



Setting up a GxP-Aligned Life Sciences Platform using Data Protection Toolkit

Solution Guide

Contents

Disclaimer	4
1. Overview	5
1.1 Solution Guide	5
1.2 Data Protection Toolkit	5
2. GxP and Life Sciences	6
3. Deploying a GxP-Aligned Life Sciences R&D Platform using DPT	9
3.1 GxP-aligned Life Sciences R&D Platform - Reference Architecture	9
3.2 In-scope Product Guidance	11
Product Guidance - Reading Instructions	11
3.2.1 Google Cloud Storage	12
3.2.2 Google BigQuery	15
3.2.3 Google Cloud Healthcare API	18
3.2.4 Google Cloud Pub/Sub	20
3.2.5 Google Kubernetes Engine	23
3.2.6 Google Cloud Functions	29
3.3 Environment Setup	31
3.4 DPT Access Control	31
3.4.1 Pre-deployment Access Control	32
3.5 Deployment Modes	33
3.5.1 Dry Run Mode	33
3.5.2 Project Creation Mode	33
3.5.3 Project Update Mode	34
3.6 Project Types and Deployment Phases	34
3.6.1 Project Types	34
3.6.2 Deployment Phases	35
3.7 Pre-Deployment setup	36
3.7.1 Initial Setup	36
3.7.2 Forseti Deployment	36
3.7.3 Data-Hosting Project Deployments	40
3.8 Post-Deployment Verification	63
3.8.1 Forseti Project	63
3.8.2 Data-hosting Project (“rnd-platform-project”)	71
4. Extended Product Guidance	91
4.1 Google Cloud Spanner	91
4.2 Google Cloud Bigtable	93
4.3 Google Cloud Firewall	95

4.4 Google Cloud SQL	97
4.5 Deep Learning Virtual Machines	102
4.6 Google Cloud Dataproc	106
4.7 Google Cloud Datalab	110
A. Appendix	113
A.1 GxP, CFR 21 Part 11, and Google Cloud	113
A.2 Compliance	114
A.3 Google Cloud Shared Responsibility Model	115
A.4 Customer Responsibilities	116

Disclaimer

- Google does not intend for this solution guide and accompanying templates, to constitute advice on implementing the various “good practice” guidelines and regulations that apply to medical products (GxP requirements).
- References to GxP requirements in this document are for informational purposes only, for a typical life sciences platform, and this guide does not purport to provide an accurate or comprehensive list of applicable GxP requirements.
 - As such, implementation of the solution guide and documentation of such cases does not constitute full GxP compliance. The customer is responsible for determining which regulatory obligations apply to them in each jurisdiction and take the necessary compliance steps to meet their individual needs.
- The customer is responsible for managing data and applications, including configuration and maintenance of services hosted utilizing Google Cloud. These responsibilities are further enumerated in section [A.4 Customer Responsibilities](#).
- The scope of this solution guide is limited to providing security guidance for protecting and monitoring data within the in-scope resources defined as part of the Reference Architecture for a GxP-Aligned Life Sciences Research and Development (R&D) Platform in Section 3.
- Implementation of the solution guide or reference architecture does not automatically cover any data assets that are stored or processed by other Google Cloud Storage services. Similar protective measures must be applied to all other data stored across the environment.
- The information mentioned in the solution guide is illustrative and not necessarily exhaustive. The information must be read alongside the official documentation.
- This solution guide can be used as an accelerator or framework and will need to be customized to include additional specific requirements and use cases to deploy GxP-aligned workloads.

1. Overview

1.1 Solution Guide

This solution guide covers the process and the guidelines to deploy a typical GxP ("good practice") aligned workload for a typical life sciences platform built on the Cloud (hereinafter, the Life Sciences R&D platform) with recommended security configurations related to role-based access control, data protection and retention, audit logging, and monitoring, amongst others required by the regulation(s). Please refer to Section 2 of this guide for details on GxP.

This guide explains security configurations related to Google Cloud products and services which can be used to provision a minimum viable product (MVP) environment, which can be customized and expanded upon as a part of other workloads, use-cases, or regulatory requirements. Furthermore, this guide also covers [Post-Deployment Verification](#) using the Google Cloud console to verify resources deployed and their corresponding security parameters.

The Life Sciences R&D Platform example used throughout this guide covers an example of how Google Cloud resources can be used in conjunction with an organization's resources to build and deploy a GxP-Aligned Life Sciences R&D Platform. While this solution guide will help accelerate the deployment of Google Cloud products in alignment with GxP, it is intended that users customize the guidance to include specific additional requirements and other use cases applicable to requirements relevant to your organization.

1.2 Data Protection Toolkit

The Data Protection Toolkit (further referred to as DPT) is an open-sourced suite of tools released by Google, for provisioning and managing Google Cloud projects. DPT combines infrastructure-as-code best practices, security configurations, and best practices for provisioning Google Cloud products into a comprehensive end-to-end framework. DPT's easy-to-use and declarative "deployment templates" (written in YAML or JSON) makes it intuitive to understand and easy to validate the deployment workflow even before its implementation. These templates help accelerate the provisioning and configuration of projects, resources, network infrastructure, access management, and monitoring, by leveraging tools like [Terraform](#) and [gcloud](#). DPT can also be used to set up Forseti, an open source tool for continuous configuration monitoring of the deployed projects and their resources. To learn more about Forseti, visit its [website](#). Due to its expressive deployment capabilities and integrated monitoring tools, DPT is a powerful tool for privacy, security, and compliance focused use cases.

DPT templates help in:

- deploying identical environments (e.g. development, test, and production) with minimal manual intervention.

- minimizing build and deployment errors in comparison to manual builds.
- zero-downtime deployment, testing, and validation of Google Cloud workloads.
- Disaster recovery by enabling rapid deployment of failed workloads.
- Deploying infrastructure-related auditing and monitoring tools in parallel with workload deployment.
- Reducing maintenance costs by automating removal of unused resources in conjunction with capacity monitoring.

DPT templates can update or restore the deployments to the required state driving development efficiency. Also, changes to the DPT template can be tracked by maintaining it in a code repository. This drives accountability and maintains discipline and quality control.

Google has published DPT as an open-source repository, which can be cloned and used to deploy the templates. To learn more about DPT, refer to the [DPT Repository](#) on GitHub. DPT, while currently targeted to the healthcare industry, can also be used to support use-cases related to banking and finance, gaming, marketing and education.

2. GxP and Life Sciences

GxP is an abbreviation that refers to “good practice,” regulations and guidelines that apply to medical products. In GxP, the “x” variable covers a wide range of processes used in the development and distribution of regulated products. GxP criteria can be found in government regulations (ex. Federal, Food, Drug & Cosmetic Act) and industry-specific best practice frameworks. Some of the most prevalent regulation sets in GxP include, but are not limited to:

- Good Manufacturing Practice (GMP): 21 CFR Parts [210](#) and [211](#), applicable to drug products.
- Quality System Regulation (QSR): [21 CFR Part 820](#), applicable to medical devices
- Good Laboratory Practice (GLP): 21 CFR Part 58, applicable to nonclinical laboratory studies
- [Good Clinical Practice](#) (GCP): Includes multiple regulations and guidance applicable to scientific studies
- [Good Distribution Practice](#) (GDP): Encompasses various provisions and guidelines addressed in 21 CFR Parts 211 and 820, including those related to handling, storage, and installation

For further details on using GCP in GxP Systems, please review this [Google Cloud Whitepaper](#).

GxP standards and regulations are implemented by different regulators in different countries and regions. Similarly, TGA in Australia and HS-SC in Canada. For the life sciences industry in the United States, GxP requirements can generally be found in the Code of Federal Regulations

(Title 21 CFR) and are intended to minimize risk to patient safety, product quality, and data integrity. E.g. Therapeutic Goods Administration in Australia and Health Canada (HC) in Canada.

For example, Title 21 CFR Part 11 establishes regulations on electronic records and electronic signatures to align with GxP protocols. Under Title 21 CFR Part 11, each computer system involved in the development, production, storage and distribution of pharmaceutical products or medical devices needs to meet the following requirements:

- Fit for purpose – i.e. doing the job it should
- Compliant with relevant regulatory requirements
- Protect regulatory critical data
- Operating in a known, reliable and predictable fashion
- Operated and maintained in a controlled environment

SDLC and related IT processes like change management, incident management, and privacy/security requirements are outside the scope of electronic records and signatures. However, they play a crucial role in building the processes and procedures for meeting requirements pertaining to CFR 21 part 11.

For more information on Title 21 CFR Part 11, please refer to this [page](#). Google Cloud's multi-layered security and privacy controls along with operational practices and compliance safeguards, enforce system validation at different levels. Life Sciences organizations can take advantage of Google Cloud's efficiencies to achieve their compliance goals. For further details on GxP, please refer to [Using Google Cloud in GxP Systems](#).

DPT helps build a baseline GCP environment with a multi-project architecture, identity and access management, data security, network design, and centralized logging that are integral parts of GxP solutions. DPT also enables controls related to configuration auditing, encryption, backup and recovery, access controls, and infrastructure security through easy-to-use, template-based security configurations. Further, DPT helps in building consistent Google Cloud services which can handle sensitive data in alignment with GxP as described in Sections 3, 4, and the Appendix.

However, this guidance serves as a starting point towards GxP compliance and will need to be customized further to include additional and specific use cases or requirements to comprehensively meet the requirements. Google Cloud's customers are responsible for defining the various GxP security and compliance requirements. Google Cloud products regularly undergo independent verification of their security, privacy, and compliance controls, achieving certifications, attestations of compliance, or audit reports against standards around the world.

For additional details and to learn more about GxP on Google Cloud, please refer to A.1 [GxP, CFR 21 Part 11, and Google Cloud](#).

3. Deploying a GxP-Aligned Life Sciences R&D Platform using DPT

This section elaborates on the GxP-Aligned Life Sciences R&D platform example and describes the steps for deploying it using DPT.

To support GxP requirements, a Life Sciences R&D platform should consider compliance through the implementation of technical security controls in addition to considering specific requirements outlined in 21 CFR Part 11 (Electronic Records Requirements). The technical controls as it pertains to this guide including the following:

- Data Retention
- Identity and Access Management
- Data Security and Audit Trail
- Infrastructure Security

Additionally, in order to satisfy GxP requirements, a Life Sciences R&D platform should consider maintaining copies of records and reports for data retention purposes and establish processes for ensuring quality requirements are met.

The sections below describe how to implement the above mentioned technical controls for a specific use-case using DPT. It includes the identified in-scope Google Cloud products and services described in the reference architecture below.

3.1 GxP-aligned Life Sciences R&D Platform - Reference Architecture

GxP requirements should be considered when planning and designing the security architecture for Life Sciences R&D platforms, particularly for the collection and processing of health information, so that appropriate technical security controls are established. DPT leverages the following reference security architecture for setting up and securing a Google Cloud Life Sciences R&D Platform for aligning with certain GxP requirements. Please note that,

- The platform storing life sciences data carries out analysis and research tasks using pre-installed tools and packages within Google Kubernetes Engine containers. Data Loss Prevention (DLP) API is used to remove any sensitive personal data during processing.
- This reference architecture is designed to provide a quick start environment and should be customized by the regulated organizations further for deploying specific R&D use cases or adding Google Cloud products and services based on their compliance requirements.

Though most of the components in the architecture below can be implemented using DPT templates, some of the capabilities, such as Cloud IAM and multi-factor authentication will require additional custom configuration apart from DPT for integration with existing internal and external systems.

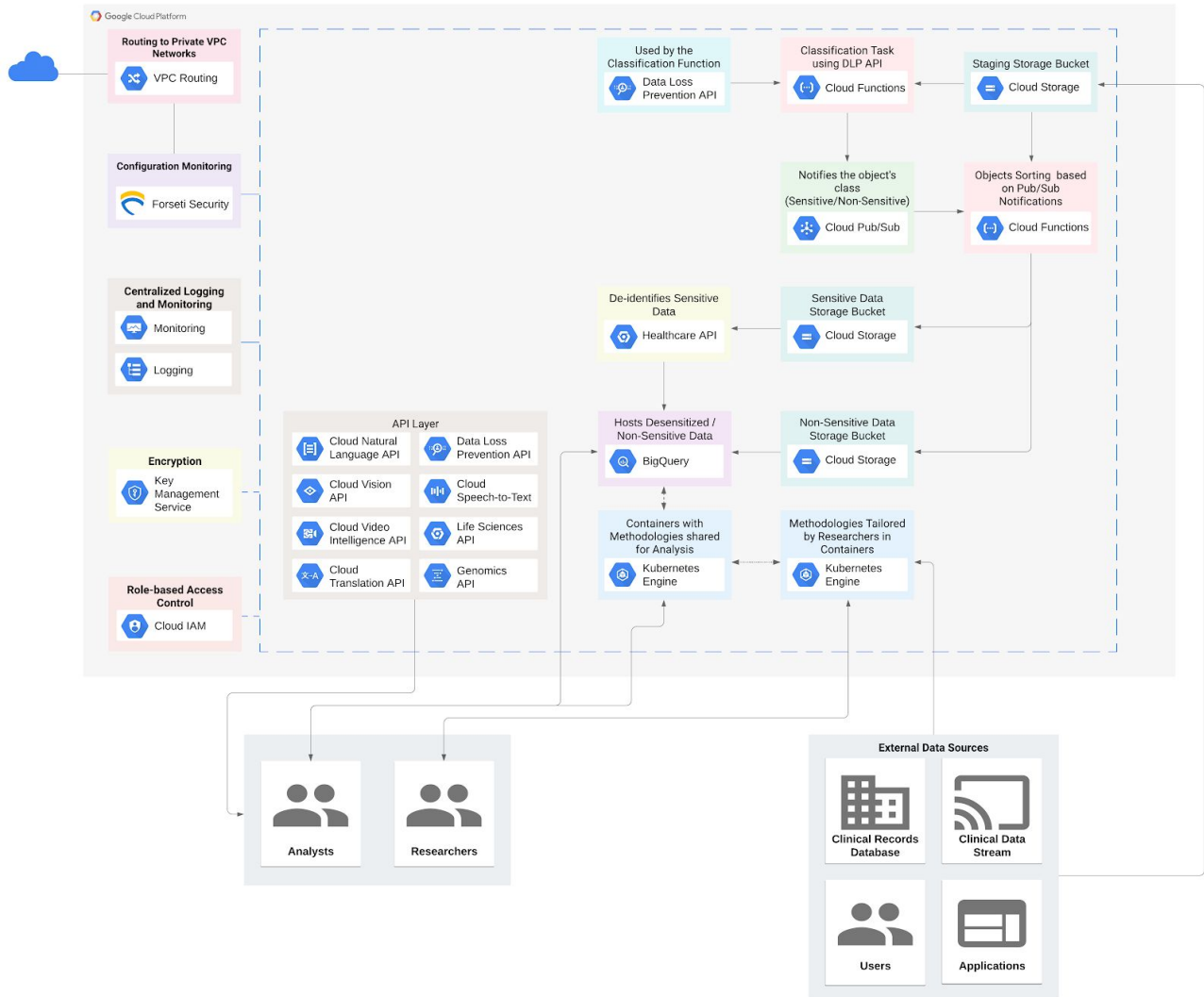


Figure 1: Example - GxP-aligned Life Sciences R&D Platform Architecture

Note: The GxP-aligned Life Sciences R&D platform template discussed in the following sections will deploy the following resources: Cloud Storage buckets, BigQuery datasets, Healthcare datasets (that interact with Healthcare API), Cloud Pub/Sub, Kubernetes Engine Clusters, and Cloud Functions. It also enables the following APIs: Cloud Natural Language, Cloud Vision, Cloud Video Intelligence, Cloud Translation, DLP API, Healthcare API, Life Sciences, and Cloud Speech-to-text. Additional services covered in the architecture can be integrated into the final solution as needed to build an entire workflow around the Life Sciences R&D Platform. DPT also

supports and can be leveraged to deploy different infrastructure environments such as Development, Test, Production etc., with different configurations. To learn more about setting-up environments using DPT, refer to the sample templates from this [link](#).

3.2 In-scope Product Guidance

The following product guidance discusses each service and its role in the R&D Platform architecture. Further, it demonstrates the security configurations for each product through DPT.

The Google Cloud resources deployed by the GxP-Aligned Life Sciences R&D Platform DPT template are:

- *Google Cloud Storage*
- *Google BigQuery*
- *Google Cloud Healthcare API (Healthcare API Datasets)*
- *Google Cloud Pub/Sub*
- *Google Kubernetes Engine Cluster*
- *Google Cloud Functions*

Besides deploying the resources, the template also enables the following Google Cloud APIs:

- *Cloud Natural Language API*
- *Cloud Vision API*
- *Cloud Video Intelligence API*
- *Cloud Translation API*
- *DLP API*
- *Cloud Speech-to-Text API*
- *Life Sciences API*

The Google Cloud API services can be used to perform operations on data based on specific requirements. For example, DLP API can be used to scan, discover, classify, and report on data.

Product Guidance - Reading Instructions

The product guidance has been divided into three sections for easy reading:

1. Product description
2. GxP Guidance for each product:
 - a. The GxP guidance for each product includes a table describing Default Configurations and User-Controlled Configuration options.
 - b. Default Configurations are applied by the GCP service and are meant to represent a reasonable default for most applications. These are enabled when a service is provisioned with default configurations through the Cloud Console, or through the API or CLI interfaces.
 - c. User Controlled Configurations can be applied using DPT, through the Cloud Console, or through the API or CLI interfaces.

3. DPT configurations for each product
 - a. The DPT configurations for each product contains a table with modular blocks of code referenced to the GxP configuration guidance for each product.

3.2.1 Google Cloud Storage

Google Cloud Storage provides worldwide, highly durable object storage that can scale up to exabytes of data. There are four storage classes - Multi-regional, Regional, Nearline, and Coldline. The appropriate storage class can be chosen based on the business purpose and other requirements (e.g., for availability).

As part of the solution architecture, Cloud Storage buckets store the raw data ingested from the healthcare systems before they are normalized and imported into BigQuery.

Cloud Storage offers features like attribute-level access control using Cloud IAM Conditions, admin activity and event logging, encryption, object lifecycle management and versioning, etc.

