
- Reference for Command Usage: https://cloud.google.com/sdk/gcloud/reference/beta/logging/metrics/create

Create prescribed Alert Policy

- Use command: gcloud alpha monitoring policies create
- Reference for command Usage: https://cloud.google.com/sdk/gcloud/reference/alpha/monitoring/policies/create

Impact:

Based on Service Tiers, Stackdriver account may be charged.

References:

1. https://cloud.google.com/logging/docs/logs-based-metrics/
2. https://cloud.google.com/monitoring/custom-metrics/
3. https://cloud.google.com/monitoring/alerts/
4. https://cloud.google.com/logging/docs/reference/tools/gcloud-logging
5. https://cloud.google.com/vpc/docs/firewalls

Notes:

Before considering this recommendation Ensure target GCP project is configured with stackdriver account.

CIS Controls:

Version 7

6.2 Activate audit logging

Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging

Enable system logging to include detailed information such as a event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

2.8 Ensure log metric filter and alerts exists for VPC network route changes (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended that a metric filter and alarm be established for VPC network route changes.

Rationale:

Google Cloud Platform (GCP) routes define the paths network traffic takes from a VM instance to another destinations. The other destination can be inside your VPC network (such as another VM) or outside of it. Every route consists of a destination and a next hop. Traffic whose destination IP is within the destination range is sent to the next hop for delivery.

Monitoring changes to route tables will help ensure that all VPC traffic flows through an expected path.

Audit:

From GCP Console, Ensure prescribed Log metric present:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`https://console.cloud.google.com/logs/metrics?
2. In `User-defined Metrics` ensure at least one metric `<Log_Metric_Name>` present with filter text:

```
resource.type="gce_route" AND
jsonPayload.event_subtype="compute.routes.delete" OR
jsonPayload.event_subtype="compute.routes.insert"
```

From Stackdriver Console, Ensure prescribed Alerting Policy present:

1. Go to stackdriver account at `https://app.google.stackdriver.com/` and select target GCP project on top bar by clicking drop-down arrow symbol.
2. Go to in Left column click `Alerting` select `Policies Overview`
3. On `POLICIES WITH BASIC CONDITIONS` page ensure at least one of the Policies with condition `Violates when: Any logging.googleapis.com/user/<Log Metric`

Name> stream is above a threshold of .001 for greater than 1 minute present and state is ON

Ensure Alerting Policy Notifications are configured to appropriate subscribers/recipients:

on POLICIES WITH BASIC CONDITIONS page click target Policy Name to open policy configuration. Check Notifications section for appropriate subscribers/Recipients.

Using CLI, Ensure prescribed Log metric present:

```
gcloud beta logging metrics list --format json
```

Output should contain a metric with filter set to

```
resource.type="gce_route" AND
jsonPayload.event_subtype="compute.routes.delete" OR
jsonPayload.event_subtype="compute.routes.insert"
```

property metricDescriptor.name for the identified metric that will be used in next step.

Using CLI, Ensure prescribed Alerting Policy present:

```
gcloud alpha monitoring policies list --format json
```

Output should contain an alert policy where:

- conditions.filter is set to "project= "<ProjectID>" AND metric.type="<metricDescriptor.type from previous command output>""
- AND conditions.filter does not contain any other parameter than metric.type and project which will restrict alerting to a particular resource/type e.g.. resource.type
- AND conditions.thresholdValue set to 0.001
- AND conditions.conditionThreshold.aggregations.alignmentPeriod set to 60s
- AND conditions.conditionThreshold.aggregations.crossSeriesReducer set to REDUCE_COUNT
- AND conditions.conditionThreshold.aggregations.perSeriesAligner set to ALIGN_RATE,
- AND enabled is set to true

Remediation:

From GCP Console, Create prescribed Log Metric:

1. Go to Logging/Logs by visiting https://console.cloud.google.com/logs/viewer?
2. Click down arrow symbol on Filter Bar at rightmost corner and select Convert to Advanced Filter
3. This will convert Filter Bar to Advanced Filter Bar
4. Clear any text from Advanced Filter and add text

```
resource.type="gce_route" AND
jsonPayload.event_subtype="compute.routes.delete" OR
jsonPayload.event_subtype="compute.routes.insert"
```

5. Click on `Submit Filter` and should display logs based on filter text set in step above
6. Click `Create Metric` it will open `Metric Export` Menu on right
7. Configure `Name`, `Description` to desired values
8. Set `Units` to `1` (default) and `Type` to `Counter`
9. Click `Create Metric`. This will take to `Logging/Logs` at `https://console.cloud.google.com/logs/metrics?`

From Stackdriver Console, Create prescribed Alert Policy:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`
2. In section `User-defined Metrics` for target metric, click 3 dot icon in rightmost column to open menu options
3. Select `Create alert from Metric`.
4. It will take to `Stackdriver Console\Alerting\Create` and directly open `Add Metric Threshold Condition` window

```
Set `Target`: `Resource Type` to `Log Metric`

Set `Configuration`:

- IF METRIC : user/<Log Metric Name>

- Condition : above

- Threshold: .001

- For: 1 minute

Set `Resource` to `ANY`
```

Click `Save Condition`

5. It will take back to `Stackdriver Console\Alerting\Create`
6. In Section `2 Notifications` click `+ Add Notification`. Add desired channel(s) as required.
7. In section `3 Documentation` optionally `+ Add Documentation`
8. In Section `4 Name this policy` leave system provided policy name (Threshold - user/) or configure custom name

9. Click `Save Policy`. It will open `Alerting/Policies Overview` page which lists all the alert policies including this one.
   For alert policy Observe Condition which is currently set to

```
Any logging.googleapis.com/user/<Log_Metric_Name> is above a threshold of
0.001 for greater than 1 minute
```

It is been observed that without following next step, Alert policy will not generate any alerts or open any incidents.

10. For newly created policy, click `EDIT` to Open `Edit alerting Policy` pane
11. At section 1 Conditions\Basic Conditions\suggested condition click `Edit`
12. In `Target` Section, for `Aggregation` drop down select `count`
13. Click `save condition` and then click `Save Policy`. It will open `Alerting/Policies Overview`

Condition for same alert policy will be updated to:

```
Violates when: Any logging.googleapis.com/user/<Log_Metric_Name> stream is
above a threshold of 0.001 for greater than 1 minute
```

Now, newly created Alert policy will be able to generate alerts or open incidents.

Using CLI

Create prescribed Log Metric

- Use command: gcloud beta logging metrics create
- Reference for Command Usage:
  https://cloud.google.com/sdk/gcloud/reference/beta/logging/metrics/create

Create prescribed Alert Policy

- Use command: gcloud alpha monitoring policies create
- Reference for command Usage:
  https://cloud.google.com/sdk/gcloud/reference/alpha/monitoring/policies/create

Impact:

Based on Service Tiers, Stackdriver account may be charged.

References:

1. https://cloud.google.com/logging/docs/logs-based-metrics/
2. https://cloud.google.com/monitoring/custom-metrics/
3. https://cloud.google.com/monitoring/alerts/
4. https://cloud.google.com/logging/docs/reference/tools/gcloud-logging
5. https://cloud.google.com/storage/docs/access-control/iam

Notes:

Before considering this recommendation Ensure target GCP project is configured with stackdriver account.

CIS Controls:

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as a event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

2.9 Ensure log metric filter and alerts exists for VPC network changes (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended that a metric filter and alarm be established for VPC network changes.

Rationale:

It is possible to have more than 1 VPC within an project, in addition it is also possible to create a peer connection between 2 VPCs enabling network traffic to route between VPCs.

Monitoring changes to VPC will help ensure VPC traffic flow is not getting impacted.

Audit:

From GCP Console, Ensure prescribed Log metric present:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`https://console.cloud.google.com/logs/metrics?
2. In `User-defined Metrics` ensure at least one metric `<Log_Metric_Name>` present with filter text:

```
resource.type=gce_network AND
jsonPayload.event_subtype="compute.networks.insert" OR
jsonPayload.event_subtype="compute.networks.patch" OR
jsonPayload.event_subtype="compute.networks.delete"  OR
jsonPayload.event_subtype="compute.networks.removePeering" OR
jsonPayload.event_subtype="compute.networks.addPeering"
```

From Stackdriver Console, Ensure prescribed Alerting Policy present:

1. Go to stackdriver account at `https://app.google.stackdriver.com/` and select target GCP project on top bar by clicking drop-down arrow symbol.
2. Go to in Left column click `Alerting` select `Policies Overview`
3. On `POLICIES WITH BASIC CONDITIONS` page ensure at least one of the Policies with condition `Violates When: Any logging.googleapis.com/user/<Log Metric Name> stream is above a threshold of .001 for greater than 1 minute` present and state is `ON`

Ensure Alerting Policy Notifications are configured to appropriate subscribers/recipients:

on `POLICIES WITH BASIC CONDITIONS` page click target `Policy Name` to open policy configuration. Check Notifications section for appropriate subscribers/Recipients.

Using CLI, Ensure prescribed Log metric present:

```
gcloud beta logging metrics list --format json
```

Output should contain a metric with filter set to

```
resource.type=gce_network AND
jsonPayload.event_subtype="compute.networks.insert" OR
jsonPayload.event_subtype="compute.networks.patch" OR
jsonPayload.event_subtype="compute.networks.delete"  OR
jsonPayload.event_subtype="compute.networks.removePeering" OR
jsonPayload.event_subtype="compute.networks.addPeering"
```

property `metricDescriptor.name` for the identified metric that will be used in next step.

Using CLI, Ensure prescribed Alerting Policy present:

```
gcloud alpha monitoring policies list --format json
```

Output should contain an alert policy where:

- `conditions.filter` is set to `"project= "<ProjectID>" AND metric.type="<metricDescriptor.type from previous command output>""`
- AND `conditions.filter` does not contain any other parameter than `metric.type` and `project` which will restrict alerting to a particular resource/type e.g.. `resource.type`
- AND `conditions.thresholdValue` set to `0.001`
- AND `conditions.conditionThreshold.aggregations.alignmentPeriod` set to `60s`
- AND `conditions.conditionThreshold.aggregations.crossSeriesReducer` set to `REDUCE_COUNT`
- AND `conditions.conditionThreshold.aggregations.perSeriesAligner` set to `ALIGN_RATE`,
- AND `enabled` is set to `true`

Remediation:

From GCP Console, Create prescribed Log Metric:

1. Go to `Logging/Logs` by visiting `https://console.cloud.google.com/logs/viewer?`
2. Click down arrow symbol on `Filter Bar` at rightmost corner and select `Convert to Advanced Filter`
3. This will convert `Filter Bar` to `Advanced Filter Bar`
4. Clear any text from `Advanced Filter` and add text

```
resource.type=gce_network AND
jsonPayload.event_subtype="compute.networks.insert" OR
jsonPayload.event_subtype="compute.networks.patch" OR
jsonPayload.event_subtype="compute.networks.delete"  OR
jsonPayload.event_subtype="compute.networks.removePeering" OR
jsonPayload.event_subtype="compute.networks.addPeering"
```

5. Click on `Submit Filter` and should display logs based on filter text set in step above
6. Click `Create Metric` it will open `Metric Export` Menu on right
7. Configure `Name`, `Description` to desired values
8. Set `Units` to `1` (default) and `Type` to `Counter`
9. Click `Create Metric`. This will take to `Logging/Logs` at `https://console.cloud.google.com/logs/metrics?`

From Stackdriver Console, Create prescribed Alert Policy:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`
2. In section `User-defined Metrics` for target metric, click 3 dot icon in rightmost column to open menu options
3. Select `Create alert from Metric`.
4. It will take to `Stackdriver Console\Alerting\Create` and directly open `Add Metric Threshold Condition` window

```
Set `Target`: `Resource Type` to `Log Metric`

Set `Configuration`:

- IF METRIC : user/<Log Metric Name>

- Condition : above

- Threshold: .001

- For: 1 minute

Set `Resource` to `ANY`
```

Click `Save Condition`

5. It will take back to `Stackdriver Console\Alerting\Create`
6. In Section `2 Notifications` click `+ Add Notification`. Add desired channel(s) as required.
7. In section `3 Documentation` optionally `+ Add Documentation`
8. In Section `4 Name this policy` leave system provided policy name (Threshold - user/) or configure custom name

9. Click `Save Policy`. It will open `Alerting/Policies Overview` page which lists all the alert policies including this one.
   For alert policy Observe Condition which is currently set to

```
Any logging.googleapis.com/user/<Log_Metric_Name> is above a threshold of
0.001 for greater than 1 minute
```

It is been observed that without following next step, Alert policy will not generate any alerts or open any incidents.

10. For newly created policy, click `EDIT` to Open `Edit alerting Policy` pane
11. At section 1 Conditions\Basic Conditions\suggested condition click `Edit`
12. In `Target` Section, for `Aggregation` drop down select `count`
13. Click `save condition` and then click `Save Policy`. It will open `Alerting/Policies Overview`

Condition for same alert policy will be updated to:

```
Violates when: Any logging.googleapis.com/user/<Log_Metric_Name> stream is
above a threshold of 0.001 for greater than 1 minute
```

Now, newly created Alert policy will be able to generate alerts or open incidents.

Using CLI

Create prescribed Log Metric

- Use command: gcloud beta logging metrics create
- Reference for Command Usage:
  https://cloud.google.com/sdk/gcloud/reference/beta/logging/metrics/create

Create prescribed Alert Policy

- Use command: gcloud alpha monitoring policies create
- Reference for command Usage:
  https://cloud.google.com/sdk/gcloud/reference/alpha/monitoring/policies/create

Impact:

Based on Service Tiers, Stackdriver account may be charged.

References:

1. https://cloud.google.com/logging/docs/logs-based-metrics/
2. https://cloud.google.com/monitoring/custom-metrics/
3. https://cloud.google.com/monitoring/alerts/
4. https://cloud.google.com/logging/docs/reference/tools/gcloud-logging
5. https://cloud.google.com/vpc/docs/overview

Notes:

Before considering this recommendation Ensure target GCP project is configured with stackdriver account.

CIS Controls:

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as a event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

2.10 Ensure log metric filter and alerts exists for Cloud Storage IAM permission changes (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended that a metric filter and alarm be established for Cloud Storage Bucket IAM changes.

Rationale:

Monitoring changes to Cloud Storage bucket permissions may reduce time to detect and correct permissions on sensitive Cloud Storage bucket and objects inside the bucket.

Audit:

From GCP Console, Ensure prescribed Log metric present:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`https://console.cloud.google.com/logs/metrics?
2. In `User-defined Metrics` ensure at least one metric `<Log_Metric_Name>` present with filter text:

```
resource.type=gcs_bucket AND
protoPayload.methodName="storage.setIamPermissions"
```

From Stackdriver Console, Ensure prescribed Alerting Policy present:

1. Go to stackdriver account at `https://app.google.stackdriver.com/` and select target GCP project on top bar by clicking drop-down arrow symbol.
2. Go to in Left column click `Alerting` select `Policies Overview`
3. On `POLICIES WITH BASIC CONDITIONS` page ensure at least one of the Policies with condition `Violates when: Any logging.googleapis.com/user/<Log Metric Name> stream is above a threshold of .001 for greater than 1 minute` present and state is `ON`

Ensure Alerting Policy Notifications are configured to appropriate subscribers/recipients:

on `POLICIES WITH BASIC CONDITIONS` page click target `Policy Name` to open policy

configuration. Check Notifications section for appropriate subscribers/Recipients.

Using CLI, Ensure prescribed Log metric present:

```
gcloud beta logging metrics list --format json
```

Output should contain a metric with filter set to

```
resource.type=gcs_bucket AND
protoPayload.methodName="storage.setIamPermissions"
```

property `metricDescriptor.name` for the identified metric that will be used in next step.