To learn more about Google Cloud Storage and the parameters discussed below, refer to the [Cloud Storage documentation](#) and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Cloud Storage

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Identity and Access Management	Users in the owners' group of the project are allowed to perform any action allowed by the project's organization policies.	Use G Suite to create users and groups. Additional role-member bindings can be added as mentioned in the template to control access to the storage bucket.
Data Security	Data is encrypted at-rest and in-transit, with Google managing the security keys.	Custom encryption keys can be used instead of Google Cloud managed keys. Refer Customer-managed Encryption Keys for more information.
	Labels are not provided by default. Labels are simple key:value pairs to identify resources as per data classification or sensitivity.	Label parameter values need to be specified per requirements in the template.

Records Retention	Default Configurations	User-Controlled Configurations (ex. via DPT)
--------------------------	-------------------------------	---

Data Security	Accidentally deleted or overwritten objects cannot be retrieved as versioning is disabled by default.	Versioning is always enabled by default when deployed using DPT. Note: DPT has versioning as a mandatory parameter for deployment.
----------------------	---	---

DPT Template Configuration for Google Cloud Storage

Note: For options for the customizable parameters in the template below, please refer to [Cloud Storage Guidance](#) for Terraform. The configurable values in the below template are indicative only. Please modify it to match specific requirements in the context of usage.

Template (Please refer to accompanying *.yaml template for detailed configuration)

```

storage_buckets:
  # Staging Bucket that stores the incoming health data containing PHI and
  # other sensitive data
  - name: {{.STAGING_STORAGE_BUCKET_NAME}}
    # Code Block 3.2.1.a
    # IAM Role Binding
    _iam_members:
      - role: roles/storage.objectCreator
        member: {{.STAGING_STORAGE_BUCKET_OBJECTCREATOR}}
      - role: roles/storage.objectViewer
        member: {{.STAGING_STORAGE_BUCKET_OBJECTVIEWER}}
    # The Cloud KMS key configuration. If not specified, Google's default
    # encryption at rest is used.
    # Code Block 3.2.1.b
    # encryption:
    #   default_kms_key_name: (ex.{google_kms_crypto_key.gcs.self_link})
    location: {{.LOCATION}}
    # Code Block 3.2.1.c
    # force_destroy: true
    versioning:
      enabled: true
    lifecycle_rule:
      - condition:
          age: #(e.g. 90)
        action:
          type: SetStorageClass
          storage_class: #(e.g.
STANDARD/REGIONAL/MULTI_REGIONAL/COLDLINE/NEARLINE)
    # By default, Storage class is set to STANDARD
    # Code Block 3.2.1.d
    labels:
  
```

<pre>data_criticality: #(e.g. low, medium, high) data_type: #(e.g. phi, pii, gcslogs, auditlogs, statefiles, and general)</pre>		
Identity and Access Management	User access control	<p>Refer to Code Block 3.2.1.a</p> <p>_iam_members - Configuration for assigning roles to members and granting appropriate level permissions to the services</p> <p>role: Role to be assigned to the user</p> <p>member: G Suite Users or Groups to which the above role is assigned</p>
Data Security	Encryption & Key Management	<p>Refer to Code Block 3.2.1.b</p> <p>default_kms_key_name - A custom Google Cloud KMS key that is used to encrypt objects added to the bucket. If a custom KMS key is not specified, Google Cloud default encryption will be used for the data at rest. See this whitepaper for more information on how Google encrypts data at rest by default and to understand the key management options.</p>
	Information Lifecycle Management	<p>Refer to Code Block 3.2.1.c</p> <p>versioning - Stores older versions of objects after modification so that in case there are accidental changes, they can be rolled back easily. Versioning is enabled here. DPT enables versioning for storage buckets through the template configuration.</p> <p>lifecycle_rule - An array of objects where each object is a rule consisting of an action and a set of conditions. If multiple conditions are specified in a rule, an object has to match all of the conditions for the action to be taken. If multiple rules are specified with the same action,</p>

		the action is taken when an object matches the condition(s) in any of the rules. Each rule should contain only one action.
	Labels	Refer to Code Block 3.2.1.d labels - They are used to identify assets based on their classification. This can be used to identify the appropriate types of data stored within the service.

3.2.2 Google BigQuery

Google BigQuery is a serverless, highly scalable, and cost-effective data warehouse that can help in analyzing big data at high speed with a zero operational overhead. BigQuery offers features like enhanced query filtering to manage and debug workloads. Additionally, the retention periods for a dataset and its tables can be set based on the data being stored.

As part of the solution architecture, BigQuery ingests data from two sources, de-identified or anonymized healthcare data from Healthcare API Datasets and non-sensitive data from Cloud Storage bucket. Having BigQuery as a repository enables the Data Visualization tools such as Data Studio and Data Transformation tools such as Dataprep and Dataflow to directly import the dataset and save the results into BigQuery. Analysts can also use Kubernetes containers which import data from BigQuery for analysis using pre-installed libraries and tools.

To learn more about BigQuery and the parameters discussed below, refer to the [BigQuery documentation](#) and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google BigQuery

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Identity and Access Management	Users in the owners' group of the project are allowed to perform any action allowed by the project's organization policies.	Use G Suite to create users and groups. Additional role-member bindings can be added as mentioned in the template using the access block to control access to the storage bucket.
Data Security	Data is encrypted at-rest and in-transit, with Google managing the security keys.	Custom encryption keys can be used instead of Google Cloud managed keys. For more information, refer to the Customer-managed Encryption

		Keys (CMEK) documentation . See this whitepaper for more information on how Google encrypts data at rest by default and to understand the key management options.
	Labels are not provided by default.	Label parameter values can be specified to identify assets as per the required values in the template.

DPT Template Configurations for Google BigQuery

Note: For options for the customizable parameters in the template below, please refer to [BigQuery guidance](#) for Terraform. The configurable values in the below template are indicative only. Please modify it to match specific requirements in the context of usage.

<p>Template (Please refer to accompanying *.yaml template for detailed configuration)</p> <pre> bigquery_datasets: - dataset_id: {{.NON_SENSITIVE_DATA_BIGQUERY_DATASET_ID}} # delete_contents_on_destroy: true depends_on: - google_service_account.{{.NON_SENSITIVE_DATA_BQ_SERVICE_ACCOUNT_NAME}} # Code Block 3.2.2.a access: - user_by_email: \${google_service_account.{{.NON_SENSITIVE_DATA_BQ_SERVICE_ACCOUNT_NAME}}.email} role: roles/bigquery.dataEditor # Can provide roles as per requirement. - special_group: {{.NON_SENSITIVE_DATA_BQ_SPECIAL_GROUP}} role: {{.NON_SENSITIVE_DATA_BQ_SPECIAL_GROUP_ROLE}} # Code Block 3.2.2.b # default_encryption_configuration: # Code Block 3.2.2.c # kms_key_name: (ex. {google_kms_crypto_key.gcs.self_link}) location: {{.LOCATION}} # Code Block 3.2.2.d labels: data_criticality: {{.NON_SENSITIVE_DATA_BIGQUERY_DATASET_DATA_CRITICALITY_LABEL}} datatype: {{.NON_SENSITIVE_DATA_BIGQUERY_DATASET_DATA_TYPE_LABEL}} project: {{.RND_PROJECT_ID}} </pre>		
Identity and Access	User access control	Refer to Code Block 3.2.2.a

Management		<p>access - An array of objects that define dataset access for one or more entities. Each object has a role and a user entity to which the role must be assigned. These user entities can be domain, group_by_email, user_by_email or special groups.</p>
Data Security	Encryption & Key Management	<p>Refer to Code Block 3.2.2.b</p> <p>default_encryption_configuration - The default encryption key for all tables in the dataset. Once this property is set, all newly created partitioned tables in the dataset will have an encryption key set to this value, unless table creation request (or query) overrides the key. It has the following parameter:</p> <p>Refer to Code Block 3.2.2.c</p> <p>kms_key_name - Google Cloud KMS encryption key that will be used to protect the destination BigQuery table.</p> <p>If a custom KMS key is not specified, Google Cloud default encryption will be used for the data at rest. See this whitepaper for more information on how Google encrypts data at rest by default and to understand the key management options.</p>
	Labels	<p>Refer to Code Block 3.2.2.d</p> <p>labels - Labels are used to identify assets based on their classification. This can be used to identify the appropriate types of data stored within the service.</p>

3.2.3 Google Cloud Healthcare API

Google Cloud Healthcare API bridges the gap between healthcare systems and applications on Google Cloud. The purpose of the Cloud Healthcare API is to ingest resources in healthcare-specific data formats, such as HL7v2, DICOM, and FHIR. The Cloud Healthcare API also provides de-identification capabilities which can be used to redact or obfuscate personal information in healthcare data before aggregation in BigQuery, enabling an extra layer of data privacy.

To learn more about Cloud Healthcare API and the parameters discussed below, refer to the [Cloud Healthcare API documentation](#) and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Cloud Healthcare API

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Identity and Access Management	Users in the owners' group of the project are allowed to perform any action allowed by the project's organization policies.	The template here provides guidance for adding additional role member bindings to the healthcare dataset and its datastores.
Data Security	The methods are not applied by default and need to be chosen based on the type of data.	Healthcare API's methods for de-identification must be used when appropriate based on the data residing in the Healthcare API datasets. See the Healthcare API Data De-Identification Guide for more information.

DPT Template Configuration for Google Cloud Healthcare API

Note: For options for the customizable parameters in the template below, please refer to Google Cloud documentation for [Cloud Healthcare API](#). The configurable values in the below template are indicative only. Please modify it to match specific requirements in the context of usage.

```

Template (Please refer to accompanying *.yaml template for detailed configuration)
healthcare_datasets:
  # appropriate name must be chosen based on its purpose
  - name: {{.HEALTHCARE_DATASET_NAME}}
    location: {{.REGION}}
  # IAM Role Binding
  # role-member bindings can be added/removed as required
  # Code Block 3.2.3.a
  _iam_members:
    - role: roles/healthcare.datasetViewer
      member: user:user@domain

```

```

- role: roles/healthcare.datasetViewer
  member: user:user@domain
_dicom_stores:
  # appropriate name must be chosen based on its purpose
- name: dicom-store
  # role-member bindings can be added/removed as required
  # Code Block 3.2.3.a
  _iam_members:
    - role: roles/healthcare.dicomEditor
      member: user:user@domain
    - role: roles/healthcare.dicomStoreAdmin
      member: user:user@domain
_fhir_stores:
  # appropriate name must be chosen based on its purpose
- name: fhir-store
  # role-member bindings can be added/removed as required
  # Code Block 3.2.3.a
  _iam_members:
    - role: roles/healthcare.fhirResourceReader
      member: user:user@domain
    - role: roles/healthcare.fhirResourceEditor
      member: user:user@domain
_hl7_v2_stores:
  # appropriate name must be chosen based on its purpose
- name: hl7-v2-store
  # role-member bindings can be added/removed as required
  # Code Block 3.2.3.a
  _iam_members:
    - role: roles/healthcare.hl7V2StoreAdmin
      member: user:user@domain
    - role: roles/healthcare.hl7V2Ingest
      member: user:user@domain
    - role: roles/healthcare.hl7V2Editor
      member: user:user@domain

```

Identity and Access Management	User Access Control	Refer to Code Blocks 3.2.3.a _iam_members - Configuration for assigning roles to members and granting appropriate level permissions to the services role: Role to be assigned to the user member: G Suite Users or Groups to which the above role is assigned
Data Security	De-identification/Anonymization	Applications can call specific methods within the Healthcare API to

		de-identify data based on specific datastores. This is a capability which can be access by applications dealing with sensitive ePHI by methods such as dicomStores.deidentify
--	--	---

3.2.4 Google Cloud Pub/Sub

Pub/Sub is a fully managed messaging service for exchanging information (e.g. events, notifications, etc.) between applications. This separation of information flows decouples application services and helps in creation of customized application responses.

In the solution architecture, Pub/Sub notifies the Cloud Function about the data objects in the staging storage bucket and their corresponding labels(sensitive vs non-sensitive) for segregating data.

To learn more about Cloud Pub/Sub and the parameters discussed below, refer to the [Pub/Sub documentation](#) and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Cloud Pub/Sub

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Identity and Access Management	Users in the owners' group of the project are allowed to perform any action allowed by the project's organization policies.	Use G Suite to create users and groups. Additional role-member bindings can be added as mentioned in the template to control access to the Pub/Sub topics and subscription.

Records Retention	Default Configurations	User-Controlled Configurations (ex. via DPT)
Data Security	Cloud Pub/Sub requires regions to be defined by default. Users can opt for different regions as per requirements or client environment constraints. Users can add values to these parameters as per the requirements or constraints.	The template allows messages published to a topic to be stored in particular regions. Some regions are already defined in the template It has default values for the message retention parameters used. It handles invalid messages by publishing them to a separate topic called "Dead-Letter"

DPT Template Configuration for Google Cloud Pub/Sub

Note: For options for the customizable parameters in the template below, please refer to [Pub/Sub guidance](#) for Terraform. The configurable values in the below template are indicative only. Please modify it to match specific requirements in the context of usage.

Template (Please refer to accompanying *.yaml template for detailed configuration)

```
pubsub_topics:
  - name: {{.PUB_SUB_TOPIC_NAME}}
    # Policy defining the set of guidelines related to storage of messages
    # published to the topic.
    message_storage_policy:
      # List of regions where the messages are allowed to be stored. Can be
      # added or subtracted or modified as per requirements.
      allowed_persistence_regions:
        - europe-west1
        - europe-west3
        - europe-west4
        - europe-north1
      # IAM Role Binding
      # Code Block 3.2.4.a
      _iam_members:
        - role: roles/pubsub.editor
          member: {{.PUB_SUB_TOPIC_EDITOR_ROLE_USER}}
        - role: roles/pubsub.viewer
          member: {{.PUB_SUB_TOPIC_VIEWER_ROLE_USER}}
      labels:
        data_criticality: {{.PUB_SUB_TOPIC_CRITICALITY_LABEL}}
        datatype: {{.PUB_SUB_TOPIC_DATA_TYPE_LABEL}}
        project: {{.RND_PROJECT_ID}}
      _subscriptions:
        - name: {{.PUB_SUB_SUBSCRIPTION_NAME}}
          # Duration of the messages stored in the regions.
          # Code Block 3.2.4.b
          message_retention_duration:
            {{.PUB_SUB_SUBSCRIPTION_MESSAGE_RETENTION_DURATION}}
          # Whether acked messages are retained or not.
          retain_acked_messages:
            {{.PUB_SUB_SUBSCRIPTION_RETAIN_ACKED_MESSAGES}}
          # This value is the maximum time after a subscriber receives a
          # message before the subscriber should acknowledge the message.
          ack_deadline_seconds:
            {{.PUB_SUB_SUBSCRIPTION_ACK_DEADLINE_SECONDS}}
          # Policy defining the guidelines for expiration of the
          # subscription.
          expiration_policy:
```

```

    # Time-to-Live duration of associated resources.
    ttl: {{.PUB_SUB_SUBSCRIPTION_TIME_TO_LIVE}}
# IAM Role Binding
_iam_members:
  - role: roles/pubsub.subscriber
    member: {{.PUB_SUB_SUBSCRIPTION_SUBSCRIBER_ROLE_USER}}
  - role: roles/pubsub.editor
    member: {{.PUB_SUB_SUBSCRIPTION_EDITOR_ROLE_USER}}
# A policy that specifies the conditions for dead lettering
messages in this subscription. If dead_letter_policy is not set, dead
lettering is disabled.
  dead_letter_policy:
    dead_letter_topic:
      ${{google_pubsub_topic.{{.PUB_SUB_DEAD_LETTER_TOPIC_NAME}}.id}}
    max_delivery_attempts: 10
# If push delivery is used with this subscription, uncomment the
following field used to configure it. Refer to
"https://www.terraform.io/docs/providers/google/r/pubsub_subscription.html"
for more information.
  # push_config:
  #   oidc_token:
  #     service_account_email:
      ${{google_service_account.forsetibqsa.name}}
  #     audience: (ex. https://example.com/push)
  #     push_endpoint: (ex. https://example.com/push)
  #     attributes: (ex. x-goog-version)

```

Identity and Access Management	User access control	Refer to Code Block 3.2.4.a iam_members - Member role for the user. G Suite users/groups and Cloud IAM roles can be used to control access.
Data Security	Information Lifecycle Management	Refer to Code Block 3.2.4.b message_retention_duration - Specifies the duration of retention of unacknowledged subscription messages. retain_acked_messages: - Indicates whether to retain acknowledged messages in the subscription backlog. ack_deadline_seconds: - Specifies the maximum time after a subscriber

		<p>receives a message before the subscriber acknowledges the message.</p> <p>ttl: - Specifies the "time-to-live" duration.</p> <p>dead_letter_policy: - Specifies the conditions for dead letter messages in the subscription.</p>
--	--	--

3.2.5 Google Kubernetes Engine

Google Kubernetes Engine (GKE) is a managed service for running containerized applications using Google infrastructure. The GKE environment consists of Compute Engine Instances grouped together to form clusters. Google Cloud provides advanced cluster management features such as Node Pools for additional flexibility inside clusters, autoscaling Compute Engine Instances, Node auto-repair, Logging, and Monitoring.

In this solution architecture, researchers and analysts use separate Kubernetes clusters. Researchers would access sensitive health data produced by on-prem systems to conduct research activities and create containers with tailored methodologies to share with developers or analysts. These ad-hoc containers with only pre-installed tools and libraries (without sensitive data) will then be deployed by analysts on the cluster dedicated to them for carrying out analysis of desensitized data stored in BigQuery.