Using CLI, Ensure prescribed Alerting Policy present:

```
gcloud alpha monitoring policies list --format json
```

Output should contain an alert policy where:

- `conditions.filter` is set to `"project= "<ProjectID>" AND metric.type="<metricDescriptor.type from previous command output>""`
- AND `conditions.filter` does not contain any other parameter than `metric.type` and `project` which will restrict alerting to a particular resource/type e.g.. `resource.type`
- AND `conditions.thresholdValue` set to `0.001`
- AND `conditions.conditionThreshold.aggregations.alignmentPeriod` set to `60s`
- AND `conditions.conditionThreshold.aggregations.crossSeriesReducer` set to `REDUCE_COUNT`
- AND `conditions.conditionThreshold.aggregations.perSeriesAligner` set to `ALIGN_RATE`,
- AND `enabled` is set to `true`

Remediation:

From GCP Console, Create prescribed Log Metric:

1. Go to `Logging/Logs` by visiting `https://console.cloud.google.com/logs/viewer?`
2. Click down arrow symbol on `Filter Bar` at rightmost corner and select `Convert to Advanced Filter`
3. This will convert `Filter Bar` to `Advanced Filter Bar`
4. Clear any text from `Advanced Filter` and add text

```
resource.type=gcs_bucket AND
protoPayload.methodName="storage.setIamPermissions"
```

5. Click on `Submit Filter` and should display logs based on filter text set in step above
6. Click `Create Metric` it will open `Metric Export` Menu on right
7. Configure `Name`, `Description` to desired values

8. Set `Units` to `1` (default) and `Type` to `Counter`
9. Click `Create Metric`. This will take to `Logging/Logs` at `https://console.cloud.google.com/logs/metrics?`

From Stackdriver Console, Create prescribed Alert Policy:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`
2. In section `User-defined Metrics` for target metric, click 3 dot icon in rightmost column to open menu options
3. Select `Create alert from Metric`.
4. It will take to `Stackdriver Console\Alerting\Create` and directly open `Add Metric Threshold Condition` window

```
Set `Target`: `Resource Type` to `Log Metric`

Set `Configuration`:

- IF METRIC : user/<Log Metric Name>

- Condition : above

- Threshold: .001

- For: 1 minute

Set `Resource` to `ANY`
```

Click `Save Condition`

5. It will take back to `Stackdriver Console\Alerting\Create`
6. In Section `2 Notifications` click `+ Add Notification`. Add desired channel(s) as required.
7. In section `3 Documentation` optionally `+ Add Documentation`
8. In Section `4 Name this policy` leave system provided policy name (Threshold - user/) or configure custom name
9. Click `Save Policy`. It will open `Alerting/Policies Overview` page which lists all the alert policies including this one.
   For alert policy Observe Condition which is currently set to

```
Any logging.googleapis.com/user/<Log_Metric_Name> is above a threshold of
0.001 for greater than 1 minute
```

It is been observed that without following next step, Alert policy will not generate any alerts or open any incidents.

10. For newly created policy, click `EDIT` to Open `Edit alerting Policy` pane

11. At section 1 Conditions\Basic Conditions\suggested condition click `Edit`
12. In `Target` Section, for `Aggregation` drop down select `count`
13. Click `save condition` and then click `Save Policy`. It will open `Alerting/Policies Overview`

Condition for same alert policy will be updated to:

```
Violates when: Any logging.googleapis.com/user/<Log_Metric_Name> stream is
above a threshold of 0.001 for greater than 1 minute
```

Now, newly created Alert policy will be able to generate alerts or open incidents.

Using CLI

Create prescribed Log Metric

- Use command: gcloud beta logging metrics create
- Reference for Command Usage: https://cloud.google.com/sdk/gcloud/reference/beta/logging/metrics/create

Create prescribed Alert Policy

- Use command: gcloud alpha monitoring policies create
- Reference for command Usage: https://cloud.google.com/sdk/gcloud/reference/alpha/monitoring/policies/create

Impact:

Based on Service Tiers, Stackdriver account may be charged.

References:

1. https://cloud.google.com/logging/docs/logs-based-metrics/
2. https://cloud.google.com/monitoring/custom-metrics/
3. https://cloud.google.com/monitoring/alerts/
4. https://cloud.google.com/logging/docs/reference/tools/gcloud-logging
5. https://cloud.google.com/storage/docs/overview
6. https://cloud.google.com/storage/docs/access-control/iam-roles

Notes:

Before considering this recommendation Ensure target GCP project is configured with stackdriver account.

CIS Controls:

Version 7

6.2 Activate audit logging

Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging

Enable system logging to include detailed information such as a event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

2.11 Ensure log metric filter and alerts exists for SQL instance configuration changes (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended that a metric filter and alarm be established for SQL Instance configuration changes.

Rationale:

Monitoring changes to Sql Instance configuration changes may reduce time to detect and correct misconfigurations done on sql server.

Below are the few of configurable Options which may impact security posture of a SQL Instance:

- Enable auto backups and high availability: Misconfiguration may adversely impact Business continuity, Disaster Recovery and High Availability
- Authorize networks : Misconfiguration may increase exposure to the untrusted networks

Audit:

From GCP Console, Ensure prescribed Log metric present:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`https://console.cloud.google.com/logs/metrics?
2. In `User-defined Metrics` ensure at least one metric `<Log_Metric_Name>` present with filter text:

```
protoPayload.methodName="cloudsql.instances.update"
```

From Stackdriver Console, Ensure prescribed Alerting Policy present:

1. Go to stackdriver account at `https://app.google.stackdriver.com/` and select target GCP project on top bar by clicking drop-down arrow symbol.
2. Go to in Left column click `Alerting` select `Policies Overview`
3. On `POLICIES WITH BASIC CONDITIONS` page ensure at least one of the Policies with condition `Violates when: Any logging.googleapis.com/user/<Log Metric`

Name> stream is above a threshold of .001 for greater than 1 minute present and state is ON

Ensure Alerting Policy Notifications are configured to appropriate subscribers/recipients:

on POLICIES WITH BASIC CONDITIONS page click target Policy Name to open policy configuration. Check Notifications section for appropriate subscribers/Recipients.

Using CLI, Ensure prescribed Log metric present:

```
gcloud beta logging metrics list --format json
```

Output should contain a metric with filter set to

```
protoPayload.methodName="cloudsql.instances.update"
```

property metricDescriptor.name for the identified metric that will be used in next step.

Using CLI, Ensure prescribed Alerting Policy present:

```
gcloud alpha monitoring policies list --format json
```

Output should contain an alert policy where:

- conditions.filter is set to "project= "<ProjectID>" AND metric.type="<metricDescriptor.type from previous command output>""
- AND conditions.filter does not contain any other parameter than metric.type and project which will restrict alerting to a particular resource/type e.g.. resource.type
- AND conditions.thresholdValue set to 0.001
- AND conditions.conditionThreshold.aggregations.alignmentPeriod set to 60s
- AND conditions.conditionThreshold.aggregations.crossSeriesReducer set to REDUCE_COUNT
- AND conditions.conditionThreshold.aggregations.perSeriesAligner set to ALIGN_RATE,
- AND enabled is set to true

Remediation:

From GCP Console, Create prescribed Log Metric:

1. Go to Logging/Logs by visiting https://console.cloud.google.com/logs/viewer?
2. Click down arrow symbol on Filter Bar at rightmost corner and select Convert to Advanced Filter
3. This will convert Filter Bar to Advanced Filter Bar
4. Clear any text from Advanced Filter and add text

```
protoPayload.methodName="cloudsql.instances.update"
```

5. Click on `Submit Filter` and should display logs based on filter text set in step above
6. Click `Create Metric` it will open `Metric Export` Menu on right
7. Configure `Name`, `Description` to desired values
8. Set `Units` to `1` (default) and `Type` to `Counter`
9. Click `Create Metric`. This will take to `Logging/Logs` at `https://console.cloud.google.com/logs/metrics?`

From Stackdriver Console, Create prescribed Alert Policy:

1. Go to `Logging/Metrics` by visiting `https://console.cloud.google.com/logs/metrics?`
2. In section `User-defined Metrics` for target metric, click 3 dot icon in rightmost column to open menu options
3. Select `Create alert from Metric`.
4. It will take to `Stackdriver Console\Alerting\Create` and directly open `Add Metric Threshold Condition` window

```
Set `Target`: `Resource Type` to `Log Metric`

Set `Configuration`:

- IF METRIC : user/<Log Metric Name>

- Condition : above

- Threshold: .001

- For: 1 minute

Set `Resource` to `ANY`
```

Click `Save Condition`

5. It will take back to `Stackdriver Console\Alerting\Create`
6. In Section `2 Notifications` click `+ Add Notification`. Add desired channel(s) as required.
7. In section `3 Documentation` optionally `+ Add Documentation`
8. In Section `4 Name this policy` leave system provided policy name (Threshold - user/) or configure custom name
9. Click `Save Policy`. It will open `Alerting/Policies Overview` page which lists all the alert policies including this one.
   For alert policy Observe Condition which is currently set to

```
Any logging.googleapis.com/user/<Log_Metric_Name> is above a threshold of
0.001 for greater than 1 minute
```

It is been observed that without following next step, Alert policy will not generate any alerts or open any incidents.

10. For newly created policy, click `EDIT` to Open `Edit alerting Policy` pane
11. At section 1 Conditions\Basic Conditions\suggested condition click `Edit`
12. In `Target` Section, for `Aggregation` drop down select `count`
13. Click `save condition` and then click `Save Policy`. It will open `Alerting/Policies Overview`

Condition for same alert policy will be updated to:

```
Violates when: Any logging.googleapis.com/user/<Log_Metric_Name> stream is
above a threshold of 0.001 for greater than 1 minute
```

Now, newly created Alert policy will be able to generate alerts or open incidents.

Using CLI

Create prescribed Log Metric

- Use command: gcloud beta logging metrics create
- Reference for Command Usage: https://cloud.google.com/sdk/gcloud/reference/beta/logging/metrics/create

Create prescribed Alert Policy

- Use command: gcloud alpha monitoring policies create
- Reference for command Usage: https://cloud.google.com/sdk/gcloud/reference/alpha/monitoring/policies/create

Impact:

Based on Service Tiers, Stackdriver account may be charged.

References:

1. https://cloud.google.com/logging/docs/logs-based-metrics/
2. https://cloud.google.com/monitoring/custom-metrics/
3. https://cloud.google.com/monitoring/alerts/
4. https://cloud.google.com/logging/docs/reference/tools/gcloud-logging
5. https://cloud.google.com/storage/docs/overview
6. https://cloud.google.com/sql/docs/
7. https://cloud.google.com/sql/docs/mysql/
8. https://cloud.google.com/sql/docs/postgres/

Notes:

Before considering this recommendation Ensure target GCP project is configured with stackdriver account.

CIS Controls:

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6.3 Enable Detailed Logging
Enable system logging to include detailed information such as a event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.

3 Networking

This section covers recommendations addressing networking on Google Cloud Platform.

3.1 Ensure the default network does not exist in a project (Scored)

Profile Applicability:

- Level 1

Description:

To prevent use of `default` network, a project should not have a `default` network.

Rationale:

The `default` network has automatically created firewall rules and has pre-fabricated network configuration. Based on your security and networking requirements, you should create your network and delete the `default` network.

Audit:

For each Google Cloud Platform project,

1. Set the project name in the Google Cloud Shell:

```
gcloud config set project <Project-ID>
```

2. List the networks configured in that project:

```
gcloud compute networks list
```

It should not list `default` as one of the available networks in that project.

Remediation:

For each Google Cloud Platform project,

1. Follow the documentation and create a new network suitable for your requirements.
2. Follow the documentation and delete the `default` network.

Impact:

None.

Default Value:

By default, for each project, a `default` network is created.

References:

1. https://cloud.google.com/compute/docs/networking#firewall_rules
2. https://cloud.google.com/compute/docs/reference/latest/networks/insert
3. https://cloud.google.com/compute/docs/reference/latest/networks/delete

CIS Controls:

Version 7

11.1 Maintain Standard Security Configurations for Network Devices

Maintain standard, documented security configuration standards for all authorized network devices.

3.2 Ensure legacy networks does not exists for a project (Scored)

Profile Applicability:

- Level 1

Description:

In order to prevent use of legacy networks, a project should not have a legacy network configured.

Rationale:

Legacy networks have a single network IPv4 prefix range and a single gateway IP address for the whole network. The network is global in scope and spans all cloud regions. You cannot create subnetworks in a legacy network or switch from legacy to auto or custom subnet networks. Legacy networks can thus have an impact for high network traffic projects and subject to the single point of contention or failure.

Audit:

For each Google Cloud Platform project,

1. Set the project name in the Google Cloud Shell:

```
gcloud config set project <Project-ID>
```

2. List the networks configured in that project:

```
gcloud compute networks list
```

None of the listed networks should be in the `legacy` mode.

Remediation:

For each Google Cloud Platform project,

1. Follow the documentation and create a non-legacy network suitable for your requirements.
2. Follow the documentation and delete the networks in the `legacy` mode.

Impact:

None.

Default Value:

By default, networks are not created in the `legacy` mode.

References:

1. https://cloud.google.com/compute/docs/networking#creating_a_legacy_network
2. https://cloud.google.com/compute/docs/networking#legacy_non-subnet_network

CIS Controls:

Version 7

11.1 Maintain Standard Security Configurations for Network Devices

Maintain standard, documented security configuration standards for all authorized network devices.

3.3 Ensure that DNSSEC is enabled for Cloud DNS (Not Scored)

Profile Applicability:

- Level 1

Description:

Cloud DNS is a fast, reliable and cost-effective Domain Name System that powers millions of domains on the internet. DNSSEC in Cloud DNS enables domain owners to take easy steps to protect their domains against DNS hijacking and man-in-the-middle and other attacks.

Rationale:

Domain Name System Security Extensions (DNSSEC) adds security to the Domain Name System (DNS) protocol by enabling DNS responses to be validated. Having a trustworthy Domain Name System (DNS) that translates a domain name like www.example.com into its associated IP address is an increasingly important building block of today's web-based applications. Attackers can hijack this process of domain/IP lookup and redirect users to a malicious site through DNS hijacking and man-in-the-middle attacks. DNSSEC helps mitigate the risk of such attacks by cryptographically signing DNS records. As a result, it prevents attackers from issuing fake DNS responses that may misdirect browsers to nefarious websites.

Audit:

- Go to `Network services`
- For each `Zone name` in `Cloud DNS`
- Ensure `DNSSEC` is set to `On`

Via CLI :
Ensure `state` property in below command's output is `on`

```
gcloud beta dns managed-zones describe <zoneName> --
format="json(dnsName,dnssecConfig.state)"
```

Remediation:

- Go to `Network services`
- For each `Zone name` in `Cloud DNS`
- Set `DNSSEC` to `On`

Via CLI :
Use the below command to enable `DNSSEC` for Cloud DNS Zone Name.

```
gcloud beta dns managed-zones update <zoneName> --dnssec-state on
```

Default Value:

By default DNSSEC is not enabled.

References:

1. https://cloudplatform.googleblog.com/2017/11/DNSSEC-now-available-in-Cloud-DNS.html
2. https://cloud.google.com/dns/dnssec-config#enabling
3. https://cloud.google.com/dns/dnssec

CIS Controls:

Version 7

11.1 Maintain Standard Security Configurations for Network Devices

Maintain standard, documented security configuration standards for all authorized network devices.

3.4 Ensure that RSASHA1 is not used for key-signing key in Cloud DNS DNSSEC (Not Scored)

Profile Applicability:

- Level 1

Description:

DNSSEC algorithm numbers in this registry may be used in CERT RRs. Zone signing (DNSSEC) and transaction security mechanisms (SIG(0) and TSIG) make use of particular subsets of these algorithms. The algorithm used for key signing should be recommended one and it should not be weak.

Rationale:

DNSSEC algorithm numbers in this registry may be used in CERT RRs. Zonesigning (DNSSEC) and transaction security mechanisms (SIG(0) and TSIG) make use of particular subsets of these algorithms.

The algorithm used for key signing should be recommended one and it should not be weak. When enabling DNSSEC for a managed zone, or creating a managed zone with DNSSEC, you can select the DNSSEC signing algorithms and the denial-of-existence type. Changing the DNSSEC settings is only effective for a managed zone if DNSSEC is not already enabled. If you need to change the settings for a managed zone where it has been enabled, you can turn DNSSEC off and then re-enable it with different settings.

Audit:

Currently there is no support to audit this setting through console.
Via CLI : Ensure the property algorithm for keyType keySigning is not using `RSASHA1`.

```
gcloud beta dns managed-zones describe <zoneName> –
format="json(dnsName,dnssecConfig.state,dnssecConfig.defaultKeySpecs)"
```

Remediation:

Use the below command update Cloud DNS managed zone key signing key algorithm to recommended algorithm.
Via CLI :

```
gcloud beta dns managed-zones update EXAMPLE_ZONE --dnssec-state on --ksk-
algorithm ECDSAP256SHA256 --ksk-key-length 256
```

Supported algorithm options and key lengths are as follows.