To learn more about Kubernetes Engine and the parameters discussed below,, refer to the Kubernetes [Documentation](#) and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Kubernetes Engine Cluster

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Identity and Access Management	Users in the owners' group of the project are allowed to perform any action allowed by the project's organization policies.	Use G Suite to create users and groups. Additional credentials for Kubernetes master need to be added as well.
Infrastructure Security	The template is configured to enable auto-scaling, auto-repair and auto-upgrade features to provide adequate resources to the Kubernetes Engine cluster nodes on demand. Private cluster config block is also	The auto-scaling policy attached to the cluster can be edited for scaling boundaries to provide fine-grained control over cluster resources throughout cluster lifetime. Maintenance policy and master

	implemented for additional network security such as private nodes and private endpoints.	authorized networks config can be used and modified according to the requirements
Data Security	<p>Data is encrypted at-rest and in-transit, with Google managing the security keys.</p> <p>The template enables security parameters such as binary authorization, pod security policy and shielded nodes.</p>	<p>Custom encryption keys can be used for encryption of data stored on Google Compute Engine disks. See Customer-managed Encryption Keys for more information.</p>

DPT Template Configuration for Google Kubernetes Engine Cluster

Note: For options for the customizable parameters in the template below, please refer to [Google Kubernetes Engine](#) guidance for Terraform. The configurable values in the below template are indicative only. Please modify it to match specific requirements in the context of usage.

```

Template (Please refer to accompanying *.yaml template for detailed configuration)
terraform_deployments:
  resources:
    config:
      resource:
        - google_container_cluster:
            # GKE Cluster dedicated for containers used by Researchers
            cluster_for_researchers:
              name: {{.RESEARCHERS_GKE_CLUSTER_NAME}}
              location: {{.REGION}}
              remove_default_node_pool: true
              initial_node_count: 1
              default_max_pods_per_node:
                {{.MAX_PODS_PER_NODE_RESEARCHERS_GKE_CLUSTER}}
              enable_binary_authorization: true
              enable_tpu: true
              enable_shielded_nodes: true
              # Available options include logging.googleapis.com(Legacy
              Stackdriver), logging.googleapis.com/kubernetes(Stackdriver Kubernetes
              Engine Logging), and none.
              logging_service: logging.googleapis.com/kubernetes
              # Uncomment the following field for maintenance configurations.
              Refer to
              "https://www.terraform.io/docs/providers/google/r/container_cluster.html"
              for field values information.
              # maintenance_policy:
  
```

```

#   daily_maintenance_window:
#   -----
#   daily_maintenance_window:
#     start_time: 03:00
#   ----- OR -----
#   recurring_window:
#     start_time: 2019-01-01T09:00:00-04:00
#     end_time: 2019-01-01T17:00:00-04:00
#     recurrence: FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR
#   -----

# Available options include monitoring.googleapis.com(Legacy
Stackdriver), monitoring.googleapis.com/kubernetes(Stackdriver Kubernetes
Engine Monitoring), and none.
  monitoring_service: monitoring.googleapis.com/kubernetes
  # master_authorized_networks_config:
  #   External networks that can access the Kubernetes cluster
master through HTTPS.
  #   cidr_blocks:
  #     cidr_block: (ex. x.x.x.x/xx)
  resource_labels:
    labels:
      data_criticality:
{{.RESEARCHERS_GKE_CLUSTER_DATA_CRITICALITY_LABEL}}
      datatype:
{{.RESEARCHERS_GKE_CLUSTER_DATASET_DATA_TYPE_LABEL}}
      project: {{.RND_PROJECT_ID}}
  pod_security_policy_config:
    enabled: true
  # authenticator_groups_config:
  # The name of the RBAC security group for use with Google
security groups in Kubernetes RBAC. Group name must be in format
gke-security-groups@yourdomain.com.
  #   security_group: (ex. gke-security-groups@yourdomain.com)
  # Network and Subnetwork used by this GKE cluster
# Code Block 3.2.5.c
  network: ${google_compute_network.private_network.self_link}
  subnetwork:
${google_compute_subnetwork.researchers-gke-cluster-subnetwork.name}
  # The authentication information for accessing the Kubernetes
master

# Code Block 3.2.2.a
master_auth:
  username: '{{.RESEARCHERS_GKE_CLUSTER_AUTH_USERNAME}}'
  password: '{{.RESEARCHERS_GKE_CLUSTER_AUTH_PASSWORD}}'
  client_certificate_config:

```



```
    issue_client_certificate: false
  # Configuration for private cluster with private nodes
  private_cluster_config:
    # Enables the private cluster feature, creating a private
endpoint on the cluster
    enable_private_nodes: true
    # The cluster's private endpoint is used as the cluster
endpoint and access through the public endpoint is disabled when true
    enable_private_endpoint: true
    # The IP range in CIDR notation to use for the hosted master
network. The range should not overlap with an existing subnet.
    master_ipv4_cidr_block:
{{.RESEARCHERS_GKE_CLUSTER_MASTER_IPV4_CIDR_BLOCK}} # (ex. 172.16.0.32/28)

    # Configuration of cluster IP allocation for VPC-native
clusters. Adding this block enables IP aliasing,
    # making the cluster VPC-native instead of routes-based.
    ip_allocation_policy:
    # The name of the existing secondary range in the cluster's
subnetwork to use for pod IP addresses.
    cluster_secondary_range_name:
researchers-gke-cluster-pods-range
    # The name of the existing secondary range in the cluster's
subnetwork to use for service cluster IPs.
    services_secondary_range_name:
researchers-gke-cluster-services-range
    # Cluster configuration of Node Auto-Provisioning with Cluster
autoscaler to automatically adjust the size of the cluster
    cluster_autoscaling:
    enabled: true
    # Limits on CPU and Memory for the cluster
    resource_limits:
    - resource_type: memory
      minimum: {{.RESEARCHERS_GKE_CLUSTER_MIN_MEMORY}}
      maximum: {{.RESEARCHERS_GKE_CLUSTER_MAX_MEMORY}}
    - resource_type: cpu
      minimum: {{.RESEARCHERS_GKE_CLUSTER_MIN_CPU}}
      maximum: {{.RESEARCHERS_GKE_CLUSTER_MAX_CPU}}
    # Contains defaults for a node pool created by Node
Auto-Provisioning.
    auto_provisioning_defaults:
    service_account:
${google_service_account.researchers_gke_cluster_sa.email}
    oauth_scopes:
    - https://www.googleapis.com/auth/bigquery.readonly
    autoscaling_profile: BALANCED # (or OPTIMIZE_UTILIZATION)
# Code Block 3.2.5.b
```

```

        database_encryption:
          state: (ex. ENCRYPTED or DECRYPTED)
          key_name: (ex.
projects/my-project/locations/global/keyRings/my-ring/cryptoKeys/my-key or
{google_kms_crypto_key.gcs.self_link}) # Refer
"https://cloud.google.com/kubernetes-engine/docs/reference/rest/v1beta1/proj
ects.locations.clusters#Cluster.DatabaseEncryption"

- google_container_node_pool:
  researchers_cluster_preemptible_nodes:
    name: researchers-gke-cluster-node-pool
    location: {{.REGION}}
    cluster:
    ${google_container_cluster.cluster_for_researchers.name}
    node_count: 1
    # Node management configuration, wherein auto-repair and
auto-upgrade is configured.
    # Code Block 3.2.5.d
    management:
      auto_repair: enable
      auto_upgrade: enable
    # Configuration required by Cluster autoscaler to adjust the
size of the node pool to the current cluster usage.
    autoscaling:
      min_node_count: 1
      max_node_count: {{.MAX_RESEARCHERS_GKE_CLUSTER_NODE_COUNT}}
    # Node configuration of the pool.
    node_config:
      preemptible: true
      machine_type: n1-standard-1
      metadata:
        disable-legacy-endpoints: 'true'
      oauth_scopes:
        - https://www.googleapis.com/auth/logging.write
        - https://www.googleapis.com/auth/monitoring

```

Identity and Access Management	User access control	Refer to Code Block 3.2.5.a master_auth: - Authentication information (username and password) for accessing Kubernetes master.
Data Security	Encryption & Key Management	Refer to Code Block 3.2.5.b database_encryption: - The name of the KMS key used under this block to encrypt various sensitive files. It is a

		<p>key stored on Google Cloud KMS. See this whitepaper for more information on how Google encrypts data at rest by default and to understand the key management options.</p> <p>enable_binary_authorization: - Specifies whether container images are validated by Google Binary Authorisation or not.</p> <p>enable_shielded_nodes: - Specifies whether to use Shielded GKE nodes which provide additional identity and integrity to the nodes.</p> <p>pod_security_policy_config: - Block that specifies whether PodSecurityPolicy is enabled or not.</p>
Infrastructure Security	Network Security	<p>Refer to Code Block 3.2.5.c</p> <p>network: - The VPC network under which the cluster is created. This setting must always be configured.</p> <p>subnetwork: - The subnet under the previously mentioned VPC network under which the cluster is created. This setting must always be configured.</p>
	Asset Management	<p>Refer to Code Block 3.2.5.d</p> <p>autoscaling: - The autoscaling policy config associated with the nodes.</p> <p>auto_repair: - Specifies whether the nodes will be automatically repaired</p> <p>auto_upgrade: - Specifies whether the nodes will be automatically upgraded.</p> <p>maintenance_policy: - Maintenance policy specifying the daily maintenance window and recurring window</p>

3.2.6 Google Cloud Functions

Google Cloud Functions is a serverless solution to develop standalone compute functions which can be run as required. JavaScript, Python 3, or Go runtimes are some of the supported runtimes on Cloud Functions.

In this solution architecture, Cloud Functions are utilized in two ways.

- Trigger function when data is imported to the staging Cloud Storage bucket. The DLP API can then be called to segregate the data based on labels.
- Function triggered by Cloud Pub/Sub to store segregated data in separate storage buckets.

To learn more about Cloud Functions and parameters discussed below, refer to the [Cloud Functions Documentation](#) and [module configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Cloud Functions

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Infrastructure Security	The template assigns 60 seconds for function timeout and allows the trigger to retry incase of failure.	The timeout value can be modified as per the requirement of the requirements.

DPT Template Configuration for Google Cloud Functions

Note: For options for the customizable parameters in the template below, please refer to [Cloud Functions](#) guidance for Terraform. The configurable values in the below template are indicative only. Please modify it to match specific requirements in the context of usage.

```

Template (Please refer to accompanying *.yaml template for detailed configuration)
terraform_deployments:
  resources:
    config:
      resource:
        - google_cloudfunctions_function:
            # Cloud Function that runs DLP job to generate classification
            labels (Sensitive/Non-sensitive) for objects in Staging Storage Bucket and
            pushes messages to Pub/Sub
            create_DLP_job:
              name: create-DLP-job
              description: This function is triggered by new files uploaded to
              the designated Cloud Storage staging bucket.
              runtime: python37
              available_memory_mb: 128
  
```

```
# Place where the code for the cloud function is stored
source_archive_bucket: {{.CLOUD_FUNCTION_ZIP_BUCKET}}
source_archive_object: {{.CLOUD_FUNCTION_ZIP_OBJECT}}
timeout: 60
region: {{.REGION}} # support only for us-central1 at present
entry_point: create_DLP_job
event_trigger:
  event_type: google.storage.object.finalize
  resource: {{.STAGING_STORAGE_BUCKET_NAME}}
  failure_policy:
    retry: true
environment_variables:
  YOUR_QUARANTINE_BUCKET: {{.STAGING_STORAGE_BUCKET_NAME}}
  YOUR_SENSITIVE_DATA_BUCKET:
{{.YOUR_SENSITIVE_DATA_BUCKET_NAME}}
  YOUR_NON_SENSITIVE_DATA_BUCKET:
{{.YOUR_NON_SENSITIVE_DATA_BUCKET_NAME}}
  PROJECT_ID_HOSTING_STAGING_BUCKET: {{.RND_PROJECT_ID}}
  PUB_SUB_TOPIC: {{.PUB_SUB_TOPIC_NAME}}
# Cloud Function that fetches classification labels
(Sensitive/Non-sensitive) for objects in Staging Storage Bucket from
# Pub/Sub and sort them into Non-sensitive and Sensitive data
storage buckets.
resolve_DLP:
  name: resolve-DLP
  description: This function listens to the Pub/Sub notification
from the create_DLP_job function.
  runtime: python37
  available_memory_mb: 128
# Place where the code for the cloud function is stored
source_archive_bucket: {{.CLOUD_FUNCTION_ZIP_BUCKET}}
source_archive_object: {{.CLOUD_FUNCTION_ZIP_OBJECT}}
# Code Block 3.2.6.a
timeout: 60
region: us-central1 # support only for us-central1 at present
entry_point: resolve_DLP
event_trigger:
  event_type: google.pubsub.topic.publish
  resource: {{.PUB_SUB_TOPIC_NAME}}
  failure_policy:
    retry: true
environment_variables:
  YOUR_QUARANTINE_BUCKET: {{.STAGING_STORAGE_BUCKET_NAME}}
  YOUR_SENSITIVE_DATA_BUCKET:
{{.YOUR_SENSITIVE_DATA_BUCKET_NAME}}
  YOUR_NON_SENSITIVE_DATA_BUCKET:
{{.YOUR_NON_SENSITIVE_DATA_BUCKET_NAME}}
```

<pre>PROJECT_ID_HOSTING_STAGING_BUCKET: {{.RND_PROJECT_ID}} PUB_SUB_TOPIC: {{.PUB_SUB_TOPIC_NAME}}</pre>		
Infrastructure Security	Asset Management	<p>Refer to Code Block 3.2.6.a</p> <p>timeout: - The value for function timeout. Cannot be more than 540 seconds</p> <p>failure_policy: - Specifies whether the function should be retired on failure or not..</p>

3.3 Environment Setup

DPT can be run locally on a computer or by using Google Cloud Shell.

Prior to running DPT locally, the following tools must be installed:

- [Bazel](#) - An open-source build and test tool
- [Terraform](#) - A Cloud provisioning tool
- [Cloud SDK](#) - A set of tools for managing resources and applications hosted on Google Cloud.
- [Git](#) - A distributed version control system.

This step is not required when using Google Cloud Shell, as the required tools are already installed and ready to use.

3.4 DPT Access Control

The access control for DPT is covered under two sections to enhance the security of the deployments. Deploying DPT requires 'owner'(privileged) rule at the organisation or folder level (<https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy>) to deploy resources. Considering privileged access security and separation of access, the access required to deploy resources should be separated from the access required to operate them.

Sections [3.4.1](#) and [3.4.2](#) explain in detail the access grants before, during and through the deployment of resources. This approach ensures that only required accesses are granted based on requirement in each stage of the project creation lifecycle. The access control of resources during the project lifecycle is not covered under this solution guide.

3.4.1 Pre-deployment Access Control

Before a template is deployed, DPT requires creation of two groups for each project in the template.

- **Owner:** {PROJECT_ID}-owners@{DOMAIN}. This group is granted the owner's role for the project, which allows members to do anything permitted by organization policies within

the project. Additions to the owner's group should be for a short term and controlled tightly.

- **Auditor:** {PROJECT_ID}-auditors@{DOMAIN}. The auditor's group is granted the permission to list resources, view IAM configuration, and view contents of audit logs, but not to view any hosted data. If there are multiple data projects, it is advisable to maintain a single auditors group across all projects.

DPT needs 'owner' permissions for the projects' in the template to provision their resources. So initially, to grant provisioning access, the user identity deploying the template is temporarily added to the owners' group for provisioning projects and resources.

Using a service account: It is ideal to use a service account to deploy a DPT workload rather than a user account, as a service account can be granted the minimum set of permissions required to deploy the DPT template (*Billing User, Project Billing Manager and Project Creator*) and its scopes can be restricted for interacting with necessary services only. Similarly, this service account must be added to the project's owners' groups for provisioning.

The *Service Account User* assumes the roles of the service account having owner permissions to deploy the workloads. The time period should be restricted to the period of deployment only.

3.4.2 Post-deployment Access Control

Post deployment of a template, DPT removes the deploying user (or user identity) from the owners' groups of all the projects in the template, effectively restricting further access to the projects. This ensures that only specific pre-approved owners continue to have control post-deployment.

Note: DPT grants roles and permissions to users, groups, and entities (e.g., service accounts). To further customize access after deployment is complete, user groups should be created to control access to the projects and their underlying resources using Google G Suite Admin Console, Cloud Identity, or Google Cloud Directory Sync (GCDS). These user groups can be granted custom roles and permissions using Cloud IAM and conditional access policies. For further information, please refer to the [Cloud IAM documentation](#).

3.5 Deployment Modes

The section describes the deployment of the GxP-aligned Life Sciences R&D Platform DPT configuration templates. Prior to launching any of the following deployment modes, please ensure the following steps are performed:

- Copy the git repository to a folder (locally or on Google Cloud Shell).
- Create a folder or directory within `./healthcare/deploy`. (In this example, the folder has been named `'gcprd'`).

- Copy the DPT GxP-aligned AI/ML templates (variables.yaml and config.yaml) into `./healthcare/deploy/gcprd`.

The DPT template file can be run in three different modes based on what is being done - validating the template, deploying it for the first time, or updating an existing deployment.

3.5.1 Dry Run Mode

Dry Run mode helps to review everything that the template might do from the beginning through the end of the deployment process without really deploying it.

This helps in validating the configuration file and ensures that it is well-formed. However, one caveat is that it may not detect any runtime errors that might happen during the deployment process.

```
bazel run cmd/apply:apply -- \  
  --enable_terraform \  
  --config_path=./healthcare/deploy/gcprd/variables.yaml \  
  --terraform_apply_flags="--parallelism=1" \  
  --dry_run
```

3.5.2 Project Creation Mode

Project Creation mode helps to deploy the project once the configuration file is validated through the Dry Run mode.

```
bazel run cmd/apply:apply -- \  
  --enable_terraform \  
  --config_path=./healthcare/deploy/gcprd/variables.yaml \  
  --terraform_apply_flags="--parallelism=1"
```

3.5.3 Project Update Mode

Project Update mode helps to make modifications to an already existing project like adding a resource or updating a setting. The relevant project ID must be specified in the `--projects` option. If there are multiple projects, their project IDs must be separated by commas against the option.

```
bazel run cmd/apply:apply -- \  
  --enable_terraform \  
  --config_path=./healthcare/deploy/gcprd/variables.yaml \  
  --terraform_apply_flags="--parallelism=1" \  
  --projects=project-id
```



```
--projects=[PROJECTS]
```

3.6 Project Types and Deployment Phases

DPT uses Terraform as the primary deployment tool.

- The deployment process kicks-off when the helper script [apply.go](#) reads the configurations from the configuration file and uses various other scripts from DPT to create Terraform scripts.
- The created Terraform scripts are then deployed in a phased manner.