Algorithm	KSK Length	ZSK Length
RSASHA1	1024,2048	1024,2048
RSASHA256	1024,2048	1024,2048
RSASHA512	1024,2048	1024,2048
ECDSAP256SHA256	256	256
ECDSAP384SHA384	384	384

References:

1. https://cloud.google.com/dns/dnssec-advanced#advanced_signing_options

Notes:

RSASHA1 key-signing support may be required for compatibility reasons

3.5 Ensure that RSASHA1 is not used for zone-signing key in Cloud DNS DNSSEC (Not Scored)

Profile Applicability:

- Level 1

Description:

DNSSEC algorithm numbers in this registry may be used in CERT RRs. Zone signing (DNSSEC) and transaction security mechanisms (SIG(0) and TSIG) make use of particular subsets of these algorithms. The algorithm used for key signing should be recommended one and it should not be weak.

Rationale:

DNSSEC algorithm numbers in this registry may be used in CERT RRs. Zonesigning (DNSSEC) and transaction security mechanisms (SIG(0) and TSIG) make use of particular subsets of these algorithms.

The algorithm used for key signing should be recommended one and it should not be weak. When enabling DNSSEC for a managed zone, or creating a managed zone with DNSSEC, you can select the DNSSEC signing algorithms and the denial-of-existence type. Changing the DNSSEC settings is only effective for a managed zone if DNSSEC is not already enabled. If you need to change the settings for a managed zone where it has been enabled, you can turn DNSSEC off and then re-enable it with different settings.

Audit:

Currently there is no support to audit this setting through console.
Via CLI :
Ensure the property algorithm for keyType zoneSigning is not using RSASHA1.

```
    gcloud beta dns managed-zones describe <zoneName> —
format="json(dnsName,dnssecConfig.state,dnssecConfig.defaultKeySpecs)"
```

Remediation:

Use the below command update Cloud DNS managed zone signing key algorithm to recommended algorithm.
Via CLI :

```
    gcloud beta dns managed-zones update EXAMPLE_ZONE --dnssec-state on --
zsk-algorithm <algorithmName> —zsk-key-length <keyLength>
```

Supported algorithm options and key lengths are as follows.

Algorithm	KSK Length	ZSK Length
RSASHA1	1024,2048	1024,2048
RSASHA256	1024,2048	1024,2048
RSASHA512	1024,2048	1024,2048
ECDSAP256SHA256	256	384
ECDSAP384SHA384	384	384

References:

1. https://cloud.google.com/dns/dnssec-advanced#advanced_signing_options

Notes:

RSASHA1 zone-signing support may be required for compatibility reasons

3.6 Ensure that SSH access is restricted from the internet (Scored)

Profile Applicability:

- Level 2

Description:

GCP `Firewall Rules` are specific to a `VPC Network`. Each rule either `allows` or `denies` traffic when its conditions are met. Its conditions allow you to specify the type of traffic, such as ports and protocols, and the source or destination of the traffic, including IP addresses, subnets, and instances. Firewall rules are defined at the VPC network level, and are specific to the network in which they are defined. The rules themselves cannot be shared among networks. Firewall rules only support IPv4 traffic. When specifying a source for an ingress rule or a destination for an egress rule by address, you can only use an `IPv4` address or `IPv4 block in CIDR` notation. Generic `(0.0.0.0/0)` incoming traffic from internet to VPC or VM instance using `SSH` on `Port 22` can be avoided.

Rationale:

GCP `Firewall Rules` within a `VPC Network`. These rules apply to outgoing (egress) traffic from instances and incoming (ingress) traffic to instances in the network. Egress and ingress traffic are controlled even if the traffic stays within the network (for example, instance-to-instance communication). For an instance to have outgoing Internet access, the network must have a valid Internet gateway route or custom route whose destination IP is specified. This route simply defines the path to the Internet, to avoid the most general `(0.0.0.0/0)` destination `IP Range` specified from Internet through `SSH` with default `Port 22`. We need to restrict generic access from Internet to specific IP Range.

Audit:

From Console

1. Go to `VPC network`.
2. Go to the `Firewall Rules`.
3. Ensure `Port` is not equal to `22` and `Action` is not `Allow`.
4. Ensure `IP Ranges` is not equal to `0.0.0.0` under `Source filters`.

Via CLI gcloud :

```
gcloud compute firewall-rules list --
format=table'(name,direction,sourceRanges,allowed.ports)'
```

Ensure that `SOURCE_RANGES` column in the output of above command does not contan

0.0.0.0 or /0.
If DIRECTION column is INGRESS, PORTS is set to 22 and Range includes 22 against any Firewall Rule.

Remediation:

From Console

1. Go to VPC Network.
2. Go to the Firewall Rules.
3. Click the Firewall Rule you want to modify.
4. Click Edit.
5. Modify Source IP ranges to specific IP.
6. Click Save.

Via CLI gcloud :
1.Update Firewall rule with new SOURCE_RANGE from below command:

```
gcloud compute firewall-rules update FirewallName --allow=[PROTOCOL[:PORT[-
PORT]],...] --source-ranges=[CIDR_RANGE,...]
```

Impact:

All SSH connections from outside of the network to the concerned VPC(s) will be blocked. There could be a business need where ssh access is required from outside of the network to access resources associated with the VPC. In that case, specific source IP(s) should be mentioned in firewall rules to white-list access to SSH port for the concerned VPC(s).

References:

1. https://cloud.google.com/vpc/docs/firewalls#blockedtraffic

Notes:

Currently GCP VPC only supports IPV4 however, Google is already working on adding IPV6 support for VPC. In that case along with source IP range 0.0.0.0, rule should be checked for IPv6 equivalent ::0 as well.

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running
Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

12.4 Deny Communication over Unauthorized Ports

Deny communication over unauthorized TCP or UDP ports or application traffic to ensure that only authorized protocols are allowed to cross the network boundary in or out of the network at each of the organization's network boundaries.

3.7 Ensure that RDP access is restricted from the internet (Scored)

Profile Applicability:

- Level 2

Description:

GCP `Firewall Rules` are specific to a `VPC Network`. Each rule either `allows` or `denies` traffic when its conditions are met. Its conditions allow you to specify the type of traffic, such as ports and protocols, and the source or destination of the traffic, including IP addresses, subnets, and instances. Firewall rules are defined at the VPC network level, and are specific to the network in which they are defined. The rules themselves cannot be shared among networks. Firewall rules only support IPv4 traffic. When specifying a source for an ingress rule or a destination for an egress rule by address, you can only use an `IPv4` address or `IPv4 block in CIDR` notation. Generic `(0.0.0.0/0)` incoming traffic from internet to VPC or VM instance using `RDP` on `Port 3389` can be avoided.

Rationale:

GCP `Firewall Rules` within a `VPC Network`. These rules apply to outgoing (egress) traffic from instances and incoming (ingress) traffic to instances in the network. Egress and ingress traffic are controlled even if the traffic stays within the network (for example, instance-to-instance communication). For an instance to have outgoing Internet access, the network must have a valid Internet gateway route or custom route whose destination IP is specified. This route simply defines the path to the Internet, to avoid the most general `(0.0.0.0/0)` destination `IP Range` specified from Internet through `RDP` with default `Port 3389`. We need to restrict generic access from Internet to specific IP Range.

Audit:

From Console

1. Go to `VPC network`.
2. Go to the `Firewall Rules`.
3. Ensure `Port` is not equal to `3389` and `Action` is not `Allow`.
4. Ensure `IP Ranges` is not equal to `0.0.0.0` under `Source filters`.

Via CLI gcloud :

```
gcloud compute firewall-rules list --
format=table'(name,direction,sourceRanges,allowed.ports)'
```

Ensure that `SOURCE_RANGES` column in the output of above command does not contan

0.0.0.0 or /0.
If DIRECTION column is INGRESS, PORTS is set to 3389 and Range includes 3389 against any Firewall Rule.

Remediation:

From Console

1. Go to VPC Network.
2. Go to the Firewall Rules.
3. Click the Firewall Rule you want to modify.
4. Click Edit.
5. Modify Source IP ranges to specific IP.
6. Click Save.

Via CLI gcloud :
1.Update RDP Firewall rule with new SOURCE_RANGE from below command:

```
gcloud compute firewall-rules update FirewallName --allow=[PROTOCOL[:PORT[-
PORT]],...] --source-ranges=[CIDR_RANGE,...]
```

Impact:

All RDP connections from outside of the network to the concerned VPC(s) will be blocked. There could be a business need where ssh access is required from outside of the network to access resources associated with the VPC. In that case, specific source IP(s) should be mentioned in firewall rules to white-list access to RDP port for the concerned VPC(s).

References:

1. https://cloud.google.com/vpc/docs/firewalls#blockedtraffic

Notes:

Currently GCP VPC only supports IPV4 however, Google is already working on adding IPV6 support for VPC. In that case along with source IP range 0.0.0.0, rule should be checked for IPv6 equivalent ::0 as well.

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running
Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

12.4 Deny Communication over Unauthorized Ports

Deny communication over unauthorized TCP or UDP ports or application traffic to ensure that only authorized protocols are allowed to cross the network boundary in or out of the network at each of the organization's network boundaries.

3.8 Ensure Private Google Access is enabled for all subnetwork in VPC Network (Scored)

Profile Applicability:

- Level 2

Description:

Private Google Access enables virtual machine instances on a subnet to reach Google APIs and services using an internal IP address rather than an external IP address. External IP addresses are routable and reachable over the Internet. Internal (private) IP addresses are internal to Google Cloud Platform and are not routable or reachable over the Internet. You can use Private Google Access to allow VMs without Internet access to reach Google APIs, services, and properties that are accessible over HTTP/HTTPS.

Rationale:

VPC networks and subnetworks provide logically isolated and secure network partitions where you can launch GCP resources. When Private Google Access is enabled, VM instances in a subnet can reach the Google Cloud and Developer APIs and services without needing an external IP address. Instead, VMs can use their internal IP addresses to access Google managed services. Instances with external IP addresses are not affected when you enable the ability to access Google services from internal IP addresses. These instances can still connect to Google APIs and managed services.

Audit:

Using Console:

1. Go to VPC network GCP Console visiting `https://console.cloud.google.com/networking/networks/list`
2. From the list of network subnets,
   make sure for each subnet `Private Google access` is set to `On`

Using Command line:
To check Private Google access status for an existing network subnets, run the following command,

```
gcloud compute networks subnets describe [SUBNET_NAME] --region [REGION] --
format json | jq '.privateIpGoogleAccess'
```

The output of the above command returns `true`, if Private Google access is set to On.
If Private Google access is set to Off above command will return false.

Remediation:

Using Console:

1. Go to VPC network GCP Console visiting `https://console.cloud.google.com/networking/networks/list`
2. Click the name of a subnet, The `Subnet details` page is displayed
3. Click on `EDIT` button
4. Set `Private Google access` to `On`
5. Click on Save

Using Command Line:

To set Private Google access for an network subnets, run the following command:

```
gcloud compute networks subnets update [SUBNET_NAME] --region [REGION] --
enable-private-ip-google-access
```

Impact:

Instances with external IP addresses are not affected when you enable the ability to access Google services from internal IP addresses. These instances can still connect to Google APIs and managed services.

Default Value:

By default, Private Google access is set to Off when you create a new VPC network subnet.

References:

1. https://cloud.google.com/vpc/docs/configure-private-google-access
2. https://cloud.google.com/vpc/docs/private-google-access

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running

Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

12.4 Deny Communication over Unauthorized Ports

Deny communication over unauthorized TCP or UDP ports or application traffic to ensure that only authorized protocols are allowed to cross the network boundary in or out of the network at each of the organization's network boundaries.

3.9 Ensure VPC Flow logs is enabled for every subnet in VPC Network (Scored)

Profile Applicability:

- Level 1

Description:

Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC Subnets. After you've created a flow log, you can view and retrieve its data in Stackdriver Logging. It is recommended that Flow Logs be enabled for every business critical VPC subnet.

Rationale:

VPC networks and subnetworks provide logically isolated and secure network partitions where you can launch GCP resources. When Flow Logs is enabled for a subnet, VMs within subnet starts reporting on all TCP and UDP flows. Each VM samples the TCP and UDP flows it sees, inbound and outbound, whether the flow is to or from another VM, a host in your on-premises datacenter, a Google service, or a host on the Internet. If two GCP VMs are communicating, and both are in subnets that have VPC Flow Logs enabled, both VMs report the flows.

Flow Logs supports following use cases:

- Network monitoring
- Understanding network usage and optimizing network traffic expenses
- Network forensics
- Real-time security analysis

Flow Logs provide visibility into network traffic for each VM inside the subnet and can be used to detect anomalous traffic or insight during security workflows.

Audit:

Using Console:

1. Go to VPC network GCP Console visiting `https://console.cloud.google.com/networking/networks/list`
2. From the list of network subnets,
   make sure for each subnet `Flow Logs` is set to `On`

Using Command line:

```
gcloud compute networks subnets describe [SUBNET_NAME] --region [REGION] --
format json | jq '.enableFlowLogs'
```

The output of the above command returns `true`, if `Flow Logs` is set to On.
If `Flow Logs` is set to Off above command will return false or null (no-output).

Remediation:

Using Console:

1. Go to VPC network GCP Console visiting `https://console.cloud.google.com/networking/networks/list`
2. Click the name of a subnet, The `Subnet details` page is displayed
3. Click on `EDIT` button
4. Set `Flow Logs` to `On`
5. Click on Save

Using Command Line:
To set Private Google access for an network subnets, run the following command:

```
gcloud compute networks subnets update [SUBNET_NAME] --region [REGION] --
enable-flow-logs
```

Impact:

Standard pricing for Stackdriver Logging, BigQuery, or Cloud Pub/Sub apply. VPC flow logs generation will be charged starting in GA as described in reference: https://cloud.google.com/vpc/

Default Value:

By default, Flow Logs is set to Off when you create a new VPC network subnet.

References:

1. https://cloud.google.com/vpc/docs/using-flow-logs#enabling_vpc_flow_logging
2. https://cloud.google.com/vpc/

CIS Controls:

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

12.8 Deploy NetFlow Collection on Networking Boundary Devices
Enable the collection of NetFlow and logging data on all network boundary devices.

4 Virtual Machines

This section covers recommendations addressing virtual machines on Google Cloud Platform.

4.1 Ensure that instances are not configured to use the default service account with full access to all Cloud APIs (Scored)

Profile Applicability:

- Level 1

Description:

To support principle of least privileges and prevent potential privilege escalation it is recommended that instances are not assigned to default service account `Compute Engine default service account` with Scope `Allow full access to all Cloud APIs`.

Rationale:

Along with ability to optionally create, manage and use user managed custom service accounts, Google Compute Engine provides default service account `Compute Engine default service account` for an instances to access necessary cloud services. `Project Editor` role is assigned to `Compute Engine default service account` hence, This service account has almost all capabilities over all cloud services except billing. However, when `Compute Engine default service account` assigned to an instance it can operate in 3 scopes.

```
1. Allow default access: Allows only minimum access required to run an
Instance (Least Privileges)

2. Allow full access to all Cloud APIs: Allow full access to all the cloud
APIs/Services (Too much access)

3. Set access for each API: Allows Instance administrator to choose only
those APIs that are needed to perform specific business functionality
expected by instance
```

When an instance is configured with `Compute Engine default service account` with Scope `Allow full access to all Cloud APIs`, based on IAM roles assigned to the user(s) accessing Instance, it may allow user to perform cloud operations/API calls that user is not supposed to perform leading to successful privilege escalation.

Audit:

From Console:

1. Go to the VM instances page in the Compute Engine using `https://console.cloud.google.com/compute/instances`
2. Click on the VM instance. It will display Instance metadata/properties
3. Scroll down to the Service Account section.