3.6.1 Project Types

The projects deployed by DPT can be broadly classified into two categories - base projects and data-hosting projects. A DPT template can have multiple data-hosting projects but only one base project per category as discussed below.

- **Data-hosting project:** Any project defined under the *projects* list of the template is a data-hosting project. All the essential workloads and use-cases are developed as data-hosting projects. Eg. Data-warehouse workload, Analytics and AI/ML platforms, etc. Audit logging and resource tracking can either be defined separately for each data-hosting project or defined to be central for all the data-hosting projects as base projects.
- **Audit logs project(optional):** This is a base project to deploy central audit logging for all the data-hosting projects in the template. It is defined under the *audit_logs_project* section of the template. Once configured, it cannot be changed.
- **DevOps project(optional):** This is a base project for centralized state management of resources across all the data hosting projects in the template. It is defined under the *devops* section outside of the data-hosting projects. Once configured, it cannot be changed.
- **Forseti project(optional):** This is a base project to deploy Forseti for configuration monitoring of resources deployed across all the data-hosting projects. It is defined under the *forseti* section of the template.

3.6.2 Deployment Phases

As mentioned above, base projects such as central *DevOps*, *Audit*, and *Forseti* are optional and deployed before the projects which host data (i.e., data-hosting projects). The projects are deployed in phases and dependencies are addressed automatically.

The phases of deploying the template are mentioned as below:

1. **Project Creation:** In this phase, the project is created first, and then a storage bucket, which stores the state of the deployments related to the project.

NOTE: If a top-level devops block is set in the config, all the state buckets will be created in the devops project.

2. **Resources:** This phase contains multiple deployments grouped as follows:
 - a. **Services:** This deployment consists of the default set of services required for the deployment of resources included in the project.
 - b. **Resources:** The deployment consists of specific preset default resources such as IAM permissions, logging metrics, and alert policies as defined resources under the project.

3. **Audit:** This phase contains a single deployment, which creates audit log resources defined in the audit block of a project (BigQuery Dataset and Cloud Storage bucket) as well as logging sinks to export audit logs.

NOTE: If a top-level audit block is set in the config, these resources will be created in the central audit project.

4. **Forseti:** If the Forseti project config is also being applied, a Forseti instance is deployed in the Forseti project at this point and granted the minimum necessary access to each project to monitor security violations across these.

The following table explains the contents of the configuration file (config.yaml) and also details the deployment of data warehouse data-hosting project and its resources in phases.

Note: config.yaml is the base template which is imported into the variables.yaml template during runtime. While the config.yaml defines the deployment process and resources of the workload; parameters and configurations of the workload are declared in the variables.yaml file. Multiple variables.yaml files can be created to re-use config.yaml file deploy same workloads with different configurations.

3.7 Pre-Deployment setup

3.7.1 Initial Setup

<ul style="list-style-type: none"> • The <code>'billing_account'</code> parameter sets up the billing account used by base projects and data-hosting projects deployed by this file. This can be used to restrict billing accounts used for projects restricting costs and controlling who can enable billing. • The domain of the organization under which the projects are deployed must be mentioned against the <code>'domain'</code> parameter. 	<pre>overall: billing_account: {{.BILLING_ACCOUNT}} domain: {{.DOMAIN}} organization_id: {{.ORGANIZATION_ID}} # folder_id: {{.FOLDER_ID}}</pre>
--	---

3.7.2 Forseti Deployment

Forseti is a collection of open-source tools using rule-based policies to monitor and enforce configuration state on Google Cloud projects and resources. The following section of DPT installs Forseti Security and its core Forseti Security modules. The following modules can then be configured to take a snapshot of GCP resources, monitor and enforce configurations.

- **Inventory** : Record a snapshot of GCP resources to Cloud SQL to maintain a historical record of resources in Google Cloud.
- **Scanner**: Use the information collected by Forseti Inventory to regularly compare role-based access policies for your GCP resources.
- **Enforcer**: Create and Use policies to compare the current state of Compute Engine firewall to the desired state.
- **Explain**: Add-on module provides visibility into your Cloud Identity and Access Management (Cloud IAM) policies.
- **Email Notifications**: Send email notifications for Inventory and Scanner using the SendGrid API. SendGrid is currently the only supported email provider.

<ul style="list-style-type: none"> • The <code>'forseti'</code> section first creates a project which hosts Forseti services. 	<pre># Path to an empty YAML file in which DPT writes all the generated fields after successful deployment. These fields are used to generate monitoring rules.</pre>
--	---

- The 'devops' section creates a storage bucket mentioned under DevOps sub-section.
- The 'terraform_deployments' sub-section, first enables the Google Compute Engine API to create compute resources and then deploys a router and configures NAT.
- The Forseti Project can be used to monitor Google Cloud resources providing a single pane of view.

Additional configuration options:

- Use Cloud IAM to restrict access to specific users and groups.
- Set specific expiration time for data at a service level to control time period of retention
- Delete contents irretrievably if required through the template - to be used in specific use-cases only
- Enable versioning to retain older copies and serve as an audit trail.
- Utilize NAT to control and provide access to the internet.
- Enable Private IP to restrict SQL access to internal network

```

generated_fields_path:
  ./generated_fields.yaml

# Forseti section deploys a forseti image
which does security monitoring of GCP
resources
forseti:
  project:
    project_id: {{.FORSETI_PROJECT_ID}}
    owners_group:
    {{.FORSETI_PROJECT_OWNERS_GROUP}}
    auditors_group:
    {{.FORSETI_PROJECT_AUDITORS_GROUP}} #
  Auditors group at project level

  # Bigquery dataset stores audit logs from
  the forseti project and its resources.
  audit:
    logs_bigquery_dataset:
      dataset_id:
    {{.FORSETI_AUDIT_LOGS_BIGQUERY_DATASET_ID}}
      delete_contents_on_destroy:
    {{.FORSETI_DELETE_CONTENTS_ON_DESTROY}}
      access:
        - user_by_email:
    {{.FORSETIBQ_SERVICE_ACCOUNT_NAME}}@{{.FORSETI_PROJECT_ID}}.iam.gserviceaccount.com
          role: roles/bigquery.dataEditor
  # Can provide roles as per requirement.
    location: {{.LOCATION}}
    # default_encryption_configuration:
    #   kms_key_name:
  (ex. {google_kms_crypto_key.gcs.self_link})
    labels:
      data_criticality: medium
      datatype: auditlogs
      project: {{.FORSETI_PROJECT_ID}}

  # Storage bucket stores the terraform
  states of the resources in the projects.
  devops:
    state_storage_bucket:
      name:
    {{.FORSETI_STATE_STORAGE_BUCKET}}
      # encryption:
      #   default_kms_key_name:
  (ex. {google_kms_crypto_key.gcs.self_link})
  
```

```
location: {{.LOCATION}}

project_services:
- service: compute.googleapis.com
- service:
servicenetworking.googleapis.com

# Setup NAT to allow private forseti to
access the internet to fetch the Forseti repo
while
# having no external IP.
# See
https://github.com/forseti-security/terraform
-google-forseti/issues/234.
terraform_deployments:
resources:
config:
resource:
-
google_service_account_iam_binding:
admin-account-iam:
service_account_id:
${google_service_account.forsetibqsa.name}
role:
roles/iam.serviceAccountUser
members:
-
group:{{.FORSETI_PROJECT_OWNERS_GROUP}}
- google_service_account:
forsetibqsa:
account_id:
{{.FORSETIBQ_SERVICE_ACCOUNT_NAME}}
# Setting up VPC
- google_compute_network:
forseti_private_network:
name:
{{.FORSETI_VPC_NETWORK_NAME}}
auto_create_subnetworks:
false
- google_compute_subnetwork:
forseti_subnetwork:
- name:
{{.FORSETI_SUBNETWORK_NAME}}
network:
${google_compute_network.forseti_private_netw
ork.self_link}
region: {{.REGION}}
```

	<pre> ip_cidr_range: {{.FORSETI_SUBNET_IP_RANGE}} # (ex. 192.168.0.0/20) - google_compute_router: forseti-router: name: {{.FORSETI_ROUTER_NAME}} project: {{.FORSETI_PROJECT_ID}} network: \${google_compute_network.forseti_private_netw ork.self_link} region: {{.REGION}} - google_compute_router_nat: forseti-nat: name: {{.FORSETI_NAT_NAME}} project: {{.FORSETI_PROJECT_ID}} region: {{.REGION}} nat_ip_allocate_option: AUTO_ONLY source_subnetwork_ip_ranges_to_nat: ALL_SUBNETWORKS_ALL_IP_RANGES router: \${google_compute_router.forseti-router.name} properties: server_private: true client_private: true cloudsql_private: true network: \${google_compute_network.forseti_private_netw ork.name} subnetwork: \${google_compute_subnetwork.forseti_subnetwor k.name} </pre>
--	---

3.7.3 Data-Hosting Project Deployments

<ul style="list-style-type: none"> • This template includes the code for one data-hosting project. <ul style="list-style-type: none"> ○ It creates a new project if a project with the same name doesn't already exist in the organization. 	<pre> projects: - project_id: {{.RND_PROJECT_ID}} owners_group: {{.RND_OWNERS_GROUP}} auditors_group: {{.RND_AUDITORS_GROUP}} # Storage bucket stores the terraform states of the resources in the projects. devops: state_storage_bucket: </pre>
--	---

- In case of an existing project, it updates the project and its resources as per configuration.
- If additional data-hosting projects are required, additional projects with their corresponding resources can be added under the projects list of the existing data-hosting template and the template can be rerun to create them.
- It also adds an 'owner' and 'auditor' group for access control on the project.

```

    name: {{.RND_STATE_STORAGE_BUCKET}}
    # encryption:
    #   default_kms_key_name:
    (ex. {google_kms_crypto_key.gcs.self_link}
    )
    location: {{.LOCATION}}
    labels:
      data_criticality: low
      datatype: statefiles
      project: {{.RND_PROJECT_ID}}
    # Bigquery dataset and data storage
    bucket to store logs from projects and
    their resources.
    audit:
      logs_bigquery_dataset:
        dataset_id:
        {{.RND_AUDIT_LOGS_BIGQUERY_DATASET_ID}}
        # delete_contents_on_destroy: true
        access:
          - user_by_email:
            {{.RND_AUDIT_BQ_SERVICE_ACCOUNT_NAME}}@{{
            .RND_PROJECT_ID}}.iam.gserviceaccount.com
            # Can provide roles as per
            requirement.
            role: roles/bigquery.dataEditor
          - special_group:
            {{.RND_AUDIT_BQ_SPECIAL_GROUP}}
            role:
            {{.RND_AUDIT_BQ_SPECIAL_GROUP_ROLE}}
            # default_encryption_configuration:
            #   kms_key_name:
            (ex. {google_kms_crypto_key.gcs.self_link}
            )

    location: {{.LOCATION}}
    labels:
      data_criticality: medium
      datatype: auditlogs
      project: {{.RND_PROJECT_ID}}
    logs_storage_bucket:
      name:
      {{.RND_GCS_LOGS_STORAGE_BUCKET_NAME}}
      location: {{.LOCATION}}
      # encryption:
      #   default_kms_key_name:
      (ex. {google_kms_crypto_key.gcs.self_link}
      )
  
```

	<pre> lifecycle_rule: - condition: # Number of days from the time of creation, after which, storage objects from GCS logs bucket are moved to secondary storage class age: {{.RND_GCS_LOGS_AGE_FOR_SECONDARY_STORAGE _CLASS}} action: type: SetStorageClass # The storage class to which, objects from GCS logs bucket will be pushed to, after the specified number of days mentioned above (e.g. STANDARD/REGIONAL/MULTI_REGIONAL/COLDLINE /NEARLINE) storage_class: {{.RND_GCS_LOGS_SECONDARY_STORAGE_CLASS}} labels: data_criticality: medium datatype: gcslogs project: {{.RND_PROJECT_ID}} </pre>
<ul style="list-style-type: none"> • This section enables a list of APIs required by the data-hosting project. 	<pre> # APIs required by the workload and also other necessary APIs are enabled here project_services: - service: compute.googleapis.com - service: servicenetworking.googleapis.com - service: dataproc.googleapis.com - service: dlp.googleapis.com - service: speech.googleapis.com - service: lifesciences.googleapis.com - service: translate.googleapis.com - service: videointelligence.googleapis.com - service: vision.googleapis.com - service: language.googleapis.com - service: cloudfunctions.googleapis.com </pre>
<ul style="list-style-type: none"> • In this section, a BigQuery dataset is deployed under the same project. 	<pre> # Data with no sensitive information is stored for the use by data analysts bigquery_datasets: - dataset_id: {{.NON_SENSITIVE_DATA_BIGQUERY_DATASET_ID }} </pre>

<ul style="list-style-type: none"> • The BigQuery dataset stores non-sensitive information from the storage bucket and desensitized information from the healthcare dataset. • The template also accommodates provisioning of role-based access to individual resources through Google Cloud IAM. <p>Note: The configurations are customizable and can be changed as required to meet specific use-cases.</p>	<pre># delete_contents_on_destroy: true depends_on: - google_service_account.{{.NON_SENSITIVE_DATA_BQ_SERVICE_ACCOUNT_NAME}} access: - user_by_email: \${google_service_account.{{.NON_SENSITIVE_DATA_BQ_SERVICE_ACCOUNT_NAME}}.email} role: roles/bigquery.dataEditor # Can provide roles as per requirement. - special_group: {{.NON_SENSITIVE_DATA_BQ_SPECIAL_GROUP}} role: {{.NON_SENSITIVE_DATA_BQ_SPECIAL_GROUP_ROLE}} # default_encryption_configuration: # kms_key_name: (ex. {google_kms_crypto_key.gcs.self_link}) location: {{.LOCATION}} labels: data_criticality: {{.NON_SENSITIVE_DATA_BIGQUERY_DATASET_DATA_CRITICALITY_LABEL}} datatype: {{.NON_SENSITIVE_DATA_BIGQUERY_DATASET_DATA_TYPE_LABEL}} project: {{.RND_PROJECT_ID}}</pre>
<ul style="list-style-type: none"> • In this section, three Cloud Storage buckets are created under the same project for storing data generated by various healthcare sources. • Staging Bucket that stores the incoming health data containing PHI and other sensitive data. • DLP API segregation process on staging bucket data generates sensitive and non-sensitive data which are stored in the other two buckets separate from each other. • In each of the buckets, template assigns two roles - objectCreator 	<pre>storage_buckets: # Staging Bucket that stores the incoming health data containing PHI and other sensitive data - name: {{.STAGING_STORAGE_BUCKET_NAME}} # force_destroy: true versioning: enabled: true # IAM Role Binding _iam_members: - role: roles/storage.objectCreator member: {{.STAGING_STORAGE_BUCKET_OBJECTCREATOR}} - role: roles/storage.objectViewer member: {{.STAGING_STORAGE_BUCKET_OBJECTVIEWER}} # encryption:</pre>

and objectViewer to the specified members.

```
# default_kms_key_name:
(ex.{google_kms_crypto_key.gcs.self_link}
)
location: {{.LOCATION}}
labels:
  data_criticality:
  {{.STAGING_STORAGE_BUCKET_DATA_CRITICALITY_LABEL}}
  datatype:
  {{.STAGING_STORAGE_BUCKET_DATA_TYPE_LABEL}}
  project: {{.RND_PROJECT_ID}}
# Bucket that stores the objects with
sensitive data coming from staging bucket
- name:
  {{.SENSITIVE_DATA_STORAGE_BUCKET_NAME}}
  # force_destroy: true
  versioning:
    enabled: true
  # IAM Role Binding
  _iam_members:
    - role: roles/storage.objectCreator
      member:
  {{.SENSITIVE_DATA_STORAGE_BUCKET_OBJECTCREATOR}}
    - role: roles/storage.objectViewer
      member:
  {{.SENSITIVE_DATA_STORAGE_BUCKET_OBJECTVIEWER}}
  # encryption:
  # default_kms_key_name:
  (ex.{google_kms_crypto_key.gcs.self_link}
  )
  location: {{.LOCATION}}
  labels:
    data_criticality:
    {{.SENSITIVE_DATA_STORAGE_BUCKET_DATA_CRITICALITY_LABEL}}
    datatype:
    {{.SENSITIVE_DATA_STORAGE_BUCKET_DATA_TYPE_LABEL}}
    project: {{.RND_PROJECT_ID}}
# Bucket that stores the objects with no
sensitive data coming from staging bucket
- name:
  {{.NON_SENSITIVE_DATA_STORAGE_BUCKET_NAME}}
  }}
```

	<pre> # force_destroy: true versioning: enabled: true # IAM Role Binding _iam_members: - role: roles/storage.objectCreator member: {{.NON_SENSITIVE_DATA_STORAGE_BUCKET_OBJECTCREATOR}} - role: roles/storage.objectViewer member: {{.NON_SENSITIVE_DATA_STORAGE_BUCKET_OBJECTVIEWER}} # encryption: # default_kms_key_name: (ex. {google_kms_crypto_key.gcs.self_link}) location: {{.LOCATION}} labels: data_criticality: {{.NON_SENSITIVE_DATA_STORAGE_BUCKET_DATA_CRITICALITY_LABEL}} datatype: {{.NON_SENSITIVE_DATA_STORAGE_BUCKET_DATA_TYPE_LABEL}} project: {{.RND_PROJECT_ID}} </pre>
<ul style="list-style-type: none"> • In this section, healthcare data stores named DICOM store, FHIR store, and hl7-v2-store are created under a new healthcare dataset. • While these datasets are provisioned, the Cloud Healthcare API will also be enabled by the template. • Along with Google Cloud IAM, access to the API can be restricted to a limited set of users based on roles, responsibilities, and dataset. 	<pre> # Data from sensitive data storage bucket gets transferred here for deanonymization and other cleanup tasks by Healthcare API healthcare_datasets: - name: {{.HEALTHCARE_DATASET_NAME}} location: {{.REGION}} # IAM Role Binding _iam_members: - role: roles/healthcare.datasetViewer member: {{.HEALTHCARE_DATASET_VIEWER}} _dicom_stores: - name: {{.HEALTHCARE_DICOM_STORE_NAME}} _iam_members: - role: roles/healthcare.dicomEditor member: {{.HEALTHCARE_DICOM_EDITOR}} - role: roles/healthcare.dicomStoreAdmin </pre>