4. Ensure scope `Allow full access to all Cloud APIs` is not selected

via CLI gcloud:

1. List Instances from project

```
gcloud compute instances list
```

2. For every Instance, get Instance metadata

```
gcloud compute instances describe [Instance_Name]
```

3. Ensure instance is do not have section `scopes` containing `https://www.googleapis.com/auth/cloud-platform`

Remediation:

From Console:

1. Go to the VM instances page in the Compute Engine using `https://console.cloud.google.com/compute/instances`
2. Click on the impacted VM instance
3. If the instance is not stopped, click the `Stop` button. Wait for the instance to be stopped.
4. Next, click the `Edit` button.
5. Scroll down to the `Service Account` section.
6. To change scopes, in the `Access scopes` section, set the appropriate scopes as per business needs.
7. Click the `Save` button to save your changes.

via CLI gcloud:

Set service account scope for an instance:

```
gcloud compute instances set-service-account [Instance_name] --service-
account [service_account_email] --scopes [scope1, scope2...]
```

Impact:

In order to change service account or scope for an instance, it needs to be stopped.

Default Value:

While creating an VM instance, default service account is used with scope `Allow default access`.

References:

1. https://cloud.google.com/compute/docs/access/create-enable-service-accounts-for-instances
2. https://cloud.google.com/compute/docs/access/service-accounts

Notes:

- User IAM roles will override service account scope but configuring minimal scope ensures defense in depth
- Non-default service accounts do not offer selection of access scopes like default service account. IAM roles with non-default service accounts should be used to control VM access.

CIS Controls:

Version 7

4.7 Limit Access to Script Tools

Limit access to scripting tools (such as Microsoft PowerShell and Python) to only administrative or development users with the need to access those capabilities.

16 Account Monitoring and Control

Account Monitoring and Control

4.2 Ensure "Block Project-wide SSH keys" enabled for VM instances (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended to user Instance specific SSH key(s) instead of using common/shared project-wide SSH key(s) to access Instances.

Rationale:

Project-wide SSH keys are stored in Compute/Project-meta-data. Project wide SSH keys can be used to login into all the instances within project. Using project-wide SSH keys eases the SSH key management but if compromised, poses the security risk which can impact all the instances within project. It is recommended to use Instance specific SSH keys which can limit the attack surface if the SSH keys are compromised.

Audit:

Using Console:

1. Go to the VM instances page using `https://console.cloud.google.com/compute/instances?`. It will list all the instances from project.
2. Click on the name of the instance
3. Under SSH Keys, Ensure `Block project-wide SSH keys` is selected.
4. Check for every Instance.

via CLI gcloud:

1. List all Instances from a project:

```
gcloud compute instances list
```

2. Get instance metadata

```
gcloud compute instacnces describe [Instance_Name]
```

3. for every instance Ensure `key: block-project-ssh-keys` set to `value: 'true'`

Remediation:

Using Console:

1. Go to the VM instances page using `https://console.cloud.google.com/compute/instances?`. It will list all the instances from project
2. Click on the name of the Impacted instance
3. Click `Edit` in the toolbar
4. Under SSH Keys, go to the `Block project-wide SSH keys` checkbox
5. To block users with project-wide SSH keys from connecting to this instance, select `Block project-wide SSH keys`
6. Click `Save` at the bottom of the page
7. Repeat steps for every impacted Instance

via CLI gcloud:
Block project-wide public SSH keys, set the metadata value to `TRUE`:

```
gcloud compute instances add-metadata [INSTANCE_NAME] --metadata block-
project-ssh-keys=TRUE
```

where [INSTANCE_NAME] is the name of the instance that you want to block project-wide public SSH keys.

Impact:

Users already having Project-wide ssh key pairs and using third party SSH clients will lose access to the impacted Instances. For Project users using gcloud or GCP Console based SSH option, no manual key creation and distribution is required and will be handled by GCE (Google compute Engine) itself. To access Instance using third party SSH clients Instance specific SSH key pairs needs to be created and distributed to the required users.

Default Value:

By Default `Block Project-wide SSH keys` is not enabled.

References:

1. https://cloud.google.com/compute/docs/instances/adding-removing-ssh-keys

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

4.3 Ensure oslogin is enabled for a Project (Scored)

Profile Applicability:

- Level 1

Description:

Enabling OS login binds SSH certificates to IAM users and facilitates effective SSH certificate management.

Rationale:

Enabling osLogin ensures that SSH keys used to connect to instances are mapped with IAM users. Revoking access to IAM user will revoke all the SSH keys associated with that particular user. It facilitates centralized and automated SSH key pair management which is useful in handling cases like response to compromised SSH key pairs and/or revocation of external/third-party/Vendor users.

Audit:

Using Console:

1. Go to the VM compute metadata page using `https://console.cloud.google.com/compute/metadata?`.
2. Ensure key `enable-oslogin` present with value set to `TRUE`

Via CLI gcloud:

```
gcloud compute project-info describe
```

Ensure section `commonInstanceMetadata` has key `enable-oslogin` set to value `TRUE`

Remediation:

Set `enable-oslogin` in project-wide metadata so that it applies to all of the instances in your project:

Using Console:

1. Go to the VM compute metadata page using `https://console.cloud.google.com/compute/metadata?`.
2. Click `Edit`.
3. Add a metadata entry where the key is `enable-oslogin` and the value is `TRUE`.
4. Click `Save` to apply the changes.

Via CLI gcloud:

```
gcloud compute project-info add-metadata --metadata enable-oslogin=TRUE
```

Impact:

Enabling OS Login on project disables metadata-based SSH key configurations on all instances from a project. Disabling OS Login restores SSH keys that you have configured in project or instance meta-data.

Default Value:

By default, parameter `enable-oslogin` is `not set` which is equivalent to setting it to `FALSE`.

References:

1. https://cloud.google.com/compute/docs/instances/managing-instance-access
2. https://cloud.google.com/compute/docs/instances/managing-instance-access#enable_oslogin

Notes:

1. In order to use osLogin, instance using Custom Images must have the latest version of the Linux Guest Environment installed. The following image families do not yet support OS Login:

```
Project cos-cloud (Container-Optimized OS) image family cos-stable.

All project coreos-cloud (CoreOS) image families

Project suse-cloud (SLES) image family sles-11

All Windows Server and SQL Server image families
```

2. Project enable-oslogin can be over-ridden by setting enable-oslogin parameter to an instance metadata individually.

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

4.4 Ensure 'Enable connecting to serial ports' is not enabled for VM Instance (Scored)

Profile Applicability:

- Level 1

Description:

Interacting with a serial port is often referred to as the serial console, which is similar to using a terminal window, in that input and output is entirely in text mode and there is no graphical interface or mouse support.

If you enable the interactive serial console on an instance, clients can attempt to connect to that instance from any IP address. Therefore interactive serial console support should be disabled.

Rationale:

A virtual machine instance has four virtual serial ports. Interacting with a serial port is similar to using a terminal window, in that input and output is entirely in text mode and there is no graphical interface or mouse support. The instance's operating system, BIOS, and other system-level entities often write output to the serial ports, and can accept input such as commands or answers to prompts. Typically, these system-level entities use the first serial port (port 1) and serial port 1 is often referred to as the serial console.

The interactive serial console does not support IP-based access restrictions such as IP whitelists. If you enable the interactive serial console on an instance, clients can attempt to connect to that instance from any IP address. This allows anybody to connect to that instance if they know the correct SSH key, username, project ID, zone, and instance name.

Therefore interactive serial console support should be disabled.

Audit:

1. Login to Google Cloud console
2. Go to Computer Engine
3. Go to VM instances
4. Click on the Specific VM
5. Ensure `Enable connecting to serial ports` below `Remote access` block is unselected.

Via CLI gcloud :

Ensure the below command's output shows `null`

```
gcloud compute instances describe <vmName> --zone=<region> --
format="json(metadata.items[].key,metadata.items[].value)

or

`key` and `value` properties from below command's json response are equal to
`serial-port-  enable` and `0` or `false` respectively.
i,e

{
  "metadata": {
    "items": [
      {
       "key": "serial-port-enable",
        "value": "0"
      }
    ]
 }
}
```

Remediation:

1. Login to Google Cloud console
2. Go to Computer Engine
3. Go to VM instances
4. Click on the Specific VM
5. Click `EDIT`
6. Unselect `Enable connecting to serial ports` below `Remote access` block.
7. Click `Save`

Via CLI gcloud :

Use the below command to disable

```
gcloud compute instances add-metadata <vmName> --zone=<region> --
metadata=serial-port-enable=false

or

gcloud compute instances add-metadata <vmName> --zone=<region> --
metadata=serial-port-enable=0
```

References:

1. https://cloud.google.com/compute/docs/instances/interacting-with-serial-console?hl=en_US&_ga=2.176763621.-1799987798.1507876265

CIS Controls:

Version 7

9.2 Ensure Only Approved Ports, Protocols and Services Are Running

Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.

4.5 Ensure that IP forwarding is not enabled on Instances (Not Scored)

Profile Applicability:

- Level 1

Description:

Compute Engine instance cannot forward a packet unless the source IP address of the packet matches the IP address of the instance. Similarly, GCP won't deliver a packet whose destination IP address is different than the IP address of the instance receiving the packet. However, both capabilities are required if you want to use instances to help route packets.

Forwarding of data packets should be disabled to prevent data loss or information disclosure.

Rationale:

Compute Engine instance cannot forward a packet unless the source IP address of the packet matches the IP address of the instance. Similarly, GCP won't deliver a packet whose destination IP address is different than the IP address of the instance receiving the packet. However, both capabilities are required if you want to use instances to help route packets. To enable this source and destination IP check, disable the `canIpForward` field, which allows an instance to send and receive packets with non-matching destination or source IPs.

Audit:

1. Go to `Compute Engine`
2. Go to the `VM Instances`
3. For every `VM Instance`
4. Ensure `IP forwarding` is set to `Off` under `Network interfaces` section.

Via CLI gcloud :

```
gcloud compute instances list --format='table(name,canIpForward)'
```

Ensure that `CAN_IP_FORWARD` column in the output of above command does not contain `True` against any VM Instance.

Remediation:

1. Go to the `Compute Engine`
2. Go to `VM Instances`

3. Select the `VM Instance`.
4. Click `Delete` button.

Via CLI gcloud :

```
gcloud compute instances delete <VM_Name>
```

As you can only set the `canIpForward` field at instance creation time. After an instance is created, the field becomes read-only. Therefore delete the VM instance where `canIpForward` is set to `true`.
And create a new VM Instance with `IP forwarding` is set to `Off`

1. Go to `Compute Engine`
2. Click the `Create instance` button.
3. Click Management, disk, networking, SSH keys.
4. Click `Networking`.
5. Click on the specific `Network interfaces`
6. Ensure `IP forwarding` is set to `off`.
7. Specify any other instance parameters you desire.
8. Click Create.

Impact:

Deleting instance(s) acting as routers/packet forwarders may break the network connectivity.

References:

1. https://cloud.google.com/compute/docs/networking#canipforward

Notes:

You can only set the `canIpForward` field at instance creation time. After an instance is created, the field becomes read-only.

CIS Controls:

Version 7

11.1 Maintain Standard Security Configurations for Network Devices
Maintain standard, documented security configuration standards for all authorized network devices.

11.2 Document Traffic Configuration Rules
All configuration rules that allow traffic to flow through network devices should be

documented in a configuration management system with a specific business reason for each rule, a specific individual's name responsible for that business need, and an expected duration of the need.

4.6 Ensure VM disks for critical VMs are encrypted with Customer-Supplied Encryption Keys (CSEK) (Scored)

Profile Applicability:

- Level 2

Description:

Customer-Supplied Encryption Keys (CSEK) are a feature in Google Cloud Storage and Google Compute Engine. If you supply your own encryption keys, Google uses your key to protect the Google-generated keys used to encrypt and decrypt your data. By default, Google Compute Engine encrypts all data at rest. Compute Engine handles and manages this encryption for you without any additional actions on your part. However, if you wanted to control and manage this encryption yourself, you can provide your own encryption keys.

Rationale:

By default, Google Compute Engine encrypts all data at rest. Compute Engine handles and manages this encryption for you without any additional actions on your part. However, if you wanted to control and manage this encryption yourself, you can provide your own encryption keys.

If you provide your own encryption keys, Compute Engine uses your key to protect the Google-generated keys used to encrypt and decrypt your data. Only users who can provide the correct key can use resources protected by a customer-supplied encryption key.

Google does not store your keys on its servers and cannot access your protected data unless you provide the key. This also means that if you forget or lose your key, there is no way for Google to recover the key or to recover any data encrypted with the lost key.

At least business critical VMs should have VM disks encrypted with CSEK.

Audit:

GCP Console

1. Go to `Compute Engine`
2. Go to `Disks`
3. For each disk
4. Ensure `Encryption` is set to `Customer supplied`

Via CLI :
Ensure `diskEncryptionKey` property in the below command's response is not null, and contains key `sha256` with corresponding value

```
gcloud beta compute disks describe <diskName> --zone <zone> --
format="json(diskEncryptionKey,name)"
```

Remediation:

Currently there is no way to update the encryption of an existing disk. Therefore create a new disk with `Encryption` set to `Customer supplied`

1. Go to `Compute Engine`
2. Go to `Disks`
3. For each disk
4. Set `Encryption` to `Customer supplied`
5. Provide the `Key` in the box
6. Select `Wrapped key`
7. Click `Create`

Via CLI :
In the gcloud compute tool, encrypt a disk using the --csek-key-file flag during instance creation. If you are using an RSA-wrapped key, use the gcloud beta component:

```
gcloud (beta) compute instances create <instanceName> --csek-key-file
<example-file.json>
```

To encrypt a standalone persistent disk:

```
gcloud (beta) compute disks create <diskName> --csek-key-file <example-
file.json>
```

References:

1. https://cloud.google.com/compute/docs/disks/customer-supplied-encryption#encrypt_a_new_persistent_disk_with_your_own_keys
2. https://cloud.google.com/compute/docs/reference/rest/v1/disks/get
3. https://cloud.google.com/compute/docs/disks/customer-supplied-encryption#key_file

Notes:

`Note 1`: When you delete a persistent disk, Google discards the cipher keys, rendering the data irretrievable. This process is irreversible.

`Note 2`: It is up to you to generate and manage your key. You must provide a key that is a 256-bit string encoded in RFC 4648 standard base64 to Compute Engine.

Note 3: An example key file looks like this.

```
[
  {
  "uri": "https://www.googleapis.com/compute/v1/projects/myproject/zones/us-
central1-a/disks/example-disk",
  "key": "acXTX3rxrKAFTF0tYVLvydU1riRZTvUNC4g5I11NY-c=",
  "key-type": "raw"
  },
  {
  "uri":
"https://www.googleapis.com/compute/v1/projects/myproject/global/snapshots/my
-private-snapshot",
  "key":
"ieCx/NcW06PcT7Ep1X6LUTc/hLvUDYyzSZPPVCVPTVEohpeHASqC8uw5TzyO9U+Fka9JFHz0mBib
XUInrC/jEk014kCK/NPjYgEMOyssZ4ZINPKxlUh2zn1bV+MCaTICrdmuSBTWlUUiFoDD6PYznLwh8
ZNdaheCeZ8ewEXgFQ8V+sDroLaN3Xs3MDTXQEMMoNUXMCZEIpg9Vtp9x2oeQ5lAbtt7bYAAHf5l+g
JWw3sUfs0/Glw5fpdjT8Uggrr+RMZezGrltJEF293rvTIjWOEB3z5OHyHwQkvdrPDFcTqsLfh+8Hr
8g+mf+7zVPEC8nEbqpdl3GPv3A7AwpFp7MA=="
  "key-type": "rsa-encrypted"
  }
]
```

CIS Controls:

Version 7

13 Data Protection
Data Protection

5 Storage

This section covers recommendations addressing storage on Google Cloud Platform.

5.1 Ensure that Cloud Storage bucket is not anonymously or publicly accessible (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended that IAM policy on Cloud Storage bucket does not allows anonymous and/or public access.

Rationale:

Allowing anonymous and/or public access grants permissions to anyone to access bucket content. Such access might not be desired if you are storing any sensitive data. Hence, ensure that anonymous and/or public access to a bucket is not allowed.

Audit:

From Console

1. Go to the Google Cloud Portal
2. Go to `Storage` Section
3. In Storage, Click `Browser`
4. Select each storage bucket and `click on menu` in right most column
5. Select `Edit Bucket Permissions`
6. Expand every role displayed.

No role should have `allUsers` and/or `allAuthenticatedUsers` as a member.
Using Rest API

1. List all buckets in a project

```
Get https://www.googleapis.com/storage/v1/b?project=<ProjectName>
```

2. Check the IAM Policy for each bucket

```
GET https://www.googleapis.com/storage/v1/b/<bucketName>/iam
```

No role should contain `allUsers` and/or `allAuthenticatedUsers` as a member.