	<pre> member: {{.HEALTHCARE_DICOM_STOREADMIN}} _fhir_stores: - name: {{.HEALTHCARE_FHIR_STORE_NAME}} _iam_members: - role: roles/healthcare.fhirResourceReader member: {{.HEALTHCARE_FHIR_STORE_READER}} - role: roles/healthcare.fhirResourceEditor member: {{.HEALTHCARE_FHIR_STORE_EDITOR}} _hl7_v2_stores: - name: {{.HEALTHCARE_HL7V2_STORE_NAME}} _iam_members: - role: roles/healthcare.hl7V2StoreAdmin member: {{.HEALTHCARE_HL7V2_STOREADMIN}} - role: roles/healthcare.hl7V2Ingest member: {{.HEALTHCARE_HL7V2_INGEST}} - role: roles/healthcare.hl7V2Editor member: {{.HEALTHCARE_HL7V2_STORE_EDITOR}} </pre>
<ul style="list-style-type: none"> ● In this section, two Pub/Sub topics are created with their corresponding subscription. ● Data classification labels of objects from the staging bucket are pushed to the first topic by the DLP task, for their consumption by objects sorting Cloud Function. ● Cloud IAM is applied to the topic as well as subscription. ● The messages qualified as dead letters are published to the second topic (dead letter topic). 	<pre> # Data classification labels of object # from staging bucket are pushed to this # topic by the DLP task, # for their consumption by objects # sorting cloud function pubsub_topics: - name: {{.PUB_SUB_TOPIC_NAME}} # IAM Role Binding _iam_members: - role: roles/pubsub.editor member: {{.PUB_SUB_TOPIC_EDITOR_ROLE_USER}} - role: roles/pubsub.viewer member: {{.PUB_SUB_TOPIC_VIEWER_ROLE_USER}} labels: data_criticality: {{.PUB_SUB_TOPIC_CRITICALITY_LABEL}} datatype: </pre>

```
{{.PUB_SUB_TOPIC_DATA_TYPE_LABEL}}
  project: {{.RND_PROJECT_ID}}
  _subscriptions:
  - name: {{.PUB_SUB_SUBSCRIPTION_NAME}}
    # Duration of the messages stored in
    the regions.
    message_retention_duration:
    {{.PUB_SUB_SUBSCRIPTION_MESSAGE_RETENTION
    _DURATION}}
    # Whether acked messages are retained
    or not.
    retain_acked_messages:
    {{.PUB_SUB_SUBSCRIPTION_RETAIN_ACKED_MESS
    AGES}}
    # This value is the maximum time
    after a subscriber receives a message
    before the subscriber should acknowledge
    the message.
    ack_deadline_seconds:
    {{.PUB_SUB_SUBSCRIPTION_ACK_DEADLINE_SECO
    NDS}}
    # Policy defining the guidelines for
    expiration of the subscription.
    expiration_policy:
    # Time-to-Live duration of
    associated resources.
    ttl:
    {{.PUB_SUB_SUBSCRIPTION_TIME_TO_LIVE}}
    # IAM Role Binding
    _iam_members:
    - role: roles/pubsub.subscriber
      member:
    {{.PUB_SUB_SUBSCRIPTION_SUBSCRIBER_ROLE_U
    SER}}
    - role: roles/pubsub.editor
      member:
    {{.PUB_SUB_SUBSCRIPTION_EDITOR_ROLE_USER
    }}
    # A policy that specifies the
    conditions for dead lettering messages in
    this subscription. If dead_letter_policy
    is not set, dead lettering is disabled.
    dead_letter_policy:
    dead_letter_topic:
    ${google_pubsub_topic.{{.PUB_SUB_DEAD_LET
    TER_TOPIC_NAME}}.id}
    max_delivery_attempts: 10
```

```

    # If push delivery is used with this
    subscription, uncomment the following
    field used to configure it. Refer to
    "https://www.terraform.io/docs/providers/
    google/r/pubsub_subscription.html" for
    more information.
    # push_config:
    #   oidc_token:
    #     service_account_email:
    ${google_service_account.forsetibqsa.name
    }
    #   audience: (ex.
    https://example.com/push)
    #   push_endpoint: (ex.
    https://example.com/push)
    #   attributes: (ex. x-goog-version)

- name:
  {{.PUB_SUB_DEAD_LETTER_TOPIC_NAME}}
  # IAM Role Binding
  _iam_members:
  - role: roles/pubsub.editor
    member:
  {{.PUB_SUB_DEAD_LETTER_EDITOR_ROLE_USER}}
  - role: roles/pubsub.viewer
    member:
  {{.PUB_SUB_DEAD_LETTER_VIEWER_ROLE_USER}}
  _subscriptions:
  - name:
  {{.PUB_SUB_DEAD_LETTER_SUBSCRIPTION_NAME}}
  }

    # Duration of the messages stored in
    the regions.
    message_retention_duration:
  {{.PUB_SUB_DEAD_LETTER_SUBSCRIPTION_MESSA
  GE_RETENTION_DURATION}}
    # Whether acked messages are retained
    or not.
    retain_acked_messages:
  {{.PUB_SUB_DEAD_LETTER_SUBSCRIPTION_RETAI
  N_ACKED_MESSAGES}}
    # This value is the maximum time
    after a subscriber receives a message
    before the subscriber should acknowledge
    the message.
    ack_deadline_seconds:
  {{.PUB_SUB_DEAD_LETTER_SUBSCRIPTION_ACK_D

```

	<pre> EADLINE_SECONDS}} # Policy defining the guidelines for expiration of the subscription. expiration_policy: # Time-to-Live duration of associated resources. ttl: {{.PUB_SUB_DEAD_LETTER_SUBSCRIPTION_TIME_ TO_LIVE}} # IAM Role Binding _iam_members: - role: roles/pubsub.subscriber member: {{.PUB_SUB_DEAD_LETTER_SUBSCRIPTION_SUBSC RIBER_ROLE_USER}} - role: roles/pubsub.editor member: {{.PUB_SUB_DEAD_LETTER_SUBSCRIPTION_EDITO R_ROLE_USER}} # If push delivery is used with this subscription, uncomment the following field used to configure it. Refer to "https://www.terraform.io/docs/providers/ google/r/pubsub_subscription.html" for more information. # push_config: # oidc_token: # service_account_email: \${google_service_account.forsetibqsa.name } # audience: (ex. https://example.com/push) # push_endpoint: (ex. https://example.com/push) # attributes: (ex. x-goog-version) </pre>
<ul style="list-style-type: none"> • This section creates the custom services accounts for the resources and provides the owners group serviceAccountUser role on the service accounts. • It also provides IAM bindings to the default created service accounts for DLP and APP Engine Admin APIs using gcloud. 	<pre> terraform_deployments: resources: config: resource: # Role bindings for the service accounts - google_service_account_iam_binding: analysts-gke-cluster-sa-iam: service_account_id: \${google_service_account.{{.ANALYSTS_GKE_ </pre>

	<pre> CLUSTER_SERVICE_ACCOUNT_NAME}}.name} role: roles/iam.serviceAccountUser members: - group:{{.RND_OWNERS_GROUP}} researchers-gke-cluster-sa-iam: service_account_id: \${google_service_account.{{.RESEARCHERS_G KE_CLUSTER_SERVICE_ACCOUNT_NAME}}.name} role: roles/iam.serviceAccountUser members: - group:{{.RND_OWNERS_GROUP}} # Adding IAM Policy Bindings for service accounts created by default by DLP and APP Engine Admin APIs - null_resource: app_engine_sa_roles: provisioner: local-exec: command: app_sa=\$(gcloud projects get-iam-policy {{.RND_PROJECT_ID}} grep appspot sed 's/^.*\({{.RND_PROJECT_ID}}.*\)/\1/g' sed '2d'); \ gcloud projects add-iam-policy-binding {{.RND_PROJECT_ID}} --member serviceAccount:\$app_sa --role roles/owner; \ gcloud projects add-iam-policy-binding {{.RND_PROJECT_ID}} --member serviceAccount:\$app_sa --role roles/dlp.admin dlp_sa_roles: provisioner: local-exec: command: dlp_sa=\$(gcloud projects get-iam-policy {{.RND_PROJECT_ID}} grep dlp-api sed 's/^.*\({{.RND_PROJECT_ID}}.*\)/\1/g' sed '2d'); \ gcloud projects add-iam-policy-binding {{.RND_PROJECT_ID}} --member </pre>
--	---

	<pre>serviceAccount:\$dlp_sa --role roles/viewer</pre>
<ul style="list-style-type: none"> • This section deploys a VPC network with two subnetworks, one for each Kubernetes cluster. • These Kubernetes clusters are attached to a private VPC network through a dedicated subnet which effectively restricts access and protects them against exposure to public networks. 	<pre>terraform_deployments: resources: config: resource: # Setting up VPC network - google_compute_network: private_network: name: {{.RND_PRIVATE_VPC_NETWORK_NAME}} auto_create_subnetworks: false # Creating a Subnetwork for the use of GKE Cluster - google_compute_subnetwork: - analysts-gke-cluster-subnetwork: - name: {{.ANALYSTS_GKE_CLUSTER_NAME}} network: \${google_compute_network.private_network. self_link} region: {{.REGION}} ip_cidr_range: {{.ANALYSTS_GKE_CLUSTER_SUBNET_IP_RANGE}} # (ex. 192.168.0.0/20) private_ip_google_access: true secondary_ip_range: - range_name: analysts-gke-cluster-pods-range ip_cidr_range: {{.ANALYSTS_GKE_CLUSTER_PODS_IP_RANGE}} # (ex. 10.4.0.0/14) - range_name: analysts-gke-cluster-services-range ip_cidr_range: {{.ANALYSTS_GKE_CLUSTER_SERVICES_IP_RANGE }} # (eg. 10.0.32.0/20) - researchers-gke-cluster-subnetwork: - name: researchers-gke-cluster-subnet network: \${google_compute_network.private_network.</pre>

	<pre> self_link} region: {{.REGION}} ip_cidr_range: {{.RESEARCHERS_GKE_CLUSTER_SUBNET_IP_RANGE}} # (ex. 192.168.0.0/20) private_ip_google_access: true secondary_ip_range: - range_name: researchers-gke-cluster-pods-range ip_cidr_range: {{.RESEARCHERS_GKE_CLUSTER_PODS_IP_RANGE}} # (ex. 10.4.0.0/14) - range_name: researchers-gke-cluster-services-range ip_cidr_range: {{.RESEARCHERS_GKE_CLUSTER_SERVICES_IP_RANGE}} # (eg. 10.0.32.0/20) </pre>
<ul style="list-style-type: none"> • This section deploys two Kubernetes clusters with corresponding node pools. • One cluster containing pre-installed tools and libraries uses non sensitive information for analysis purposes. This cluster is accessible to analysts. • Second cluster, pre-tailored for research purposes, can have sensitive information. These are only accessible to the researchers. 	<pre> terraform_deployments: resources: config: resource: - google_container_cluster: # GKE Cluster dedicated for containers used by Researchers cluster_for_researchers: name: {{.RESEARCHERS_GKE_CLUSTER_NAME}} location: {{.REGION}} remove_default_node_pool: true initial_node_count: 1 default_max_pods_per_node: {{.MAX_PODS_PER_NODE_RESEARCHERS_GKE_CLUSTER}} enable_binary_authorization: true #enable_shielded_nodes: true logging_service: logging.googleapis.com/kubernetes # Available options include logging.googleapis.com(Legacy Stackdriver), logging.googleapis.com/kubernetes(Stackdriver Kubernetes Engine Logging), and none. </pre>

```
        # Uncomment the following
        field for maintenance configurations.
        Refer to
        "https://www.terraform.io/docs/providers/
        google/r/container_cluster.html" for
        field values information.
        # maintenance_policy:
        #   daily_maintenance_window:
        #
        -----
        ----
        #
        daily_maintenance_window:
        #   start_time: 03:00
        #   ----- OR
        -----
        #   recurring_window:
        #   start_time:
        2019-01-01T09:00:00-04:00
        #   end_time:
        2019-01-01T17:00:00-04:00
        #   recurrence:
        FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR
        #
        -----
        ----

        # Available options include
        monitoring.googleapis.com(Legacy
        Stackdriver),
        monitoring.googleapis.com/kubernetes(Stac
        kdriver Kubernetes Engine Monitoring),
        and none.
        monitoring_service:
        monitoring.googleapis.com/kubernetes

        master_authorized_networks_config:
        # External networks that
        can access the Kubernetes cluster master
        through HTTPS.
        cidr_blocks:
        - cidr_block:
        {{.RESEARCHERS_GKE_CLUSTER_EXTERNAL_NETWO
        RK_CIDR}}
        resource_labels:
        data_criticality:
        {{.RESEARCHERS_GKE_CLUSTER_DATA_CRITICALI
```

```
TY_LABEL}}
    datatype:
{{.RESEARCHERS_GKE_CLUSTER_DATASET_DATA_T
YPE_LABEL}}
    project:
{{.RND_PROJECT_ID}}
    #
authenticator_groups_config:
    # The name of the RBAC
security group for use with Google
security groups in Kubernetes RBAC. Group
name must be in format
gke-security-groups@yourdomain.com.
    # security_group: (ex.
gke-security-groups@yourdomain.com)
    # Network and Subnetwork used
by this GKE cluster
    network:
${google_compute_network.private_network.
self_link}
    subnetwork:
${google_compute_subnetwork.researchers-g
ke-cluster-subnetwork.self_link}
    # The authentication
information for accessing the Kubernetes
master
    master_auth:
    username:
'{{.RESEARCHERS_GKE_CLUSTER_AUTH_USERNAME
}}'
    password:
'{{.RESEARCHERS_GKE_CLUSTER_AUTH_PASSWORD
}}'
    client_certificate_config:
    issue_client_certificate:
false
    # Configuration for private
cluster with private nodes
    private_cluster_config:
    # Enables the private
cluster feature, creating a private
endpoint on the cluster
    enable_private_nodes: true
    # The cluster's private
endpoint is used as the cluster endpoint
and access through the public endpoint is
disabled when true
```

	<pre>enable_private_endpoint: true # The IP range in CIDR notation to use for the hosted master network. The range should not overlap with an existing subnet. master_ipv4_cidr_block: {{.RESEARCHERS_GKE_CLUSTER_MASTER_IPV4_CI DR_BLOCK}} # (ex. 172.16.0.32/28) # Configuration of cluster IP allocation for VPC-native clusters. Adding this block enables IP aliasing, # making the cluster VPC-native instead of routes-based. ip_allocation_policy: # The name of the existing secondary range in the cluster's subnetwork to use for pod IP addresses. cluster_secondary_range_name: researchers-gke-cluster-pods-range # The name of the existing secondary range in the cluster's subnetwork to use for service cluster IPs. services_secondary_range_name: researchers-gke-cluster-services-range # Cluster configuration of Node Auto-Provisioning with Cluster Autoscaler to automatically adjust the size of the cluster cluster_autoscaling: enabled: true #autoscaling_profile: BALANCED # (or OPTIMIZE_UTILIZATION) # Limits on CPU and Memory for the cluster resource_limits: - resource_type: memory minimum: {{.RESEARCHERS_GKE_CLUSTER_MIN_MEMORY}} maximum: {{.RESEARCHERS_GKE_CLUSTER_MAX_MEMORY}} - resource_type: cpu minimum:</pre>
--	---

	<pre> {{.RESEARCHERS_GKE_CLUSTER_MIN_CPU}} maximum: {{.RESEARCHERS_GKE_CLUSTER_MAX_CPU}} # Contains defaults for a node pool created by Node Auto-Provisioning. auto_provisioning_defaults: service_account: \${google_service_account.{{.RESEARCHERS_G KE_CLUSTER_SERVICE_ACCOUNT_NAME}}.email} oauth_scopes: - https://www.googleapis.com/auth/bigquery. readonly # database_encryption: # state: (ex. ENCRYPTED or DECRYPTED) # key_name: (ex. projects/my-project/locations/global/keyR ings/my-ring/cryptoKeys/my-key or {google_kms_crypto_key.gcs.self_link}) # Refer "https://cloud.google.com/kubernetes-engi ne/docs/reference/rest/v1beta1/projects.l ocations.clusters#Cluster.DatabaseEncrypt ion" # GKE Cluster dedicated for containers used by Analysts cluster_for_analysts: name: {{.ANALYSTS_GKE_CLUSTER_NAME}} location: {{.REGION}} remove_default_node_pool: true initial_node_count: 1 default_max_pods_per_node: {{.MAX_PODS_PER_NODE_ANALYSTS_GKE_CLUSTER }} enable_binary_authorization: true #enable_shielded_nodes: true logging_service: logging.googleapis.com/kubernetes # Available options include logging.googleapis.com(Legacy Stackdriver), logging.googleapis.com/kubernetes(Stackdr </pre>
--	--