Using Command Line

1. List all buckets in a project

```
gsutil ls
```

2. Check the IAM Policy for each bucket

```
gsutil iam get <bucketName>
```

No role should contain `allUsers` and/or `allAuthenticatedUsers` as a member.

Remediation:

From Console

1. Go to the Google Cloud Portal
2. Go to `Storage` Section
3. In Storage, Click `Browser`
4. Select each storage bucket and `click on menu` in right most column
5. Select `Edit Bucket Permissions`
6. Expand every role displayed.
7. Click `Delete` button in front of `allUsers` and/or `allAuthenticatedUsers` to remove that particular role assignment

Impact:

No storage buckets would be publicly accessible. You would have to explicitly administer bucket access.

Default Value:

By Default, Storage buckets are not publicly shared.

References:

1. https://cloud.google.com/storage/docs/access-control/iam-reference
2. https://cloud.google.com/storage/docs/access-control/making-data-public

Notes:

To implement Access restrictions on buckets, configuring Bucket IAM is preferred way than configuring Bucket ACL. On GCP console, "Edit Permissions" for bucket exposes IAM

configurations only. Bucket ACLs are configured automatically as per need in order to implement/support User enforced Bucket IAM policy. In-case administrator changes bucket ACL using command-line(gsutils)/API bucket IAM also gets updated automatically.

CIS Controls:

Version 7

12.4 Deny Communication over Unauthorized Ports

Deny communication over unauthorized TCP or UDP ports or application traffic to ensure that only authorized protocols are allowed to cross the network boundary in or out of the network at each of the organization's network boundaries.

16 Account Monitoring and Control

Account Monitoring and Control

5.2 Ensure that there are no publicly accessible objects in storage buckets (Not Scored)

Profile Applicability:

- Level 1

Description:

It is recommended that storage object ACL should not grant access to "allUsers".

Rationale:

Allowing public access to objects allows anyone with an internet connection to access sensitive data that is important to your business. IAM is used to control access over an entire bucket however to customize access to individual objects within a bucket ACLs are used. Even if IAM applied on storage does not allow access to "allUsers" there could be object specific ACLs that allows public access to the specific objects inside the bucket. Hence it is important to check ACLs at individual object level.

Audit:

Using Console:

1. Go to `console.cloud.google.com`
2. Go to `Storage` Section
3. In Storage, Click `Browser`
4. Click on listed Storage bucket, This will open bucket showing objects and directories inside the bucket
5. For every object at every directory level, check if column "`Share publicly`" is blank. If it represents "`Public Link`", It means object is publicly accessible using Public Link
6. Repeat steps 4, 5 for every storage bucket

Using Rest API:

1. List All the storage buckets by calling: `Get https://www.googleapis.com/storage/v1/b?project=<projectName>`
2. For Every bucket, List objects inside the bucket by calling: `GET https://www.googleapis.com/storage/v1/b/<buvketName>/o/`
3. For Every Object, get object ACL by calling: `GET https://www.googleapis.com/storage/v1/b/pp-test-storage-bucket/o/<objectName>/acl`
4. In Object ACL returned check if `entity "allUsers"` does not exists. If it exists, object is publicly accessible.

Remediation:

Using Console:

1. Go to `console.cloud.google.com`
2. Go to `Storage` Section
3. In Storage, Click `Browser`
4. Click on listed Storage bucket, This will open bucket showing objects and directories inside the bucket
5. For every object at every directory level that is publicly shared, `uncheck checkbox` in column "`Share publicly`". This will remove "`Public Link`" as well.

Impact:

There could be a business need to share/access certain objects publicly. By removing acl with entity "allUsers" will make such object publicly inaccessible.

Default Value:

By Default, object ACLs do not share an object for "allUsers" unless it is inherited from corresponding bucket IAM policy.

References:

1. https://cloud.google.com/storage/docs/access-control/create-manage-lists

Notes:

To provide access restriction for an object, GCP Console has exposed "Edit Permissions" settings which exposed Object ACL configurations. Configuring Object ACL rather than Object IAM is the preferred way. Object IAM will be configured automatically to support enforced object ACL.

CIS Controls:

Version 7

12.4 Deny Communication over Unauthorized Ports

Deny communication over unauthorized TCP or UDP ports or application traffic to ensure that only authorized protocols are allowed to cross the network boundary in or out of the network at each of the organization's network boundaries.

13 Data Protection

Data Protection

5.3 Ensure that logging is enabled for Cloud storage buckets (Scored)

Profile Applicability:

- Level 1

Description:

Storage Access Logging generates a log that contains access records for each request made to the Storage bucket. An access log record contains details about the request, such as the request type, the resources specified in the request worked, and the time and date the request was processed. Cloud Storage offers access logs and storage logs in the form of CSV files that can be downloaded and used for analysis/incident response. Access logs provide information for all of the requests made on a specified bucket and are created hourly, while the daily storage logs provide information about the storage consumption of that bucket for the last day. The access logs and storage logs are automatically created as new objects in a bucket that you specify. An access log record contains details about the request, such as the request type, the resources specified in the request worked, and the time and date the request was processed. While storage Logs helps to keep track the amount of data stored in the bucket. It is recommended that storage Access Logs and Storage logs are enabled for every Storage Bucket.

Rationale:

By enabling access and storage logs on target Storage buckets, it is possible to capture all events which may affect objects within target buckets. Configuring logs to be placed in a separate bucket allows access to log information which can be useful in security and incident response workflows.

In most cases, Cloud Audit Logging is the recommended method for generating logs that track API operations performed in Cloud Storage:

- Cloud Audit Logging tracks access on a continuous basis.
- Cloud Audit Logging produces logs that are easier to work with.
- Cloud Audit Logging can monitor many of your Google Cloud Platform services, not just Cloud Storage.

In some cases, you may want to use access & storage logs instead.

You most likely want to use access logs if:

- You want to track access for public objects.
- You use Access Control Lists (ACLs) to control access to your objects.

- You want to track changes made by the Object Lifecycle Management feature.
- You want your logs to include latency information, or the request and response size of individual HTTP requests.

You most likely want to use storage logs if:

- You want to track the amount of data stored in your buckets.

Audit:

Using Gsutils:

1. To list all the buckets, run command: `gsutil ls`
2. For every bucket, to ensure if storage logging is enabled or not run command: `gsutil logging get gs://<bucketName>/`
3. If output is `"gs://<bucketName>/ has no logging configuration."`, Storage Access Logs and Storage logs are not enabled for a bucket.
4. Expected Output for a bucket with logging enabled: `{"logBucket": "<bucketName for a bucket used to store logs>", "logObjectPrefix": "<prefix set to identify specific storage bucket logs, bucketName by default>"}`

Remediation:

Using Gsutils:
To set Storage Access Logs and Storage logs for a bucket run:

```
gsutil logging set on -b gs://<bucketName for a bucket used to store logs>
gs://<your bucket name>
```

Default Value:

By Default, Access logs and storage logs are not enabled for storage buckets.

References:

1. https://cloud.google.com/storage/docs/access-logs

CIS Controls:

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

6 Cloud SQL Database Services

This section covers security recommendations that you should follow to secure Cloud SQL database services.

6.1 Ensure that Cloud SQL database instance requires all incoming connections to use SSL (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended to enforce all incoming connections to SQL database instance to use SSL.

Rationale:

SQL database connections if successfully trapped (MITM); can reveal sensitive data like credentials, database queries, query outputs etc. For security, it is recommended to always use SSL encryption when connecting to your instance. This recommendation is applicable for Postgresql, MySql generation 1 and MySql generation 2 Instances.

Audit:

Using Command-line:

1. List all SQL database Instances

```
gcloud sql instances list
```

2. Get detailed configuration for every SQL database instance

```
gcloud sql instances describe [INSTANCE_NAME]
```

Ensure that section `settings: ipConfiguration` has parameter `requireSsl` set to `true`.

Remediation:

Using Command-line:
To enforce SSL encryption for an Instance run command:

```
gcloud sql instances patch [INSTANCE_NAME] --require-ssl
```

Note:
`RESTART` is required for type MySQL Generation 1 Instances (`backendType: FIRST_GEN`) to get this configuration in effect.

Impact:

After enforcing SSL connection, existing client will not be able to communicate with SQL server unless configured with appropriate client-certificates to communicate to SQL database instance.

Default Value:

By default parameter `settings: ipConfiguration: requireSsl` is not set which is equivalent to `requireSsl:false`.

References:

1. https://cloud.google.com/sql/docs/postgres/configure-ssl-instance

Notes:

By default `Settings: ipConfiguration` has no `authorizedNetworks` set/configured. In that case even if by default `requireSsl` is not set, which is equivalent to `requireSsl:false` there is no risk as instance cannot be accessed outside of the network unless `authorizedNetworks` are configured. However, If default for `requireSsl` is not updated to `true` any `authorizedNetworks` created later on will not enforce SSL only connection.

CIS Controls:

Version 7

13 Data Protection
Data Protection

14.4 Encrypt All Sensitive Information in Transit
Encrypt all sensitive information in transit.

16.5 Encrypt Transmittal of Username and Authentication Credentials
Ensure that all account usernames and authentication credentials are transmitted across networks using encrypted channels.

6.2 Ensure that Cloud SQL database Instances are not open to the world (Scored)

Profile Applicability:

- Level 1

Description:

Database Server should accept connections only from trusted Network(s)/IP(s) and restrict access from the world.

Rationale:

To minimize attack surface on a Database server Instance, only trusted/known and required IP(s) should be white-listed to connect to it.

Authorized network should not have IPs/networks configured to `0.0.0.0` or `/0` which will allow access to the instance from anywhere in the world.

Audit:

Using Command-line:

1. List all Cloud SQL database Instances

```
gcloud sql instances list
```

2. Get detailed configuration for every Cloud SQL database instance

```
gcloud sql instances describe [INSTANCE_NAME]
```

Ensure that the section `settings: ipConfiguration : authorizedNetworks` does not have any parameter `value` containing `0.0.0.0` or `<Network>/0`.

Remediation:

Using Command-line:
Update the authorized network list by dropping off any addresses

```
gcloud sql instances patch [INSTANCE_NAME] --authorized-
networks=[IP_ADDR1],[IP_ADDR2]...
```

Impact:

The Cloud SQL database instance would not be available to the world.

Default Value:

By default, authorized networks are not configured. Remote connection to Cloud SQL database instance is not possible unless authorized networks are configured.

References:

1. https://cloud.google.com/sql/docs/postgres/configure-ip

Notes:

There is no IPv6 configuration found for Google cloud SQL server services.

CIS Controls:

Version 7

13 Data Protection
Data Protection

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

6.3 Ensure that MySql database instance does not allow anyone to connect with administrative privileges. (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended to set a password for the administrative user (`root` by default) to prevent unauthorized access to the SQL database Instances.

This recommendation is applicable only for MySql Instances. PostgreSQL does not offer any setting for No Password from cloud console.

Rationale:

At the time of MySql Instance creation, not providing a administrative password allows anyone to connect to the SQL database instance with administrative privileges. Root password should be set to ensure only authorized users have these privileges.

Audit:

Using Command Line:

1. List All SQL database instances of type MySql

```
gcloud sql instances list --filter='DATABASE_VERSION:MYSQL*'
```

2. For every MySql instance try to connect from `authorized network`:

```
mysql -u root -h <Instance_IP>
```

Command should return Either Error message or password prompt.
Sample Error message:

```
ERROR 1045 (28000): Access denied for user 'root'@'[Inatance_IP]' (using
password: NO)
```

If command produces `mysql prompt`, SQL Instance allows anyone to connect with administrative privileges without needing password.
Note: `No Password` setting is exposed only at the time of MySql Instance Creation. Once Instance is created, Google cloud UI does not exposes setting to confirm whether password for administrative user is set to a mysql instance.

Remediation:

Using Google Cloud Console:

1. Go to the Cloud SQL Instances page in the Google Cloud Platform Console using `https://console.cloud.google.com/sql/`
2. Select the instance to open its Overview page.
3. Select `Access Control > Users`.
4. Click `more actions icon` for the user you want to update.
5. Select `Change password`, specify a `new password`, and click `OK`.

Using Command-line:

Set password to MySql instance:

```
gcloud sql users set-password [USER_NAME] [HOST] --instance=[INSTANCE_NAME] --password=[PASSWORD]
```

Impact:

Connection strings for administrative clients needs to be reconfigured to use password.

Default Value:

From Google cloud console (UI), `Create Instance` workflow enforces to enter root password unless option `No Password` is selected explicitly.

References:

1. https://cloud.google.com/sql/docs/mysql/create-manage-users
2. https://cloud.google.com/sql/docs/mysql/create-instance

CIS Controls:

Version 7

4.2 Change Default Passwords

Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts.

6.4 Ensure that MySQL Database Instance does not allows root login from any Host (Scored)

Profile Applicability:

- Level 1

Description:

It is recommended that root access to a MySql Database Instance should be allowed only through specific white-listed trusted IPs.

Rationale:

When root access is allowed for any host, any host from authorized networks can attempt to authenticate to a MySql Database Instance using administrative privileges. To minimize attack surface root access can explicitly allowed from only trusted IPs (Hosts) to support database related administrative tasks.

Audit:

Using Google Cloud Console:

1. Go to the MySql Instances page in the Google Cloud Platform Console using `https://console.cloud.google.com/MySql/`
2. Select the instance to open its Overview page.
3. Select `Access Control > Users`.
4. User Name root should not be associated with Host Name containing %(any host) or 0.0.0.0 or /0

Using Command-line:

1. List all MySql database Instances

```
gcloud MySql instances list --filter='DATABASE_VERSION:MYSQL*'
```

2. For Every MySql Database Instance Listed above,

```
 gcloud MySql users list --instance [INSTANCE_NAME]
```

User `root` should not have host configured to `% (any)` or `0.0.0.0` or `/0`

Remediation:

Using Command-line:

Note: We haven't come across any setting provided by Google cloud console or gcloud utility to update host for a root user. Below remediation uses myMySql-client binary to set the host for root user. Similarly, for PostgreMySql instance,

1. Login to MySql database instance from `authorized network`

```
mysql connect -u root [INSTANCE_ADRESS]
```

2. Set host for root user

```
UPDATE MySql.user SET Host=[Host_name/IP] WHERE User='root';
```

Impact:

Only configured hosts will be able access MySql database Instance with root privileges.

Default Value:

By default, `root` access to MySql Database Instance is allowed for any Host.

References:

1. https://cloud.google.com/MySql/docs/MySql/create-manage-users

Notes:

This recommendation is only applicable to MySQL database Instances. As of now for PostgreSQL, google cloud console/gcloud utility does not offer any relevant setting/option.

Apart from default `root` if there are any other database users with administrative privileges, those should be considered for this recommendation as well.

CIS Controls:

Version 7

4 Controlled Use of Administrative Privileges
Controlled Use of Administrative Privileges

14.6 Protect Information through Access Control Lists
Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

7 Kubernetes Engine

This section covers recommendations addressing Google Kubernetes Engine (GKE) on Google Cloud Platform.

7.1 Ensure Stackdriver Logging is set to Enabled on Kubernetes Engine Clusters (Scored)

Profile Applicability:

- Level 1

Description:

Stackdriver Logging is part of the Stackdriver suite of products in Google Cloud Platform. It includes storage for logs, a user interface called the Logs Viewer, and an API to manage logs programmatically. Stackdriver Logging lets you have Kubernetes Engine automatically collect, process, and store your container and system logs in a dedicated, persistent datastore. Container logs are collected from your containers. System logs are collected from the cluster's components, such as docker and kubelet. Events are logs about activity in the cluster, such as the scheduling of Pods.

Rationale:

By Enabling you will have container and system logs, Kubernetes Engine deploys a per-node logging agent that reads container logs, adds helpful metadata, and then stores them. The logging agent checks for container logs in the following sources:

- Standard output and standard error logs from containerized processes
- kubelet and container runtime logs
- Logs for system components, such as VM startup scripts

For events, Kubernetes Engine uses a Deployment in the kube-system namespace which automatically collects events and sends them to Stackdriver Logging.

Stackdriver Logging is compatible with JSON and glog formats. Logs are stored for up to 30 days.

Audit:

Using Console:

1. Go to Kubernetes GCP Console by visiting
   `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster 'Stackdriver Logging' is set to Enabled under Cluster section

Using Command line:

To check logging status for an existing cluster, run the following command,

```
gcloud container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
format json | jq '.loggingService'
```

The output should return `logging.googleapis.com` if logging is Enabled.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting `https://console.cloud.google.com/kubernetes/list?`
2. Select reported Kubernetes clusters for which logging is disabled
3. Click on EDIT button and Set 'Stackdriver Logging' to Enabled

Using Command Line:

To enable logging for an existing cluster, run the following command:

```
gcloud container clusters update [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
logging-service logging.googleapis.com
```

Impact:

You are charged for the accrued storage costs when you export logs to another Google Cloud Platform service, such as BigQuery. Exporting logs from Stackdriver has no Stackdriver charge.

Default Value:

By default, Stackdriver Logging is enabled when you create a new cluster using the gcloud command-line tool or Google Cloud Platform Console.

References:

1. https://cloud.google.com/kubernetes-engine/docs/how-to/creating-a-container-cluster
2. https://cloud.google.com/kubernetes-engine/docs/how-to/logging?hl=en_US
3. https://cloud.google.com/logging/docs/basic-concepts

CIS Controls:

Version 7

6.2 Activate audit logging

Ensure that local logging has been enabled on all systems and networking devices.

7.2 Ensure Stackdriver Monitoring is set to Enabled on Kubernetes Engine Clusters (Scored)

Profile Applicability:

- Level 1

Description:

Stackdriver Monitoring to monitor signals and build operations in your Kubernetes Engine clusters. Stackdriver Monitoring can access metrics about CPU utilization, some disk traffic metrics, network traffic, and uptime information. Stackdriver Monitoring uses the Monitoring agent to access additional system resources and application services in virtual machine instances.

Rationale:

By Enabling Stackdriver Monitoring you will have system metrics and custom metrics. System metrics are measurements of the cluster's infrastructure, such as CPU or memory usage. For system metrics, Stackdriver creates a Deployment that periodically connects to each node and collects metrics about its Pods and containers, then sends the metrics to Stackdriver. Metrics for usage of system resources are collected from the CPU, Memory, Evictable memory, Non-evictable memory, and Disk sources.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting
   `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster 'Stackdriver Monitoring' is set to Enabled under Cluster section

Using Command line:

To check monitoring status for an existing cluster, run the following command,

```
gcloud container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
format json | jq '.monitoringService'
```

The output should return `monitoring.googleapis.com` if monitoring is Enabled.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting
   `https://console.cloud.google.com/kubernetes/list?`
2. Select reported Kubernetes clusters for which monitoring is disabled
3. Click on EDIT button and Set 'Stackdriver Monitoring' to Enabled

Using Command Line:

To enable monitoring for an existing cluster, run the following command:

```
gcloud container clusters update [CLUSTER_NAME] --zone [COMPUTE_ZONE] -
monitoring-service monitoring.googleapis.com
```

Impact:

You are charged for the accrued storage costs when you export logs to another Google Cloud Platform service, such as BigQuery. Exporting logs from Stackdriver has no Stackdriver charge.

Default Value:

By default, Stackdriver Monitoring is enabled when you create a new cluster using the gcloud command-line tool or Google Cloud Platform Console.

References:

1. https://cloud.google.com/kubernetes-engine/docs/how-to/creating-a-container-cluster
2. https://cloud.google.com/kubernetes-engine/docs/how-to/monitoring?hl=en_US
3. https://cloud.google.com/monitoring/agent/

Notes:

If you are using Stackdriver Logging, Stackdriver Error Reporting, Debugging, or Stackdriver Trace, and you are not using any services from Stackdriver Monitoring, then you do not have to associate your GCP project with a Stackdriver account and you do not have to select a service tier. By default, Stackdriver limits the features available to your project to those features in the Basic Tier of service.

CIS Controls:

Version 7

6.2 Activate audit logging
Ensure that local logging has been enabled on all systems and networking devices.

7.3 Ensure Legacy Authorization is set to Disabled on Kubernetes Engine Clusters (Scored)

Profile Applicability:

- Level 1

Description:

In Kubernetes, authorizers interact by granting a permission if any authorizer grants the permission. The legacy authorizer in Kubernetes Engine grants broad, statically defined permissions. To ensure that RBAC limits permissions correctly, you must disable the legacy authorizer. RBAC has significant security advantages, can help you ensure that users only have access to cluster resources within their own namespace and is now stable in Kubernetes.

Rationale:

Enable Legacy Authorization for in-cluster permissions that support existing clusters or workflows. Disable legacy authorization for full RBAC support for in-cluster permissions. In Kubernetes, RBAC is used to grant permissions to resources at the cluster and namespace level. RBAC allows you to define roles with rules containing a set of permissions.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster 'Legacy Authorization' is set to Disabled under Cluster section

Using Command line:
To check Legacy Authorization status for an existing cluster, run the following command:

```
gcloud container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
format json | jq '.legacyAbac'
```

The output should return `null set({})` if Legacy Authorization is Disabled.
If Legacy Authorization is enabled above command will return true value set to enabled.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting
   `https://console.cloud.google.com/kubernetes/list?`
2. Select reported Kubernetes clusters for which Legacy Authorization is enabled
3. Click on EDIT button and Set 'Legacy Authorization' to Disabled

Using Command Line:

To disable Legacy Authorization for an existing cluster, run the following command:

```
gcloud container clusters update [CLUSTER_NAME] --zone [COMPUTE_ZONE] --no-enable-legacy-authorization
```

Impact:

Once the cluster has the legacy authorizer disabled, you must grant your user the ability to create authorization roles to ensure that your role-based access control permissions take effect.

Default Value:

Kubernetes Engine clusters running Kubernetes version 1.8 and later disable the legacy authorization system by default, and thus role-based access control permissions take effect with no special action required.

References:

1. https://cloud.google.com/kubernetes-engine/docs/how-to/role-based-access-control?hl=en_US
2. https://cloud.google.com/kubernetes-engine/docs/how-to/creating-a-container-cluster

Notes:

On clusters running Kubernetes 1.6 or 1.7, Kubernetes service accounts have full permissions on the Kubernetes API by default. To ensure that your role-based access control permissions take effect for a Kubernetes service account, you must create or update your cluster with the option --no-enable-legacy-authorization. This requirement is removed for clusters running Kubernetes version 1.8 or higher.

CIS Controls:

Version 7

4 Controlled Use of Administrative Privileges
Controlled Use of Administrative Privileges

16 Account Monitoring and Control

Account Monitoring and Control

7.4 Ensure Master authorized networks is set to Enabled on Kubernetes Engine Clusters (Not Scored)

Profile Applicability:

- Level 1

Description:

Authorized networks are a way of specifying a restricted range of IP addresses that are permitted to access your container cluster's Kubernetes master endpoint. Kubernetes Engine uses both Transport Layer Security (TLS) and authentication to provide secure access to your container cluster's Kubernetes master endpoint from the public internet. This provides you the flexibility to administer your cluster from anywhere; however, you might want to further restrict access to a set of IP addresses that you control. You can set this restriction by specifying an authorized network.

Rationale:

By Enabling, Master authorized networks blocks untrusted IP addresses from outside Google Cloud Platform and Addresses from inside GCP (such as traffic from Compute Engine VMs) can reach your master through HTTPS provided that they have the necessary Kubernetes credentials.

Restricting access to an authorized network can provide additional security benefits for your container cluster, including:

- Better Protection from Outsider Attacks: Authorized networks provide an additional layer of security by limiting external, non-GCP access to a specific set of addresses you designate, such as those that originate from your premises. This helps protect access to your cluster in the case of a vulnerability in the cluster's authentication or authorization mechanism.
- Better Protection from Insider Attacks: Authorized networks help protect your cluster from accidental leaks of master certificates from your company's premises. Leaked certificates used from outside GCP and outside the authorized IP ranges--for example, from addresses outside your company--are still denied access.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting
   `https://console.cloud.google.com/kubernetes/list?`

2. From the list of clusters,
   make sure for each cluster 'Master authorized networks (beta)' is set to Enabled under Cluster section

Using Command line:

To check Master authorized networks status for an existing cluster, run the following command,

```
gcloud container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
format json | jq '.masterAuthorizedNetworksConfig'
```

The output should return `"enabled": true` in set if Master authorized networks is Enabled.

If Master authorized networks disabled above command will return null set.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting
   `https://console.cloud.google.com/kubernetes/list?`
2. Select reported Kubernetes clusters for which Master authorized networks is disabled
3. Click on EDIT button and Set 'Master authorized networks (beta)' to Enabled

Using Command Line:

To enable Master authorized networks for an existing cluster, run the following command:

```
gcloud container clusters update [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
enable-master-authorized-networks
```

Along with this, you can list authorized networks using the `--master-authorized-networks` flag which contains a list of up to 20 external networks that are allowed to connect to your cluster's Kubernetes master through HTTPS. You provide these networks as a comma-separated list of addresses in CIDR notation (such as 192.168.100.0/24).

Default Value:

By default, Master authorized networks is disabled when you create a new cluster using the gcloud command-line tool or Google Cloud Platform Console.

References:

1. https://cloud.google.com/kubernetes-engine/docs/how-to/authorized-networks?hl=en_US

Notes:

This is a Beta release of Specifying master authorized networks. This feature is not covered by any GCP SLA or deprecation policy and might be subject to backward-incompatible changes.

CIS Controls:

Version 7

14.6 Protect Information through Access Control Lists

Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.

7.5 Ensure Kubernetes Clusters are configured with Labels (Not Scored)

Profile Applicability:

- Level 1

Description:

A cluster label is a key-value pair that helps you organize your Google Cloud Platform resources, such as clusters. You can attach a label to each resource, then filter the resources based on their labels. Information about labels is forwarded to the billing system, so you can break down your billing charges by the label.

Rationale:

Configured Labels can be used to organize and to select subsets of objects. Labels can be attached to objects at creation time and subsequently added and modified at any time. Each object can have a set of key/value labels defined. Each Key must be unique for a given object. Labels enable users to map their own organizational structures onto system objects in a loosely coupled fashion, without requiring clients to store these mappings. Labels can also be used to apply specific security settings and 'auto configure' objects at creation.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting
   `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster the Key and value pair is set under Clusters 'Labels' section

Using Command line:
To check for the existence of Labels for an existing cluster, run the following command,

```
gcloud container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
format json | jq '.resourceLabels'
```

The output should return Key and value pairs in set if Labels are configured.
If Labels are not configured above command will return null set.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting
   `https://console.cloud.google.com/kubernetes/list?`
2. Select reported Kubernetes clusters for which Master authorized networks is disabled
3. Click on EDIT button and Set 'Master authorized networks (beta)' to Enabled

Using Command Line:

To configure Labels for an existing cluster, run the following command:

```
gcloud container clusters update [CLUSTER_NAME] --zone [COMPUTE_ZONE]  --
update-labels [Key]=[Value]
```

Impact:

Any labels you apply to your clusters propagate via a background process that runs hourly. It can take up to one hour for a label to appear on all resources associated with a given cluster.

Default Value:

By default, Labels are not configured when you create a new cluster using the gcloud command-line tool or Google Cloud Platform Console.

References:

1. https://cloud.google.com/kubernetes-engine/docs/how-to/creating-managing-labels?hl=en_US

Notes:

The value of these labels is cloud provider specific and is not guaranteed to be reliable. For example, the value of kubernetes.io/hostname may be the same as the Node name in some environments and a different value in other environments.

7.6 Ensure Kubernetes web UI / Dashboard is disabled (Scored)

Profile Applicability:

- Level 1

Description:

Dashboard is a web-based Kubernetes user interface. You can use Dashboard to deploy containerized applications to a Kubernetes cluster, troubleshoot your containerized application, and manage the cluster itself along with its attendant resources. You can use Dashboard to get an overview of applications running on your cluster, as well as for creating or modifying individual Kubernetes resources (such as Deployments, Jobs, DaemonSets, etc). For example, you can scale a Deployment, initiate a rolling update, restart a pod or deploy new applications using a deploy wizard.

Rationale:

You should disable the Kubernetes Web UI (Dashboard) when running on Kubernetes Engine. The Kubernetes Web UI (Dashboard) is backed by a highly privileged Kubernetes Service Account.

Audit:

From Console

1. Go to `Kubernetes Engine`.
2. Go to `Kubernetes clusters`.
3. For every `Kubernetes cluster`.
4. Click on `Add-ons`.
5. Ensure `Kubernetes dashboard` is `Disabled`

Using Command line:

```
gcloud container clusters describe [CLUSTER_NAME] --zone [ZONE] --format json
| jq '.addonsConfig.kubernetesDashboard'
```

Ensure the output of the above command has JSON key attribute `disabled` set to `true`

```
{
    "disabled": true
    }
```

Remediation:

From Console

1. Go to `Kubernetes Engine`.
2. Go to `Kubernetes clusters`.
3. For every `Kubernetes cluster` click on edit.
4. Click on `Add-ons`.
5. Select `Disabled` from dropdown of `Kubernetes dashboard`.

Using Command line:

To disable the Kubernetes Web UI:

```
gcloud container clusters update [CLUSTER_NAME]  --update-
addons=KubernetesDashboard=DISABLED --zone [ZONE]
```

Default Value:

The Kubernetes web UI (Dashboard) does not have admin access by default in Kubernetes Engine 1.7 and higher. The Kubernetes web UI is disabled by default in Kubernetes Engine 1.10 and higher.

References:

1. https://cloud.google.com/kubernetes-engine/docs/how-to/hardening-your-cluster

CIS Controls:

Version 7

4 Controlled Use of Administrative Privileges
Controlled Use of Administrative Privileges

7.7 Ensure `Automatic node repair` is enabled for Kubernetes Clusters (Scored)

Profile Applicability:

- Level 1

Description:

Kubernetes Engine's node auto-repair feature helps you keep the nodes in your cluster in a healthy, running state. When enabled, Kubernetes Engine makes periodic checks on the health state of each node in your cluster. If a node fails consecutive health checks over an extended time period, Kubernetes Engine initiates a repair process for that node. If you disable node auto-repair at any time during the repair process, the in-progress repairs are not cancelled and still complete for any node currently under repair.

Rationale:

Kubernetes Engine uses the node's health status to determine if a node needs to be repaired. A node reporting a Ready status is considered healthy. Kubernetes Engine triggers a repair action if a node reports consecutive unhealthy status reports for a given time threshold. An unhealthy status can mean:

- A node reports a NotReady status on consecutive checks over the given time threshold (approximately 10 minutes).
- A node does not report any status at all over the given time threshold (approximately 10 minutes).
- A node's boot disk is out of disk space for an extended time period (approximately 30 minutes).

You can enable node auto-repair on a per-node pool basis. When you create a cluster, you can enable or disable auto-repair for the cluster's default node pool. If you create additional node pools, you can enable or disable node auto-repair for those node pools, independent of the auto-repair setting for the default node pool. Kubernetes Engine generates an entry in its operation logs for any automated repair event. You can check the logs by using the gcloud container operations list command.

Audit:

From Console

1. Go to Kubernetes GCP Console by visiting
   `https://console.cloud.google.com/kubernetes/list?`

2. From the list of clusters, make sure for each cluster `Automatic node repair` is set to Enabled under `Node Pools'.

Using Command line:

```
gcloud container node-pools describe default-pool --cluster timos-cluster --
zone us-central1-a --format json | jq '.management'
```

Ensure the output of the above command has JSON key attribute `autoRepair` set to `true`

```
{
  "autoRepair": true
}
```

Remediation:

From Console

1. Go to Kubernetes GCP Console by visiting https://console.cloud.google.com/kubernetes/list?
2. Select reported Kubernetes clusters for which `Automatic node repair` is disabled
3. Click on EDIT button and Set `Automatic node repair` to Enabled

Using Command line:

To enable `Automatic node repair` for an existing cluster with `node pool`, run the following command:

```
gcloud container node-pools update [POOL_NAME] --cluster [CLUSTER_NAME] --
zone [COMPUTE_ZONE] --enable-autorepair
```

Note: Node auto-repair is only available for nodes that use Container-Optimized OS as their node image. Node auto-repair is not available on Alpha Clusters.