	<pre> iver Kubernetes Engine Logging), and none. # Uncomment the following field for maintenance configurations. Refer to "https://www.terraform.io/docs/providers/ google/r/container_cluster.html" for field values information. # maintenance_policy: # daily_maintenance_window: # ----- ----- # daily_maintenance_window: # start_time: 03:00 # ----- OR ----- # recurring_window: # start_time: 2019-01-01T09:00:00-04:00 # end_time: 2019-01-01T17:00:00-04:00 # recurrence: FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR # ----- ----- # Available options include monitoring.googleapis.com(Legacy Stackdriver), monitoring.googleapis.com/kubernetes(Stac kdriver Kubernetes Engine Monitoring), and none. monitoring_service: monitoring.googleapis.com/kubernetes master_authorized_networks_config: # External networks that can access the Kubernetes cluster master through HTTPS. cidr_blocks: - cidr_block: {{.ANALYSTS_GKE_CLUSTER_EXTERNAL_NETWORK_ CIDR}} resource_labels: </pre>
--	--

```
        data_criticality:
{{.ANALYSTS_GKE_CLUSTER_DATA_CRITICALITY_
LABEL}}
        datatype:
{{.ANALYSTS_GKE_CLUSTER_DATASET_DATA_TYPE
_LABEL}}
        project:
{{.RND_PROJECT_ID}}
        #
authenticator_groups_config:
        # The name of the RBAC
security group for use with Google
security groups in Kubernetes RBAC. Group
name must be in format
gke-security-groups@yourdomain.com.
        # security_group: (ex.
gke-security-groups@yourdomain.com)
        # Network and Subnetwork used
by this GKE cluster
        network:
${google_compute_network.private_network.
self_link}
        subnetwork:
${google_compute_subnetwork.analysts-gke-
cluster-subnetwork.self_link}
        # The authentication
information for accessing the Kubernetes
master
        master_auth:
        username:
'{{.ANALYSTS_GKE_CLUSTER_AUTH_USERNAME}}'
        password:
'{{.ANALYSTS_GKE_CLUSTER_AUTH_PASSWORD}}'
        client_certificate_config:
        issue_client_certificate:
false
        # Configuration for private
cluster with private nodes
        private_cluster_config:
        # Enables the private
cluster feature, creating a private
endpoint on the cluster
        enable_private_nodes: true
        # The cluster's private
endpoint is used as the cluster endpoint
and access through the public endpoint is
disabled when true
```


	<pre>enable_private_endpoint: true # The IP range in CIDR notation to use for the hosted master network. The range should not overlap with an existing subnet. master_ipv4_cidr_block: {{.ANALYSTS_GKE_CLUSTER_MASTER_IPV4_CIDR_ BLOCK}} # (ex. 172.16.0.32/28) # Configuration of cluster IP allocation for VPC-native clusters. Adding this block enables IP aliasing, # making the cluster VPC-native instead of routes-based. ip_allocation_policy: # The name of the existing secondary range in the cluster's subnetwork to use for pod IP addresses. cluster_secondary_range_name: analysts-gke-cluster-pods-range # The name of the existing secondary range in the cluster's subnetwork to use for service cluster IPs. services_secondary_range_name: analysts-gke-cluster-services-range # Cluster configuration of Node Auto-Provisioning with Cluster Autoscaler to automatically adjust the size of the cluster cluster_autoscaling: enabled: true #autoscaling_profile: BALANCED # (or OPTIMIZE_UTILIZATION) # Limits on CPU and Memory for the cluster resource_limits: - resource_type: memory minimum: {{.ANALYSTS_GKE_CLUSTER_MIN_MEMORY}} maximum: {{.ANALYSTS_GKE_CLUSTER_MAX_MEMORY}} - resource_type: cpu minimum:</pre>
--	--

	<pre> {{.ANALYSTS_GKE_CLUSTER_MIN_CPU}} maximum: {{.ANALYSTS_GKE_CLUSTER_MAX_CPU}} # Contains defaults for a node pool created by Node Auto-Provisioning. auto_provisioning_defaults: service_account: \${google_service_account.{{.ANALYSTS_GKE_ CLUSTER_SERVICE_ACCOUNT_NAME}}.email} oauth_scopes: - https://www.googleapis.com/auth/bigquery. readonly # database_encryption: # state: (ex. ENCRYPTED or DECRYPTED) # key_name: (ex. projects/my-project/locations/global/keyR ings/my-ring/cryptoKeys/my-key or {google_kms_crypto_key.gcs.self_link}) # Refer "https://cloud.google.com/kubernetes-engi ne/docs/reference/rest/v1beta1/projects.l ocations.clusters#Cluster.DatabaseEncrypt ion" - google_container_node_pool: researchers_cluster_preemptible_nodes: name: researchers-gke-cluster-node-pool location: {{.REGION}} cluster: \${google_container_cluster.cluster_for_re searchers.name} node_count: 1 # Node management configuration, wherein auto-repair and auto-upgrade is configured. management: auto_repair: true auto_upgrade: true # Configuration required by cluster autoscaler to adjust the size of the node pool to the current cluster usage. autoscaling: </pre>
--	---

	<pre> min_node_count: 1 max_node_count: {{.MAX_RESEARCHERS_GKE_CLUSTER_NODE_COUNT }} # Node configuration of the pool. node_config: preemptible: true machine_type: n1-standard-1 metadata: disable-legacy-endpoints: 'true' oauth_scopes: - https://www.googleapis.com/auth/logging.w rite - https://www.googleapis.com/auth/monitorin g analysts_cluster_preemptible_nodes: name: analysts-gke-cluster-node-pool location: {{.REGION}} cluster: \${google_container_cluster.cluster_for_an alysts.name} node_count: 1 # Node management configuration, wherein auto-repair and auto-upgrade is configured. management: auto_repair: true auto_upgrade: true # Configuration required by cluster autoscaler to adjust the size of the node pool to the current cluster usage. autoscaling: min_node_count: 1 max_node_count: {{.MAX_ANALYSTS_GKE_CLUSTER_NODE_COUNT}} # Node configuration of the pool. node_config: preemptible: true machine_type: n1-standard-1 </pre>
--	--

	<pre> metadata: disable-legacy-endpoints: 'true' oauth_scopes: - https://www.googleapis.com/auth/logging.w rite - https://www.googleapis.com/auth/monitorin g </pre>
<ul style="list-style-type: none"> • This section deploys two Cloud Functions. • The first Cloud Function runs a DLP job to generate classification labels (Sensitive / Non-sensitive) for objects in Staging Storage Bucket and pushes messages to Pub/Sub. It is triggered when data is loaded into Staging Storage Bucket from the healthcare data sources. • The second Cloud Function fetches classification labels (Sensitive / Non-sensitive) for objects in Staging Storage Bucket from Pub/Sub and sorts them into Non-sensitive and Sensitive data storage buckets. 	<pre> terraform_deployments: resources: config: resource: - google_cloudfunctions_function: # Cloud function that runs DLP job to generate classification labels (Sensitive/Non-sensitive) for objects in Staging Storage Bucket and pushes messages to PubSub create_DLP_job: name: create-DLP-job description: This function is triggered by new files uploaded to the designated Cloud Storage staging bucket. runtime: python37 available_memory_mb: 128 # Place where the code for the cloud function is stored source_archive_bucket: {{.CLOUD_FUNCTION_ZIP_BUCKET}} source_archive_object: {{.CLOUD_FUNCTION_ZIP_OBJECT}} timeout: 60 region: {{.REGION}} # support only for us-central1 at present entry_point: create_DLP_job event_trigger: event_type: google.storage.object.finalize resource: {{.STAGING_STORAGE_BUCKET_NAME}} failure_policy: retry: true environment_variables: YOUR_QUARANTINE_BUCKET: </pre>

```

{{.STAGING_STORAGE_BUCKET_NAME}}
    YOUR_SENSITIVE_DATA_BUCKET:
{{.SENSITIVE_DATA_STORAGE_BUCKET_NAME}}

YOUR_NON_SENSITIVE_DATA_BUCKET:
{{.NON_SENSITIVE_DATA_STORAGE_BUCKET_NAME
}}

PROJECT_ID_HOSTING_STAGING_BUCKET:
{{.RND_PROJECT_ID}}
    PUB_SUB_TOPIC:
{{.PUB_SUB_TOPIC_NAME}}
    # Cloud function that fetches
classification labels
(Sensitive/Non-sensitive) for objects in
Staging Storage Bucket from
    # Pubsub and sort them into
Non-sensitive and Sensitive data storage
buckets.
    resolve_DLP:
        name: resolve-DLP
        description: This function
listens to the pub/sub notification from
the create_DLP_job function.
        runtime: python37
        available_memory_mb: 128
        # Place where the code for
the cloud function is stored
        source_archive_bucket:
{{.CLOUD_FUNCTION_ZIP_BUCKET}}
        source_archive_object:
{{.CLOUD_FUNCTION_ZIP_OBJECT}}
        timeout: 60
        region: us-central1 # support
only for us-central1 at present
        entry_point: resolve_DLP
        event_trigger:
            event_type:
google.pubsub.topic.publish
            resource:
{{.PUB_SUB_TOPIC_NAME}}
        failure_policy:
            retry: true
        environment_variables:
            YOUR_QUARANTINE_BUCKET:
{{.STAGING_STORAGE_BUCKET_NAME}}
            YOUR_SENSITIVE_DATA_BUCKET:

```

```

{{ .SENSITIVE_DATA_STORAGE_BUCKET_NAME }}

YOUR_NON_SENSITIVE_DATA_BUCKET :
{{ .NON_SENSITIVE_DATA_STORAGE_BUCKET_NAME
}}

PROJECT_ID_HOSTING_STAGING_BUCKET :
{{ .RND_PROJECT_ID }}
PUB_SUB_TOPIC :
{{ .PUB_SUB_TOPIC_NAME }}

```

Note: The deployment template can have the deployment variables statically declared or passed during execution. The template above implements the latter - it integrates with another template which contains all the variable declarations required during the runtime. This helps in separating the code from the variables, enabling re-usability, modularity and security. To learn more about modularization and reusability of DPT templates, please refer [here](#).

3.8 Post-Deployment Verification

Post-deployment of the DPT template, two projects are created:

- Primary data project (“dpth-ai-ml” in this example)
- A separate Forseti project for monitoring (“forseti-project-id” in this example)

3.8.1 Forseti Project

Cloud Console Dashboard: Using the Cloud Console, a list of projects that are deployed can be viewed and selected. Under the Forseti project (forseti-project-id), the project information and the resources that are deployed using DPT are listed as highlighted in Figure 2.

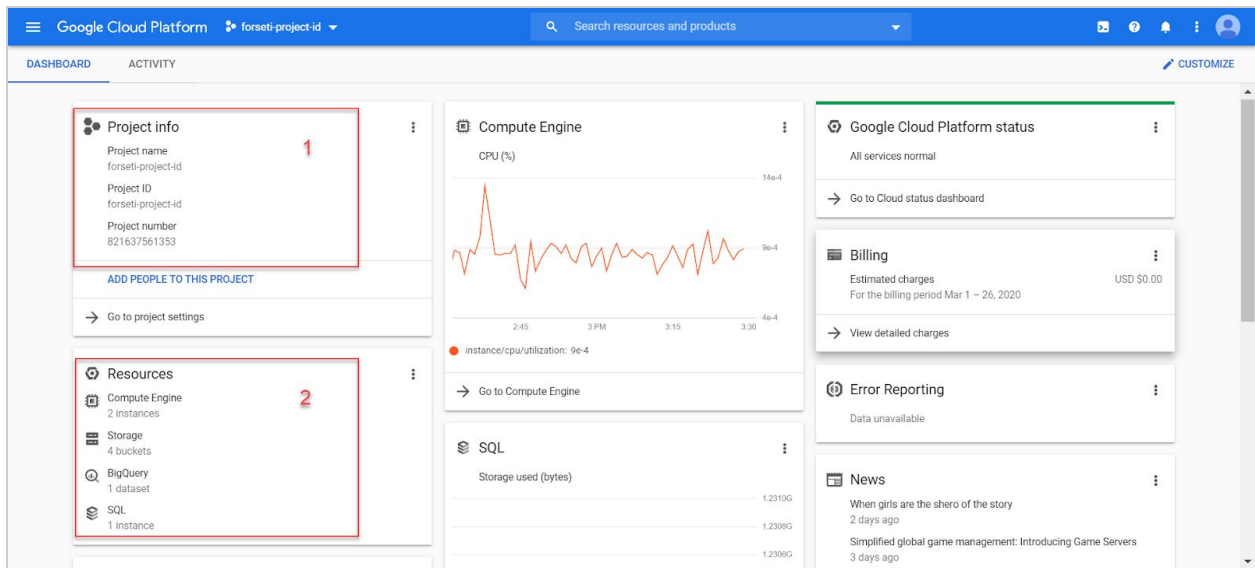


Figure 2 - 1. Forseti Project Info 2. Forseti Project Resources

IAM Console: The IAM permissions can be viewed through the IAM console (Figure 3). Here, the Forseti service accounts are created, and the appropriate roles are assigned as defined in the template. The Storage Object Viewer role is provided to one of the service accounts, and the AppEngine Viewer role is provided to the other account.

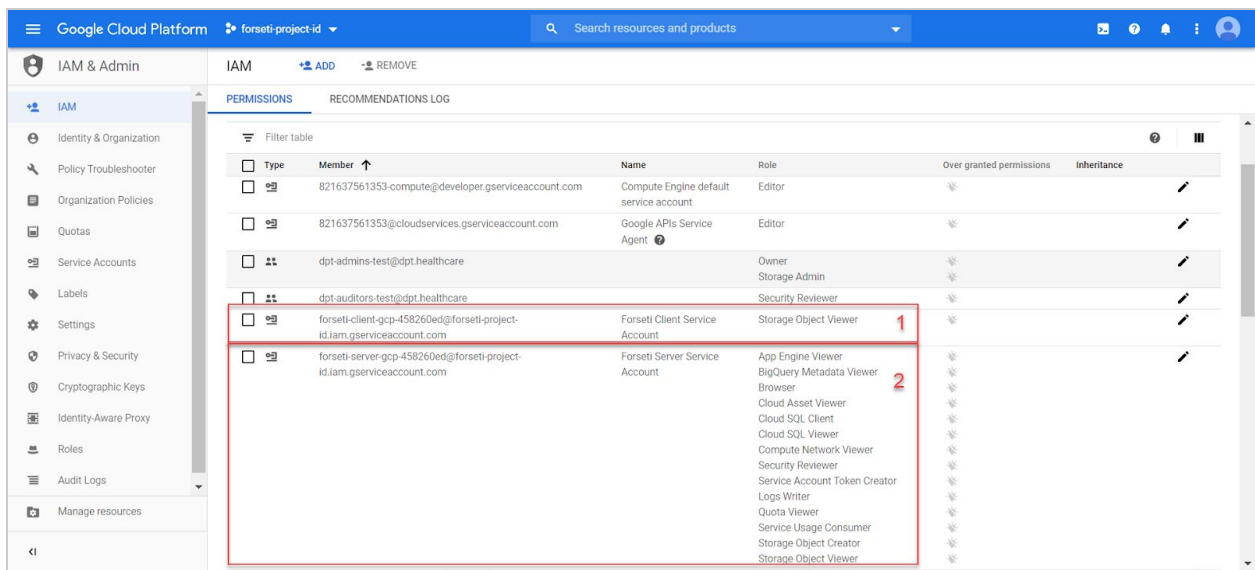


Figure 3 - 1. Forseti Client Service Account 2. Forseti Server Service Account

Compute Engine Console: View the list of servers created on the 'VM instances' console. Two Forseti Virtual Machine (VM) instances are created as a part of the Forseti project. The list of permissions on the VM's, as highlighted in Figure 4 and 5, can be viewed by selecting the VM.

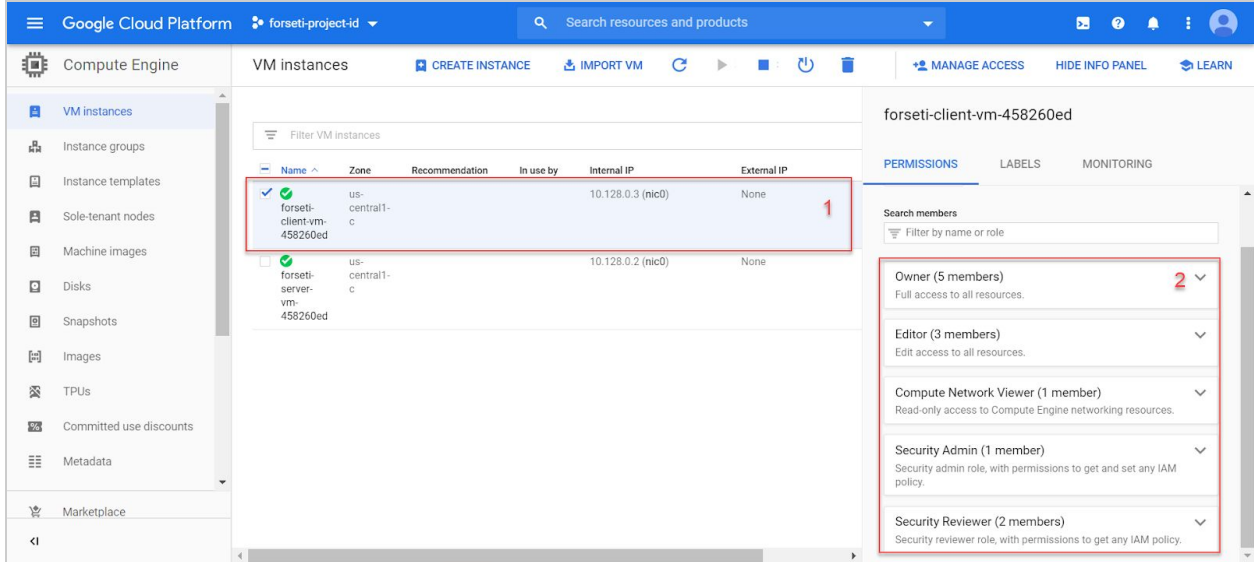


Figure 4 - 1. Forseti Client Instance 2. Instance Permissions

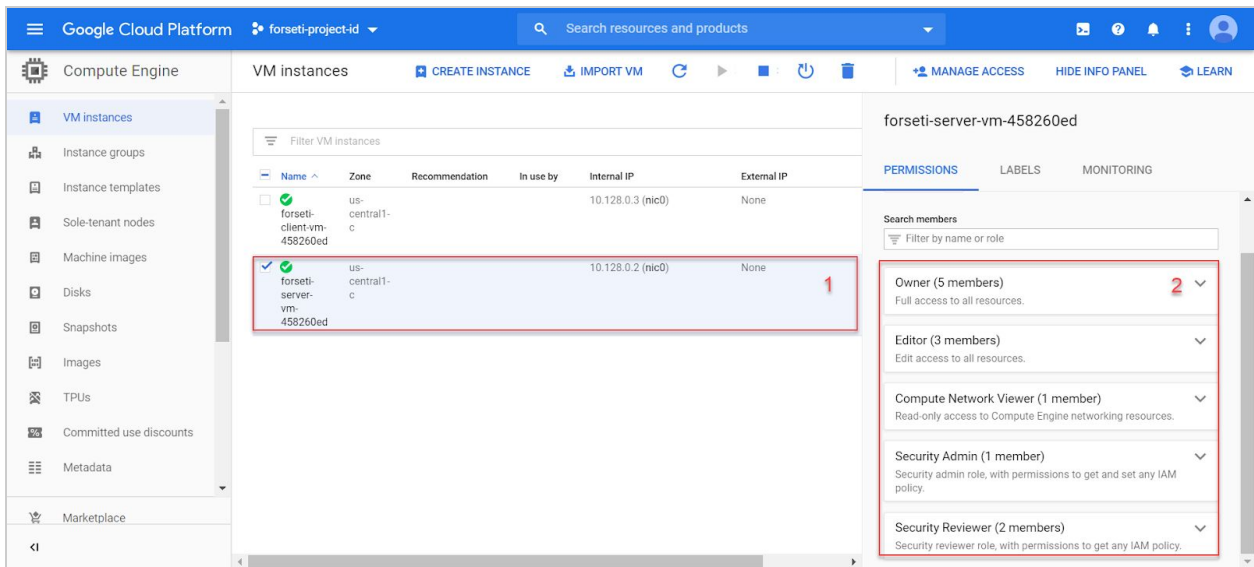


Figure 5 - 1. Forseti Server Instance 2. Instance Permissions

VPC Network Console: Figure 6 shows the Virtual Private Network created for Forseti on the [VPC network Console](#).