Impact:

If multiple nodes require repair, Kubernetes Engine might repair them in parallel. Kubernetes Engine limits number of repairs depending on the size of the cluster (bigger clusters have a higher limit) and the number of broken nodes in the cluster (limit decreases if many nodes are broken)

References:

1. https://cloud.google.com/kubernetes-engine/docs/concepts/node-auto-repair

7.8 Ensure Automatic node upgrades is enabled on Kubernetes Engine Clusters nodes (Scored)

Profile Applicability:

- Level 1

Description:

Node auto-upgrades help you keep the nodes in your cluster or node pool up to date with the latest stable version of Kubernetes. Auto-Upgrades use the same update mechanism as manual node upgrades.

Rationale:

Node pools with auto-upgrades enabled are automatically scheduled for upgrades when a new stable Kubernetes version becomes available. When the upgrade is performed, the node pool is upgraded to match the current cluster master version. Some benefits of using enabling auto-upgrades are:

- Lower management overhead: You don't have to manually track and update to the latest version of Kubernetes.
- Better security: Sometimes new binaries are released to fix a security issue. With auto-upgrades, Kubernetes Engine automatically ensures that security updates are applied and kept up to date.
- Ease of use: Provides a simple way to keep your nodes up to date with the latest Kubernetes features.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster `Automatic node upgrades` is set to Enabled under Node Pools section

Using Command line:
To check existence of Automatic node upgrades for an existing cluster's node pool, run the following command,

```
$ gcloud container node-pools describe [NODE_POOL] --cluster [CLUSTER_NAME] --zone [COMPUTE_ZONE] --format json | jq '.management'
```

Ensure the output of the above command has JSON key attribute `autoUpgrade` set to `true`

```
{
  "autoUpgrade": true
}
```

If autoUpgrade is disabled above command output will not contain the autoUpgrade entry.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting `https://console.cloud.google.com/kubernetes/list?`
2. Select reported Kubernetes clusters for which Automatic node upgrades is disabled
3. Click on EDIT button and Set `Automatic node upgrades` to Enabled

Using Command Line:
To enable Automatic node upgrades for an existing cluster's node pool, run the following command:

```
gcloud container node-pools update [NODE_POOL] --cluster [CLUSTER_NAME] --
zone [COMPUTE_ZONE] --enable-autoupgrade
```

Impact:

Enabling auto-upgrades does not cause your nodes to upgrade immediately. Automatic upgrades occur at regular intervals at the discretion of the Kubernetes Engine team.

Default Value:

By default, subsequent node pools do not have auto-upgrades enabled.

References:

1. https://cloud.google.com/kubernetes-engine/docs/concepts/node-auto-upgrades

Notes:

Node auto-upgrades is not available for Alpha Clusters or clusters running the Ubuntu node image.

7.9 Ensure Container-Optimized OS (cos) is used for Kubernetes Engine Clusters Node image (Not Scored)

Profile Applicability:

- Level 2

Description:

Container-Optimized OS is an operating system image for your Compute Engine VMs that is optimized for running Docker containers. With Container-Optimized OS, you can bring up your Docker containers on Google Cloud Platform quickly, efficiently, and securely.

Rationale:

The Container-Optimized OS node image is based on a recent version of the Linux kernel and is optimized to enhance node security. It is backed by a team at Google that can quickly patch it for security and iterate on features. The Container-Optimized OS image provides better support, security, and stability than previous images. Container-Optimized OS requires Kubernetes version 1.4.0 or higher.

Enabling Container-Optimized OS provides the following benefits:

- Run Containers Out of the Box: Container-Optimized OS instances come pre-installed with the Docker runtime and cloud-init. With a Container-Optimized OS instance, you can bring up your Docker container at the same time you create your VM, with no on-host setup required.
- Smaller attack surface: Container-Optimized OS has a smaller footprint, reducing your instance's potential attack surface.
- Locked-down by default: Container-Optimized OS instances include a locked-down firewall and other security settings by default.
- Automatic Updates: Container-Optimized OS instances are configured to automatically download weekly updates in the background; only a reboot is necessary to use the latest updates.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster nodes `Container-Optimized OS (cos)` is selected under Node Pools section

Using Command line:

To check Node image type for an existing cluster nodes, run the following command:

```
$ gcloud container node-pools describe [NODE_POOL] --cluster [CLUSTER_NAME] -
-zone [COMPUTE_ZONE] --format json | jq '.config.imageType'
```

The output of the above command returns `cos`, if Container-Optimized OS (cos) used for Node images.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting `https://console.cloud.google.com/kubernetes/list?`
2. Select Kubernetes clusters for which Container-Optimized OS (cos) is not used
3. Click on EDIT button and Set `Node image` to `Container-Optimized OS` under Node Pools section

Using Command Line:

To enable Automatic node upgrades for an existing cluster's node pool, run the following command:

```
gcloud container clusters upgrade --image-type cos [CLUSTER_NAME] --zone
[COMPUTE_ZONE] --node-pool [POOL_NAME]
```

Impact:

Upgrade operation is long-running and will block other operations on the cluster (including delete) until it has run to completion.

Default Value:

Container-Optimized OS is the default option for a cluster node image.

References:

1. https://cloud.google.com/kubernetes-engine/docs/concepts/node-images
2. https://cloud.google.com/container-optimized-os/docs/

Notes:

Container-Optimized OS is still under active development, and some of the features and limitations below are subject to change by Google, and functionality should improve in future versions.

7.10 Ensure Basic Authentication is disabled on Kubernetes Engine Clusters (Scored)

Profile Applicability:

- Level 1

Description:

Basic authentication allows a user to authenticate to the cluster with a username and password and it is stored in plain text without any encryption. Disabling Basic authentication will prevent attacks like brute force. Its recommended to use either client certificate or IAM for authentication.

Rationale:

When disabled, you will still be able to authenticate to the cluster with client certificate or IAM. A client certificate is a base64-encoded public certificate used by clients to authenticate to the cluster endpoint. Disable client certificate generation to create a cluster without a client certificate.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters, Click on EDIT button and make sure for each cluster nodes `Basic authentication` is set to Disabled under Clusters section.

Using Command line:
To check Basic authentication status for an existing cluster nodes, run the following command:

```
gcloud container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
format json | jq '.masterAuth.password and .masterAuth.username'
```

The output of the above command should return `false`, if Basic authentication is disabled. If Basic authentication is enabled above command will return true.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting `https://console.cloud.google.com/kubernetes/list?`
2. Select Kubernetes clusters for which Basic authentication is not used
3. Click on EDIT button and Set `Basic authentication` to `Disabled` under Cluster section

No CLI

Default Value:

By default, Basic authentication is enabled when you create a new cluster.

References:

1. https://cloud.google.com/kubernetes-engine/docs/how-to/iam-integration

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

7.11 Ensure Network policy is enabled on Kubernetes Engine Clusters (Scored)

Profile Applicability:

- Level 1

Description:

A network policy is a specification of how groups of pods are allowed to communicate with each other and other network endpoints. NetworkPolicy resources use labels to select pods and define rules which specify what traffic is allowed to the selected pods. The Kubernetes Network Policy API allows the cluster administrator to specify what pods are allowed to communicate with each other.

Rationale:

By default, pods are non-isolated; they accept traffic from any source. Pods become isolated by having a NetworkPolicy that selects them. Once there is any NetworkPolicy in a namespace selecting a particular pod, that pod will reject any connections that are not allowed by any NetworkPolicy. (Other pods in the namespace that are not selected by any NetworkPolicy will continue to accept all traffic.)

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster `Network policy for master` and `Network policy for nodes` are Enabled under Cluster section

Using Command line:

To check Network policy is enabled for an existing cluster, run the following command,

```
gcloud container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
format json | jq '.networkPolicy'
```

Ensure the output of the above command has JSON key attribute `enabled` set to `true`

```
{
  "enabled": true
}
```

If Network policy is disabled above command output will return null.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting `https://console.cloud.google.com/kubernetes/list?`
2. Select Kubernetes clusters for which Network policy is disabled
3. Click on EDIT button and Set `Network policy for master` and `Network policy for nodes` to `Enabled` under Cluster section

Using Command Line:

To enable Network policy for an existing cluster, run the following command:

```
gcloud container clusters update [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
enable-network-policy
```

Impact:

Enabling/Disabling Network Policy causes a rolling update of all cluster nodes, similar to performing a cluster upgrade. This operation is long-running and will block other operations on the cluster (including delete) until it has run to completion.

Default Value:

By default, Network Policy is disabled when you create a new cluster.

References:

1. https://kubernetes.io/docs/concepts/services-networking/network-policies/#the-networkpolicy-resource
2. https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.10/#networkpolicy-v1-networking

Notes:

Google Kubernetes Engine has partnered with Tigera to provide Project Calico to enforce network policies within your cluster. So all above remediation works well only with the same.

CIS Controls:

Version 7

11 Secure Configuration for Network Devices, such as Firewalls, Routers and Switches
Secure Configuration for Network Devices, such as Firewalls, Routers and Switches

14.1 Segment the Network Based on Sensitivity
Segment the network based on the label or classification level of the information stored on the servers, locate all sensitive information on separated Virtual Local Area Networks (VLANs).

7.12 Ensure Kubernetes Cluster is created with Client Certificate enabled (Scored)

Profile Applicability:

- Level 1

Description:

A client certificate is a base64-encoded public certificate used by clients to authenticate to the cluster endpoint.

Rationale:

If you disable client certificate generation to create a cluster without a client certificate. You will still be able to authenticate to the cluster with basic auth or IAM. But basic auth allows a user to authenticate to the cluster with a username and password which are stored in plain text without any encryption and might lead brute force attacks.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting
   `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster `Client certificate` is Enabled under Cluster section

Using Command line:
To check Client certificate is enabled for an existing cluster, run the following command,

```
gcloud container clusters describe change-acc-group-test-cluster --zone us-
central1-a --format json | jq '.masterAuth.clientKey'
```

Ensure the output of the above command should not return null or empty value.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting
   `https://console.cloud.google.com/kubernetes/list?`
2. Click on `CREATE CLUSTER`
3. Choose required name/value for cluster fields
4. Click on `More`

5. Set `Client certificate` to `Enabled`
6. Click on `Create`

Using Command Line:

To enable Network policy for an existing cluster, run the following command:

```
gcloud container clusters create [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
issue-client-certificate
```

Impact:

Cluster create command will create a new cluster with Client certificate enabled, But already existing once are remains unchanged.

Default Value:

By default, the Client certificate is enabled when you create a new cluster.

References:

1. https://cloud.google.com/sdk/gcloud/reference/container/clusters/create

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

7.13 Ensure Kubernetes Cluster is created with Alias IP ranges enabled (Scored)

Profile Applicability:

- Level 1

Description:

Google Cloud Platform Alias IP Ranges lets you assign ranges of internal IP addresses as aliases to a virtual machine's network interfaces. This is useful if you have multiple services running on a VM and you want to assign each service a different IP address.

Rationale:

With Alias IPs ranges enabled, Kubernetes Engine clusters can allocate IP addresses from a CIDR block known to Google Cloud Platform. This makes your cluster more scalable and allows your cluster to better interact with other GCP products and entities. Using Alias IPs has several benefits:

- Pod IPs are reserved within the network ahead of time, which prevents conflict with other compute resources.
- The networking layer can perform anti-spoofing checks to ensure that egress traffic is not sent with arbitrary source IPs.
- Firewall controls for Pods can be applied separately from their nodes.
- Alias IPs allow Pods to directly access hosted services without using a NAT gateway.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster `VPC native (using alias IP)` is Enabled under Cluster section

Using Command line:
To check Alias IP is enabled for an existing cluster, run the following command:

```
gcloud container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
format json | jq '.ipAllocationPolicy.useIpAliases'
```

The output of the above command should return `true`, if VPC native (using alias IP) is

enabled.
If VPC native (using alias IP) is disabled above command will return null.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting `https://console.cloud.google.com/kubernetes/list?`
2. Click on `CREATE CLUSTER`
3. Choose required name/value for cluster fields
4. Click on `More`
5. Set `VPC native (using alias IP)` to `Enabled`
6. Click on `Create`

Using Command Line:
To enable Alias IP for an existing cluster, run the following command:

```
gcloud container clusters create [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
enable-ip-alias
```

Impact:

You cannot currently migrate an existing cluster that uses routes for Pod routing to a cluster that uses Alias IPs. Cluster IPs for internal Services remain only available from within the cluster. If you want to access a Kubernetes Service from within the VPC, but from outside of the cluster, use an internal load balancer.

Default Value:

By default, VPC native (using alias IP) is enabled when you create a new cluster.

References:

1. https://cloud.google.com/kubernetes-engine/docs/how-to/alias-ips
2. https://cloud.google.com/vpc/docs/alias-ip

CIS Controls:

Version 7

11 Secure Configuration for Network Devices, such as Firewalls, Routers and Switches
Secure Configuration for Network Devices, such as Firewalls, Routers and Switches

14.1 Segment the Network Based on Sensitivity
Segment the network based on the label or classification level of the information stored

on the servers, locate all sensitive information on separated Virtual Local Area Networks (VLANs).

7.14 Ensure PodSecurityPolicy controller is enabled on the Kubernetes Engine Clusters (Scored)

Profile Applicability:

- Level 1

Description:

A Pod Security Policy is a cluster-level resource that controls security sensitive aspects of the pod specification. The PodSecurityPolicy objects define a set of conditions that a pod must run with in order to be accepted into the system, as well as defaults for the related fields.

Rationale:

The PodSecurityPolicy defines a set of conditions that Pods must meet to be accepted by the cluster; when a request to create or update a Pod does not meet the conditions in the PodSecurityPolicy, that request is rejected and an error is returned. The PodSecurityPolicy admission controller validates requests against available PodSecurityPolicies. PodSecurityPolicies specify a list of restrictions, requirements, and defaults for Pods created under the policy.

Audit:

Using Command line:
To check Pod Security Policy is enabled for an existing cluster, run the following command,

```
gcloud beta container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE]
--format json | jq '.podSecurityPolicyConfig'
```

Ensure the output of the above command has JSON key attribute `enabled` set to `true`

```
{
  "enabled": true
}
```

If Pod Security Policy is disabled above command output will return null set.

Remediation:

Using Command Line:
To enable Pod Security Policy for an existing cluster, run the following command:

```
gcloud beta container clusters update [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
enable-pod-security-policy
```

Impact:

If you enable the PodSecurityPolicy controller without first defining and authorizing any actual policies, no users, controllers, or service accounts can create or update Pods. If you are working with an existing cluster, you should define and authorize policies before enabling the controller.

Default Value:

By default, Pod Security Policy is disabled when you create a new cluster.

References:

1. https://cloud.google.com/kubernetes-engine/docs/how-to/pod-security-policies
2. https://kubernetes.io/docs/concepts/policy/pod-security-policy

Notes:

This is a Beta release of PodSecurityPolicies in Kubernetes Engine. This feature is not covered by any SLA or deprecation policy and might be subject to backward-incompatible changes.

7.15 Ensure Kubernetes Cluster is created with Private cluster enabled (Scored)

Profile Applicability:

- Level 1

Description:

A private cluster is a cluster that makes your master inaccessible from the public internet. In a private cluster, nodes do not have public IP addresses, so your workloads run in an environment that is isolated from the internet. Nodes have addressed only in the private RFC 1918 address space. Nodes and masters communicate with each other privately using VPC peering.

Rationale:

With a Private cluster enabled, VPC network peering gives you several advantages over using external IP addresses or VPNs to connect networks, including:

- Network Latency: Public IP networking suffers higher latency than private networking.
- Network Security: Service owners do not need to have their services exposed to the public Internet and deal with its associated risks.
- Network Cost: GCP charges egress bandwidth pricing for networks using external IPs to communicate even if the traffic is within the same zone. If however, the networks are peered they can use internal IPs to communicate and save on those egress costs. Regular network pricing still applies to all traffic.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster `Private cluster` is Enabled under Cluster section

Using Command line:
To check Private cluster is enabled for an existing cluster, run the following command,

```
gcloud beta container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE]
--format json | jq '.privateCluster'
```

The output of the above command should return `true`, if the Private cluster is enabled. If the Private cluster is disabled above command will return null.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting `https://console.cloud.google.com/kubernetes/list?`
2. Click on `CREATE CLUSTER`
3. Choose required name/value for cluster fields
4. Click on `More`
5. From the `Private cluster` drop-down menu, select `Enabled`
6. Verify that `VPC native (alias IP)` is set to `Enabled`
7. Set `Master IP range` to as per your required IP range
8. Click on `Create`

Using Command Line:

To create cluster with Private cluster enabled, run the following command:

```
gcloud beta container clusters create [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
private-cluster --master-ipv4-cidr 172.16.0.16/28 --enable-ip-alias --create-
subnetwork ""
```

NOTE: When you create a private cluster, you must specify a /28 CIDR range for the VMs that run the Kubernetes master components. You also need to enable Alias IPs. The range you specify for the masters must not overlap with any subnet in your cluster's VPC.