Region	Network Name	Mode	IP Range 1	IP Range 2	Private
asia-southeast1	default	10.148.0.0/20	10.148.0.1	Off	
us-east4	default	10.150.0.0/20	10.150.0.1	Off	
australia-southeast1	default	10.152.0.0/20	10.152.0.1	Off	
europa-west2	default	10.154.0.0/20	10.154.0.1	Off	
europa-west3	default	10.156.0.0/20	10.156.0.1	Off	
southamerica-east1	default	10.158.0.0/20	10.158.0.1	Off	
asia-south1	default	10.160.0.0/20	10.160.0.1	Off	
northamerica-northeast1	default	10.162.0.0/20	10.162.0.1	Off	
europa-west4	default	10.164.0.0/20	10.164.0.1	Off	
europa-north1	default	10.166.0.0/20	10.166.0.1	Off	
us-west2	default	10.168.0.0/20	10.168.0.1	Off	
asia-east2	default	10.170.0.0/20	10.170.0.1	Off	
europa-west6	default	10.172.0.0/20	10.172.0.1	Off	
asia-northeast2	default	10.174.0.0/20	10.174.0.1	Off	
asia-northeast3	default	10.178.0.0/20	10.178.0.1	Off	
us-west3	default	10.180.0.0/20	10.180.0.1	Off	
private-vpc-networks	0	Custom	0	Off	

Figure 6 - Private VPC

The firewall rules section on the [VPC network Console](#) displays firewall rules created as a part of this project (Figure 7). The 'Firewall rules' console displays the Rule name, rule type selected (Ingress/Egress), IP ranges, and the Instance to which the firewall rules are applied.

Name	Type	Targets	Filters	Protocols / ports	Action	Priority	Network
forseti-client-ssh-external-458260ed	Ingress	forseti-client-gcp-458260ed	IP ranges: 0.0.0.0/0	tcp:22	Allow	100	default
forseti-server-allow-grpc-458260ed	Ingress	forseti-server-gcp-458260ed	IP ranges: 10.128.0.0/9	tcp:50051,50052	Allow	100	default
forseti-server-ssh-external-458260ed	Ingress	forseti-server-gcp-458260ed	IP ranges: 0.0.0.0/0	tcp:22	Allow	100	default
forseti-client-deny-all-458260ed	Ingress	forseti-client-gcp-458260ed	IP ranges: 0.0.0.0/0	tcp udp icmp	Deny	200	default
forseti-server-deny-all-458260ed	Ingress	forseti-server-gcp-458260ed	IP ranges: 0.0.0.0/0	tcp udp icmp	Deny	200	default

Figure 7 - Forseti Firewall Rules

Network Services Console: The Cloud NAT created under the Forseti project is deployed on the [Network Services Console](#) (Figure 8).

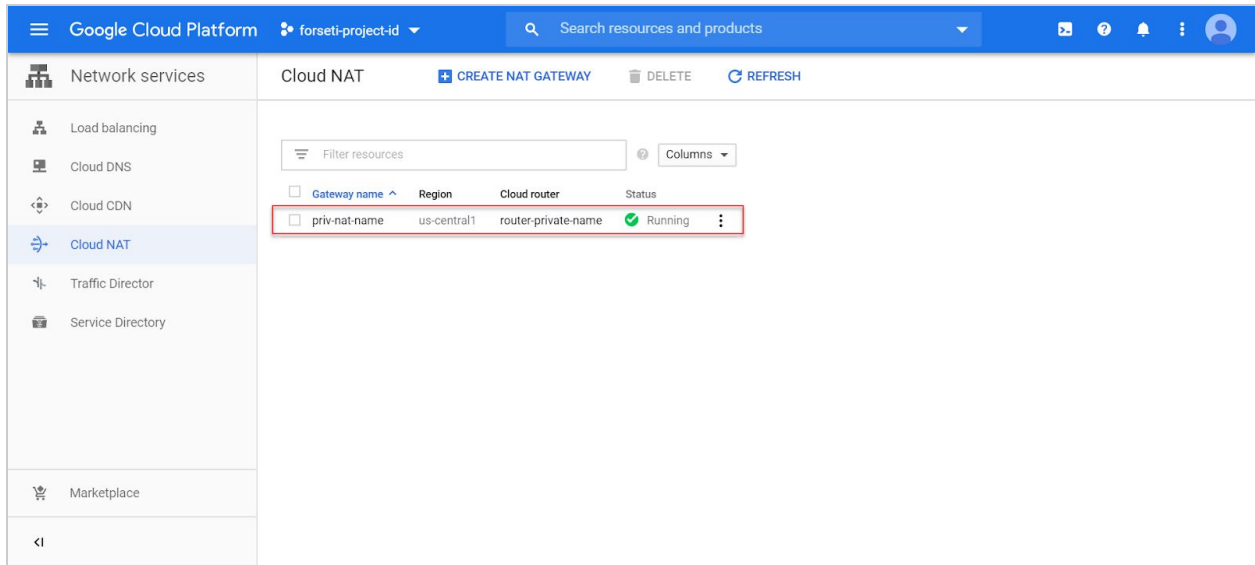


Figure 8 - Network Services Console

Hybrid Connectivity Console: The details of the NAT gateway are available in the [Network Services Console](#) (Figure 9). It provides the details about the subnets selected and the routes configured on the NAT gateway.

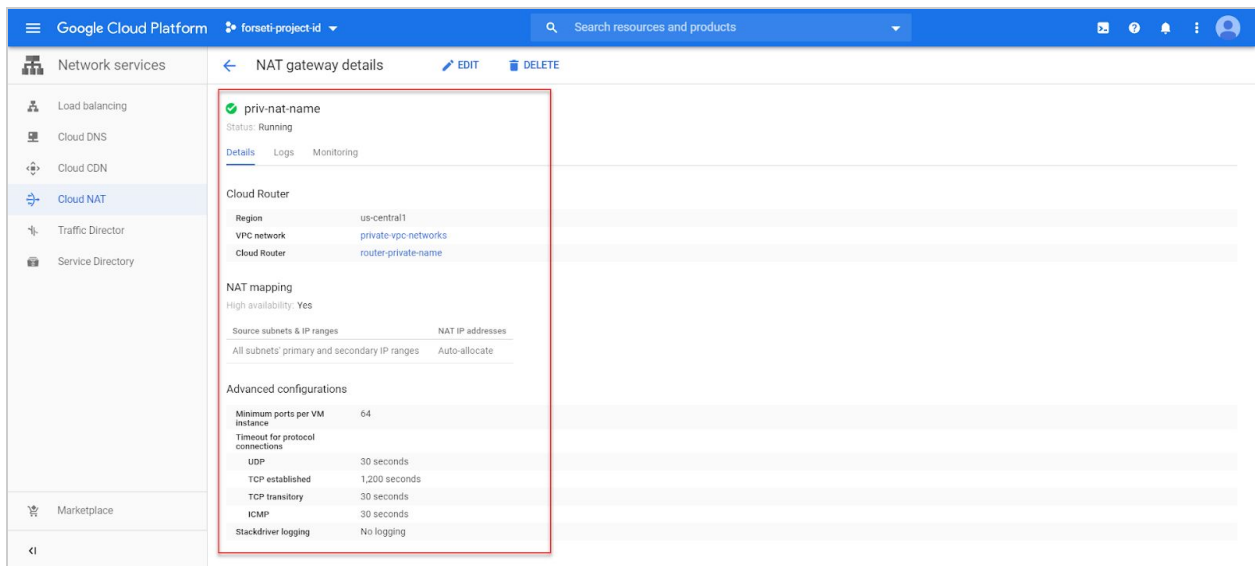


Figure 9 - Hybrid Connectivity Console

BigQuery Console: A BigQuery dataset has been created for analyzing the logs generated by resources, as can be seen in the [BigQuery Console](#) under the Forseti project (Figure 10).

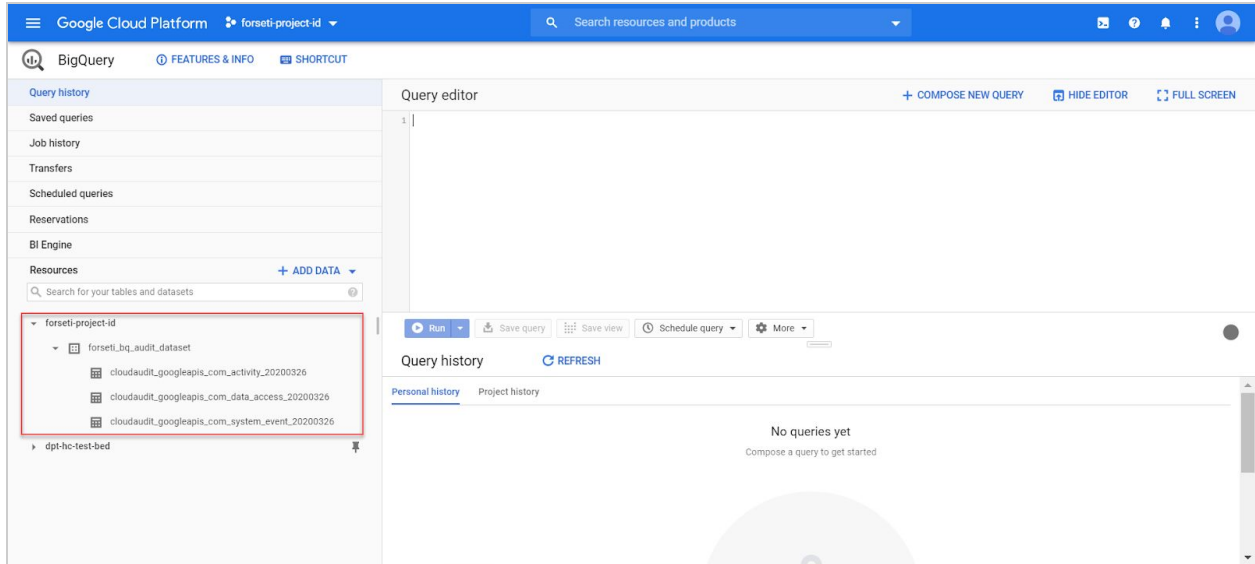


Figure 10 - BigQuery Console

Please note that access to the datasets can be further regulated using Cloud IAM. The IAM permissions provided for the BigQuery dataset are viewable under the *Share Dataset* option (Figure 11). These can be further customized to control access to the BigQuery datasets.

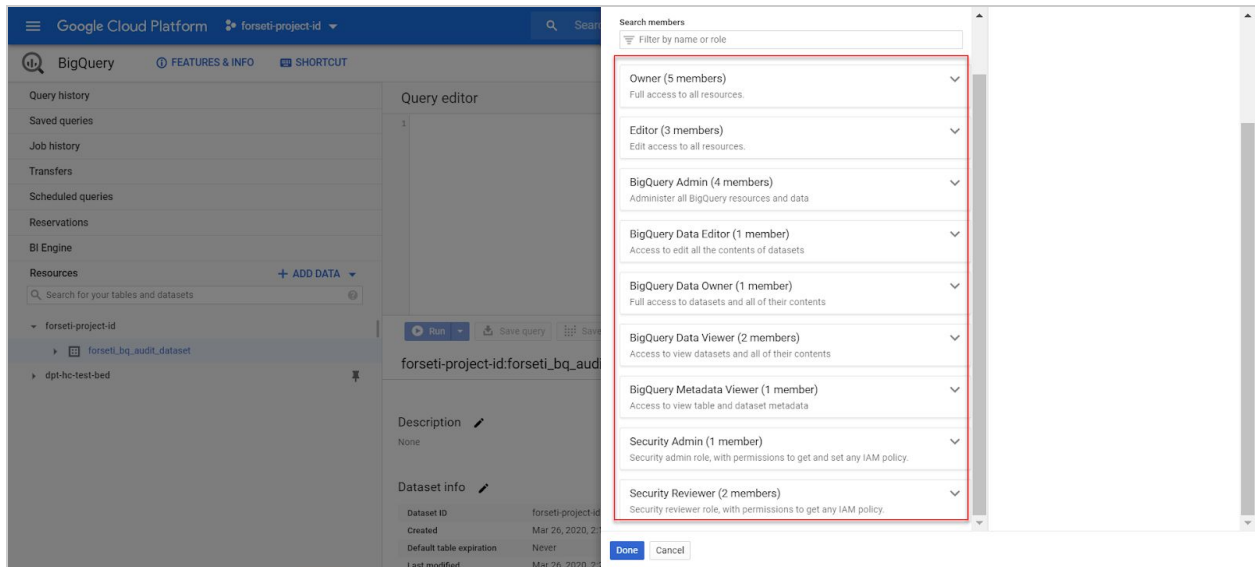


Figure 11 - IAM Permissions for BigQuery Data

Cloud SQL Console: A Forseti SQL Instance has been created, as can be seen in the [Cloud SQL Console](#) screen (Figure 12).

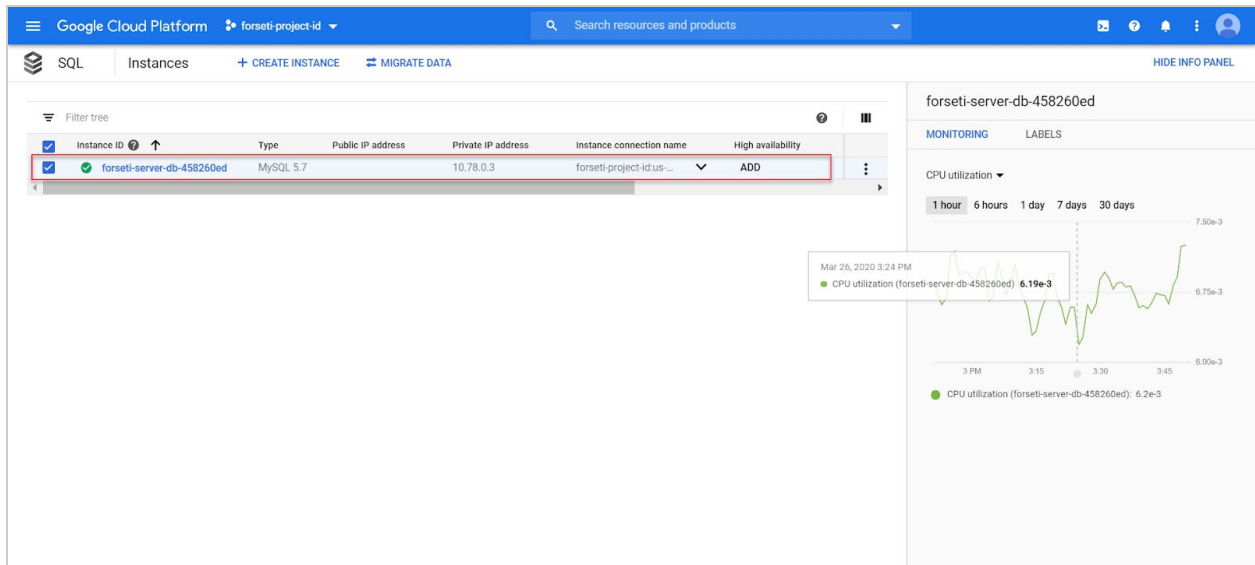


Figure 12 - Cloud SQL Console

Storage Browser Console: In the [Storage Browser Console](#), the storage buckets that are deployed through the template as part of the Forseti project can be seen (Figure 13). The storage buckets and their corresponding permissions can be viewed by selecting the respective buckets in the list.

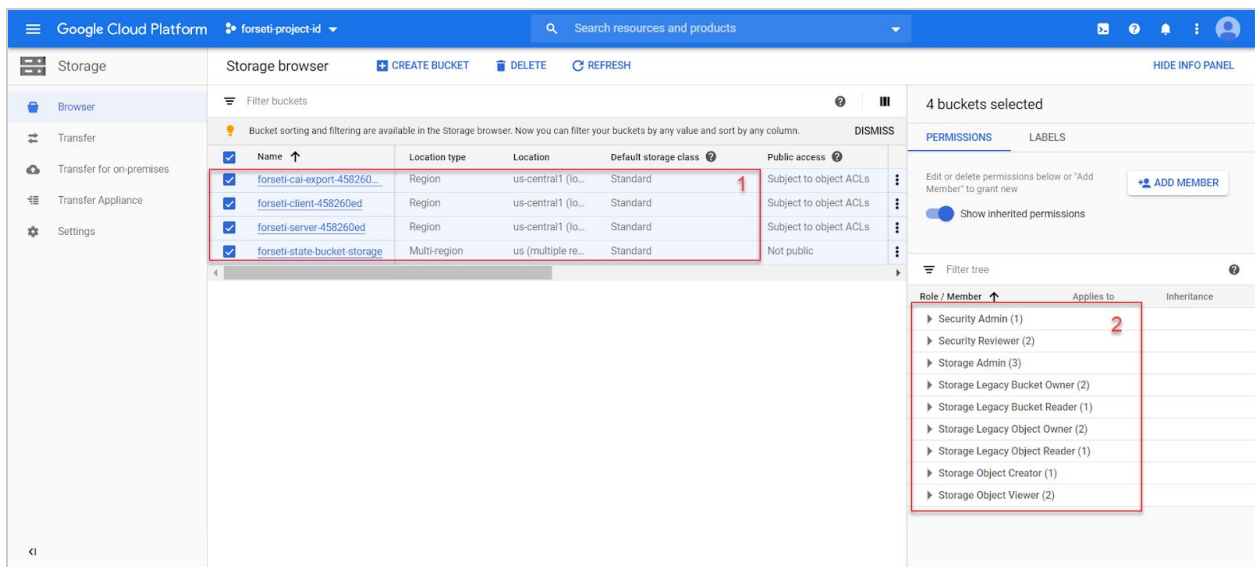


Figure 13 - 1. Forseti Storage Buckets 2. Bucket Permissions

3.8.2 Data-hosting Project (“*rnd-platform-project*”)

BigQuery Cloud Console: As a part of *rnd-platform-project* project, two datasets are mapped - one dataset for storing *audit logs* and the other for storing *non-sensitive data*. The datasets can be seen in the [BigQuery Console](#) (Figure 14).