Default Value:

By default, Private cluster is disabled when you create a new cluster.

References:

1. https://cloud.google.com/kubernetes-engine/docs/how-to/private-clusters

Notes:

This is a Beta release of private clusters. This feature is not covered by any SLA or deprecation policy and might be subject to backward-incompatible changes.

CIS Controls:

Version 7

11 Secure Configuration for Network Devices, such as Firewalls, Routers and Switches
Secure Configuration for Network Devices, such as Firewalls, Routers and Switches

7.16 Ensure Private Google Access is set on Kubernetes Engine Cluster Subnets (Scored)

Profile Applicability:

- Level 1

Description:

Private Google Access enables your cluster hosts, which have only private IP addresses, to communicate with Google APIs and services using an internal IP address rather than an external IP address. External IP addresses are routable and reachable over the Internet. Internal (private) IP addresses are internal to Google Cloud Platform and are not routable or reachable over the Internet. You can use Private Google Access to allow VMs without Internet access to reach Google APIs, services, and properties that are accessible over HTTP/HTTPS.

Rationale:

VPC networks and subnetworks provide logically isolated and secure network partitions where you can launch GCP resources. When Private Google Access is enabled, VM instances in a subnet can reach the Google Cloud and Developer APIs and services without needing an external IP address. Instead, VMs can use their internal IP addresses to access Google managed services. Instances with external IP addresses are not affected when you enable the ability to access Google services from internal IP addresses. These instances can still connect to Google APIs and managed services.

Audit:

Using Console:

1. Go to Kubernetes GCP Console visiting `https://console.cloud.google.com/kubernetes/list`
2. From the list of clusters, for each clusters note the `Subnet` name
3. Go to VPC network GCP Console visiting `https://console.cloud.google.com/networking/networks/list`
4. From the list of network subnets, choose noted subnet and make sure subnet has `Private Google access` is set to `On`

Using Command line:

To get subnet name of the Cluster, run the following command:

```
gcloud beta container clusters describe [CLUSTER_NAME] --zone [COMPUTE_ZONE]
--format json | jq '.subnetwork'
```

The command will return you subnet name.
Note down the subnet name and mention the same name at following command which will check Private Google access status, run the following command:

```
gcloud compute networks subnets describe [SUBNET_NAME] --region [REGION] --
format json | jq '.privateIpGoogleAccess'
```

The output of the above command returns `true`, if Private Google access is set on Cluster subnetwork.
If Private Google access is set to Off above command will return false.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console visiting `https://console.cloud.google.com/kubernetes/list`
2. From the list of clusters, for each clusters note the `Subnet` name
3. Go to VPC network GCP Console visiting `https://console.cloud.google.com/networking/networks/list`
4. Click noted subnet, The `Subnet details` page is displayed
5. Click on Edit button
6. Set `Private Google access` to `On`
7. Click on Save

Using Command Line:
To set Private Google access for a network subnet, run the following command:

```
gcloud compute networks subnets update [SUBNET_NAME] --region [REGION] --
enable-private-ip-google-access
```

Impact:

Instances with external IP addresses are not affected when you enable the ability to access Google services from internal IP addresses. These instances can still connect to Google APIs and managed services.

Default Value:

By default, Private Google access is set to Off when you create a new cluster/cluster subnetwork.

References:

1. https://cloud.google.com/vpc/docs/configure-private-google-access

2. https://cloud.google.com/vpc/docs/private-google-access

CIS Controls:

Version 7

11 Secure Configuration for Network Devices, such as Firewalls, Routers and Switches
Secure Configuration for Network Devices, such as Firewalls, Routers and Switches

12 Boundary Defense
Boundary Defense

7.17 Ensure default Service account is not used for Project access in Kubernetes Clusters (Scored)

Profile Applicability:

- Level 2

Description:

A service account is an identity that an instance or an application can use to run API requests on your behalf. This identity is used to identify applications running on your virtual machine instances to other Google Cloud Platform services. By default, Kubernetes Engine nodes are given the Compute Engine default service account. This account has broad access by default, making it useful to a wide variety of applications, but it has more permissions than are required to run your Kubernetes Engine cluster.

Rationale:

You should create and use a minimally privileged service account to run your Kubernetes Engine cluster instead of using the Compute Engine default service account. If you are not creating a separate service account for your nodes, you should limit the scopes of the node service account to reduce the possibility of a privilege escalation in an attack. Kubernetes Engine requires, at a minimum, the service account to have the monitoring.viewer, monitoring.metricWriter, and logging.logWriter roles. This ensures that your default service account does not have permissions beyond those necessary to run your cluster. While the default scopes are limited, they may include scopes beyond the minimally required scopes needed to run your cluster.

Audit:

Using Console:

1. Go to Kubernetes GCP Console by visiting `https://console.cloud.google.com/kubernetes/list?`
2. From the list of clusters,
   make sure for each cluster `Service account` is not set to `Compute Engine default service account` under Permissions drop-down

Using Command line:
To check which Service account is set for an existing cluster, run the following command:

```
$ gcloud container node-pools describe [NODE_POOL] --cluster [CLUSTER_NAME] --zone [COMPUTE_ZONE] --format json | jq '.config.serviceAccount'
```

The output of the above command will return `default` if default Service account is used for Project access.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting `https://console.cloud.google.com/kubernetes/list?`
2. Click on `CREATE CLUSTER`
3. Choose required name/value for cluster fields
4. Click on `More`
5. Choose `Service account` which has the least privilege under Project access section, Instead of default `Compute Engine default service account`

NOTE: The default scopes for the nodes in Kubernetes Engine are `devstorage.read_only`, `logging.write`, `monitoring`, `service.management.readonly`, `servicecontrol`, and `trace.append`. If you are accessing private images in Google Container Registry, the minimally required scopes are only `logging.write`, `monitoring`, and `devstorage.read_only`.
You can configure a service account with minimal privileges and assign the same.

6. Click on `Create`

Using Command Line:
To create a new node pool with service account with least privilege:, run the following command:

```
gcloud container node-pools create [NODE_POOL] --service-
account=[SA_NAME]@[PROJECT_ID].iam.gserviceaccount.com" --
cluster=[CLUSTER_NAME] --zone [COMPUTE_ZONE]
``
```

Default Value:

By default, Compute Engine default service account is chosen when you create a new cluster.

References:

1. https://cloud.google.com/compute/docs/access/service-accounts#compute_engine_default_service_account

CIS Controls:

Version 7

4 Controlled Use of Administrative Privileges
Controlled Use of Administrative Privileges

16 Account Monitoring and Control
Account Monitoring and Control

7.18 Ensure Kubernetes Clusters created with limited service account Access scopes for Project access (Scored)

Profile Applicability:

- Level 1

Description:

Access scopes are the legacy method of specifying permissions for your instance. Before the existence of IAM roles, access scopes were the only mechanism for granting permissions to service accounts. By default, your node service account has access scopes.

Rationale:

If you are not creating a separate service account for your nodes, you should limit the scopes of the node service account to reduce the possibility of a privilege escalation in an attack. This ensures that your default service account does not have permissions beyond those necessary to run your cluster. While the default scopes are limited, they may include scopes beyond the minimally required scopes needed to run your cluster.

Audit:

Using Command line:
To check Access scopes set for an existing cluster, run the following command:

```
gcloud container node-pools describe [NODE_NAME] --cluster [CLUSTER_NAME] --
zone [COMPUTE_ZONE] --format json | jq '.config.oauthScopes'
```

The output of the above command will return array set access scopes. Make sure you have provided limited required scopes for each node clusters.
If you are accessing private images in Google Container Registry, the minimally required scopes are only `logging.write`, `monitoring`, and `devstorage.read_only`.

Remediation:

Using Console:

1. Go to Kubernetes GCP Console by visiting `https://console.cloud.google.com/kubernetes/list?`
2. Click on `CREATE CLUSTER`
3. Choose required name/value for cluster fields
4. Click on `More`

5. Under `Access scopes` select `Set access for each API` and choose minimal API access as you desired
6. Click on `Create`

Using Command Line:

To create a cluster with least privileged/Custom Access scopes:, run the following command:

```
gcloud container clusters create [CLUSTER_NAME] --zone [COMPUTE_ZONE] --
scopes=[CUSTOM_SCOPES]
```

NOTE: The default scopes for the nodes in Kubernetes Engine are `devstorage.read_only`, `logging.write`, `monitoring`, `service.management.readonly`, `servicecontrol`, and `trace.append`. When setting scopes, these are specified as `gke-default`. If you are accessing private images in Google Container Registry, the minimally required scopes are only `logging.write`, `monitoring`, and `devstorage.read_only`.

Default Value:

By default, 'Allow default access' is chosen under Access scopes when you create a new cluster.

References:

1. https://cloud.google.com/compute/docs/access/service-accounts?hl=en_US#the_default_service_account

CIS Controls:

Version 7

16 Account Monitoring and Control
Account Monitoring and Control

Appendix: Summary Table

Control		Set Correctly	
		Yes	No
1	**Identity and Access Management**		
1.1	Ensure that corporate login credentials are used instead of Gmail accounts (Scored)	☐	☐
1.2	Ensure that multi-factor authentication is enabled for all non-service accounts (Not Scored)	☐	☐
1.3	Ensure that there are only GCP-managed service account keys for each service account (Scored)	☐	☐
1.4	Ensure that ServiceAccount has no Admin privileges. (Scored)	☐	☐
1.5	Ensure that IAM users are not assigned Service Account User role at project level (Scored)	☐	☐
1.6	Ensure user-managed/external keys for service accounts are rotated every 90 days or less (Scored)	☐	☐
1.7	Ensure that Separation of duties is enforced while assigning service account related roles to users (Not Scored)	☐	☐
1.8	Ensure Encryption keys are rotated within a period of 365 days (Scored)	☐	☐
1.9	Ensure that Separation of duties is enforced while assigning KMS related roles to users (Scored)	☐	☐
1.10	Ensure API keys are not created for a project (Not Scored)	☐	☐
1.11	Ensure API keys are restricted to use by only specified Hosts and Apps (Not Scored)	☐	☐
1.12	Ensure API keys are restricted to only APIs that application needs access (Not Scored)	☐	☐
1.13	Ensure API keys are rotated every 90 days (Scored)	☐	☐
2	**Logging and Monitoring**		
2.1	Ensure that Cloud Audit Logging is configured properly across all services and all users from a project (Scored)	☐	☐
2.2	Ensure that sinks are configured for all Log entries (Scored)	☐	☐
2.3	Ensure that object versioning is enabled on log-buckets (Scored)	☐	☐
2.4	Ensure log metric filter and alerts exists for Project Ownership assignments/changes (Scored)	☐	☐
2.5	Ensure log metric filter and alerts exists for Audit Configuration Changes (Scored)	☐	☐
2.6	Ensure log metric filter and alerts exists for Custom Role changes (Scored)	☐	☐
2.7	Ensure log metric filter and alerts exists for VPC Network Firewall rule changes (Scored)	☐	☐

2.8	Ensure log metric filter and alerts exists for VPC network route changes (Scored)	☐	☐
2.9	Ensure log metric filter and alerts exists for VPC network changes (Scored)	☐	☐
2.10	Ensure log metric filter and alerts exists for Cloud Storage IAM permission changes (Scored)	☐	☐
2.11	Ensure log metric filter and alerts exists for SQL instance configuration changes (Scored)	☐	☐
3	**Networking**		
3.1	Ensure the default network does not exist in a project (Scored)	☐	☐
3.2	Ensure legacy networks does not exists for a project (Scored)	☐	☐
3.3	Ensure that DNSSEC is enabled for Cloud DNS (Not Scored)	☐	☐
3.4	Ensure that RSASHA1 is not used for key-signing key in Cloud DNS DNSSEC (Not Scored)	☐	☐
3.5	Ensure that RSASHA1 is not used for zone-signing key in Cloud DNS DNSSEC (Not Scored)	☐	☐
3.6	Ensure that SSH access is restricted from the internet (Scored)	☐	☐
3.7	Ensure that RDP access is restricted from the internet (Scored)	☐	☐
3.8	Ensure Private Google Access is enabled for all subnetwork in VPC Network (Scored)	☐	☐
3.9	Ensure VPC Flow logs is enabled for every subnet in VPC Network (Scored)	☐	☐
4	**Virtual Machines**		
4.1	Ensure that instances are not configured to use the default service account with full access to all Cloud APIs (Scored)	☐	☐
4.2	Ensure "Block Project-wide SSH keys" enabled for VM instances (Scored)	☐	☐
4.3	Ensure oslogin is enabled for a Project (Scored)	☐	☐
4.4	Ensure 'Enable connecting to serial ports' is not enabled for VM Instance (Scored)	☐	☐
4.5	Ensure that IP forwarding is not enabled on Instances (Not Scored)	☐	☐
4.6	Ensure VM disks for critical VMs are encrypted with Customer-Supplied Encryption Keys (CSEK) (Scored)	☐	☐
5	**Storage**		
5.1	Ensure that Cloud Storage bucket is not anonymously or publicly accessible (Scored)	☐	☐
5.2	Ensure that there are no publicly accessible objects in storage buckets (Not Scored)	☐	☐
5.3	Ensure that logging is enabled for Cloud storage buckets (Scored)	☐	☐

6	**Cloud SQL Database Services**		
6.1	Ensure that Cloud SQL database instance requires all incoming connections to use SSL (Scored)	☐	☐
6.2	Ensure that Cloud SQL database Instances are not open to the world (Scored)	☐	☐
6.3	Ensure that MySql database instance does not allow anyone to connect with administrative privileges. (Scored)	☐	☐
6.4	Ensure that MySQL Database Instance does not allows root login from any Host (Scored)	☐	☐
7	**Kubernetes Engine**		
7.1	Ensure Stackdriver Logging is set to Enabled on Kubernetes Engine Clusters (Scored)	☐	☐
7.2	Ensure Stackdriver Monitoring is set to Enabled on Kubernetes Engine Clusters (Scored)	☐	☐
7.3	Ensure Legacy Authorization is set to Disabled on Kubernetes Engine Clusters (Scored)	☐	☐
7.4	Ensure Master authorized networks is set to Enabled on Kubernetes Engine Clusters (Not Scored)	☐	☐
7.5	Ensure Kubernetes Clusters are configured with Labels (Not Scored)	☐	☐
7.6	Ensure Kubernetes web UI / Dashboard is disabled (Scored)	☐	☐
7.7	Ensure `Automatic node repair` is enabled for Kubernetes Clusters (Scored)	☐	☐
7.8	Ensure Automatic node upgrades is enabled on Kubernetes Engine Clusters nodes (Scored)	☐	☐
7.9	Ensure Container-Optimized OS (cos) is used for Kubernetes Engine Clusters Node image (Not Scored)	☐	☐
7.10	Ensure Basic Authentication is disabled on Kubernetes Engine Clusters (Scored)	☐	☐
7.11	Ensure Network policy is enabled on Kubernetes Engine Clusters (Scored)	☐	☐
7.12	Ensure Kubernetes Cluster is created with Client Certificate enabled (Scored)	☐	☐
7.13	Ensure Kubernetes Cluster is created with Alias IP ranges enabled (Scored)	☐	☐
7.14	Ensure PodSecurityPolicy controller is enabled on the Kubernetes Engine Clusters (Scored)	☐	☐
7.15	Ensure Kubernetes Cluster is created with Private cluster enabled (Scored)	☐	☐
7.16	Ensure Private Google Access is set on Kubernetes Engine Cluster Subnets (Scored)	☐	☐
7.17	Ensure default Service account is not used for Project access in Kubernetes Clusters (Scored)	☐	☐
7.18	Ensure Kubernetes Clusters created with limited service	☐	☐

	account Access scopes for Project access (Scored)		

Appendix: Change History

Date	Version	Changes for this version
Aug 21, 2018	1.0.0	Document Created