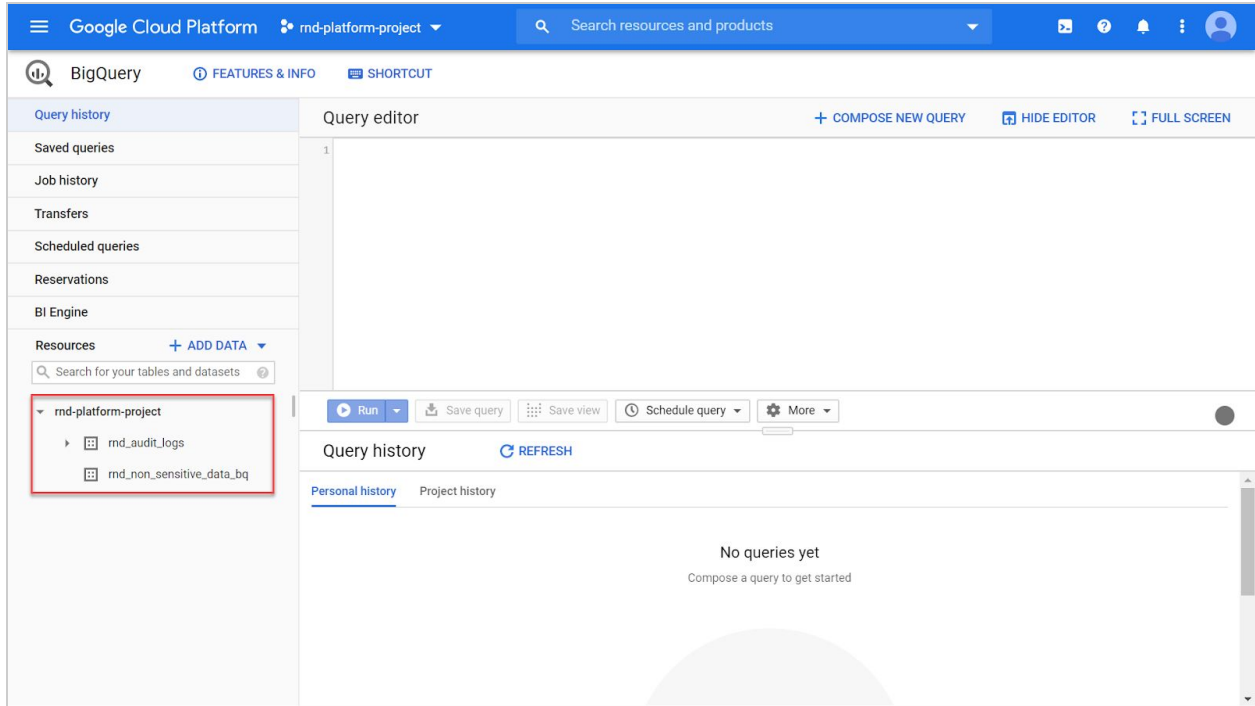


Figure 14 - BigQuery Datasets

Figure 15 and Figure 16 below indicate the permissions granted on the *audit logs BigQuery* dataset and the *non-sensitive data* BigQuery dataset respectively. Since the project *rnd-platform-project* is created inside a folder, some permissions are also inherited from the *Folder level permissions* by the datasets. Under the project name *dpth-ai-ml*, select a dataset and click on *SHARE DATASET* to view the respective permissions.

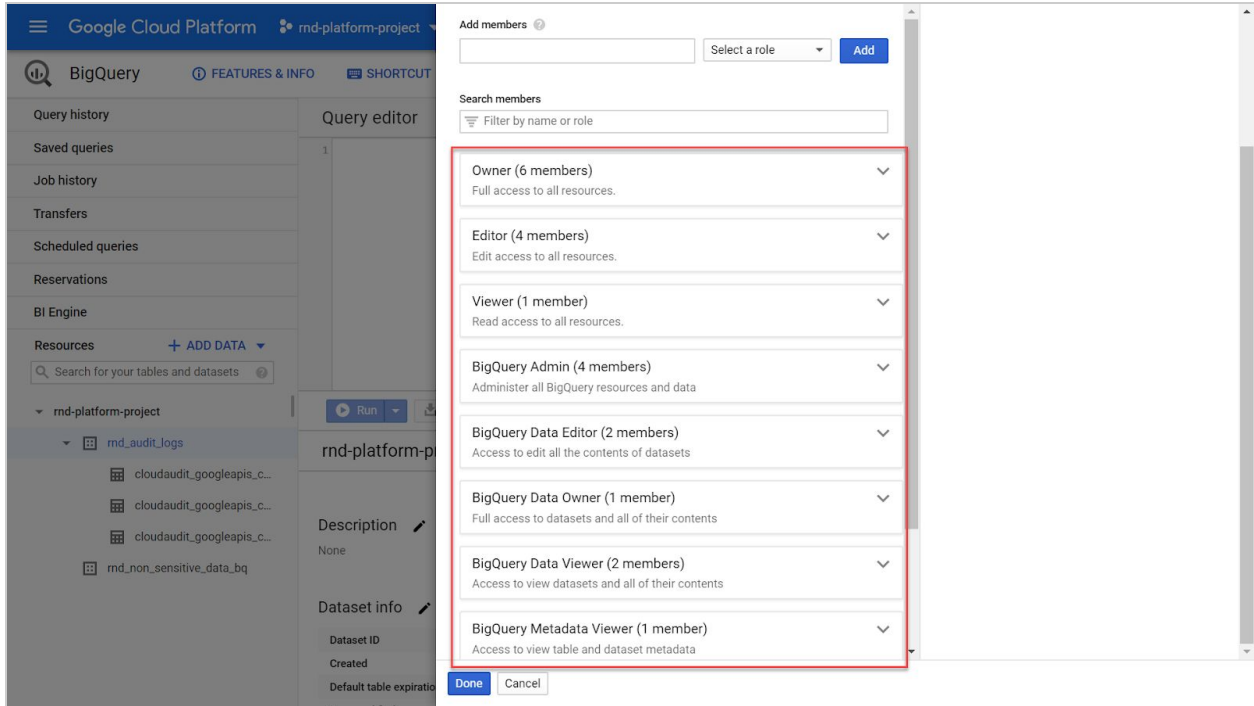


Figure 15 - Audit BigQuery dataset permissions

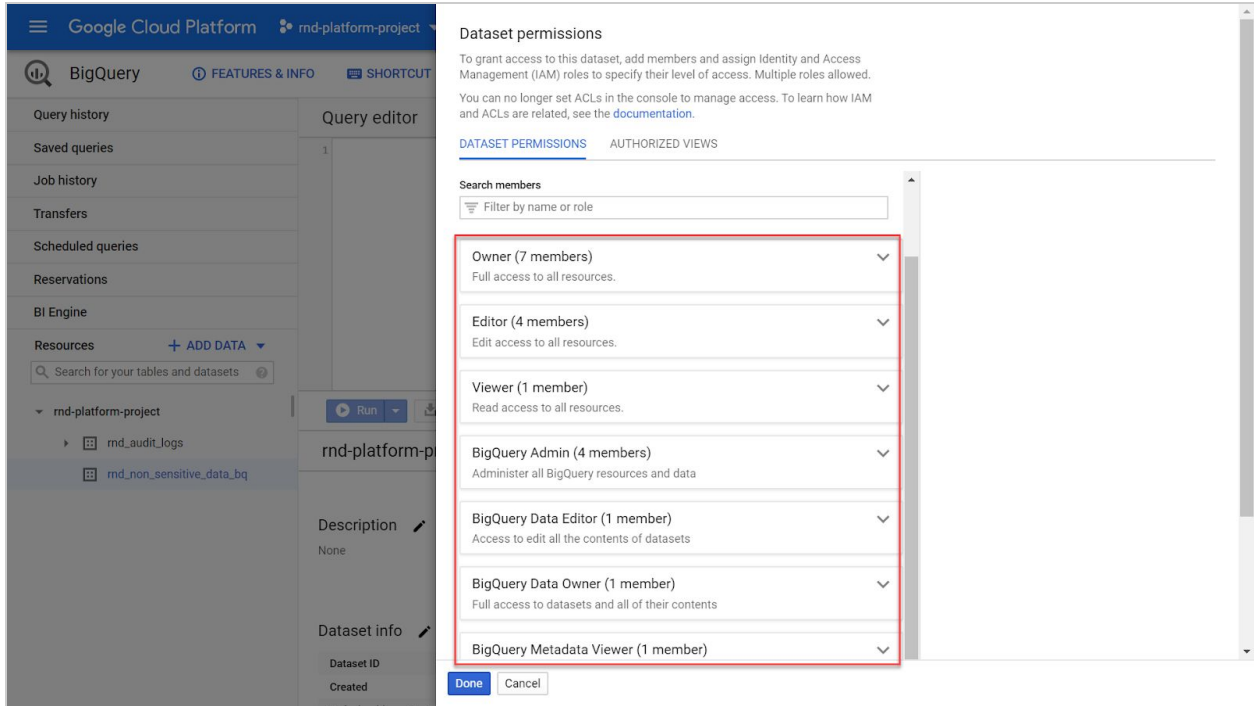


Figure 16 - Non-Sensitive Data dataset permissions

Storage Console: [Storage Browser Console](#) shows five Storage buckets created - One bucket for staging data, two buckets for audit logs and devops state storage and two buckets for sensitive and non-sensitive data. Select a bucket to show the permissions on it as highlighted in Figure 17, Figure 18, Figure 19, Figure 20, and Figure 21 below.

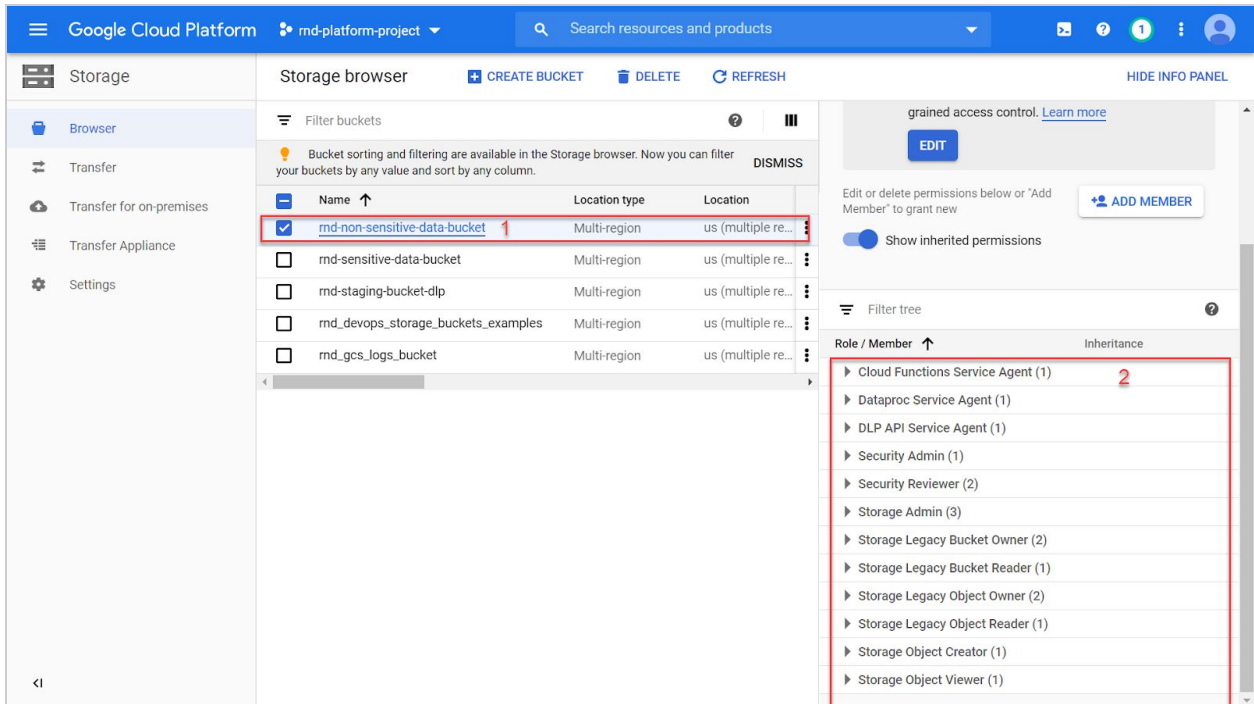


Figure 17 - 1. Non-Sensitive Data Storage Bucket 2. Bucket Permissions

The screenshot shows the Google Cloud Platform interface for the 'md-platform-project'. The 'Storage browser' is active, displaying a list of buckets. The bucket 'md-sensitive-data-bucket' is selected and highlighted with a red box, with a red '1' next to it. To the right, the 'Permissions' panel is open, showing a list of roles and members. A red box highlights the list of roles, with a red '2' next to it. The roles listed are:

- Cloud Functions Service Agent (1)
- Dataproc Service Agent (1)
- DLP API Service Agent (1)
- Security Admin (1)
- Security Reviewer (2)
- Storage Admin (3)
- Storage Legacy Bucket Owner (2)
- Storage Legacy Bucket Reader (1)
- Storage Legacy Object Owner (2)
- Storage Legacy Object Reader (1)
- Storage Object Creator (1)
- Storage Object Viewer (1)

Figure 18 - 1. Sensitive Data Storage Bucket 2. Bucket Permissions

The screenshot shows the Google Cloud Platform interface for the 'md-platform-project'. The 'Storage browser' is active, displaying a list of buckets. The bucket 'md-staging-bucket-dlp' is selected and highlighted with a red box, with a red '1' next to it. To the right, the 'Permissions' panel is open, showing a list of roles and members. A red box highlights the list of roles, with a red '2' next to it. The roles listed are:

- Cloud Functions Service Agent (1)
- Dataproc Service Agent (1)
- DLP API Service Agent (1)
- Security Admin (1)
- Security Reviewer (2)
- Storage Admin (3)
- Storage Legacy Bucket Owner (2)
- Storage Legacy Bucket Reader (1)
- Storage Legacy Object Owner (2)
- Storage Legacy Object Reader (1)
- Storage Object Creator (1)
- Storage Object Viewer (1)

Figure 19 - 1. Staging Storage Bucket 2. Bucket Permissions

The screenshot shows the Google Cloud Platform Storage browser interface. The bucket 'md_devops_storage_buckets_examples' is selected and highlighted with a red box. The permissions panel on the right shows a list of roles and members, with a red box highlighting the first five roles: Cloud Functions Service Agent (1), Dataproc Service Agent (1), DLP API Service Agent (1), Security Admin (1), and Security Reviewer (2). A red '2' is next to the first role.

Name	Location type	Location
<input type="checkbox"/> md-non-sensitive-data-bucket	Multi-region	us (multiple re...)
<input type="checkbox"/> md-sensitive-data-bucket	Multi-region	us (multiple re...)
<input type="checkbox"/> md-staging-bucket-dlp	Multi-region	us (multiple re...)
<input checked="" type="checkbox"/> md_devops_storage_buckets_examples 1	Multi-region	us (multiple re...)
<input type="checkbox"/> md_gcs_logs_bucket	Multi-region	us (multiple re...)

Role / Member	Inheritance
Cloud Functions Service Agent (1)	2
Dataproc Service Agent (1)	
DLP API Service Agent (1)	
Security Admin (1)	
Security Reviewer (2)	
Storage Admin (3)	
Storage Legacy Bucket Owner (2)	
Storage Legacy Bucket Reader (1)	
Storage Legacy Object Owner (2)	
Storage Legacy Object Reader (1)	

Figure 20 - 1. Devops Storage Bucket 2. Bucket Permissions

The screenshot shows the Google Cloud Platform Storage browser interface. The bucket 'md_gcs_logs_bucket' is selected and highlighted with a red box. The permissions panel on the right shows a list of roles and members, with a red box highlighting the first five roles: Cloud Functions Service Agent (1), Dataproc Service Agent (1), DLP API Service Agent (1), Security Admin (1), and Security Reviewer (2). A red '2' is next to the first role.

Name	Location type	Location
<input type="checkbox"/> md-non-sensitive-data-bucket	Multi-region	us (multiple re...)
<input type="checkbox"/> md-sensitive-data-bucket	Multi-region	us (multiple re...)
<input type="checkbox"/> md-staging-bucket-dlp	Multi-region	us (multiple re...)
<input type="checkbox"/> md_devops_storage_buckets_examples	Multi-region	us (multiple re...)
<input checked="" type="checkbox"/> md_gcs_logs_bucket 1	Multi-region	us (multiple re...)

Role / Member	Inheritance
Cloud Functions Service Agent (1)	2
Dataproc Service Agent (1)	
DLP API Service Agent (1)	
Security Admin (1)	
Security Reviewer (2)	
Storage Admin (3)	
Storage Legacy Bucket Owner (2)	
Storage Legacy Bucket Reader (1)	
Storage Legacy Object Owner (2)	
Storage Legacy Object Reader (1)	
Storage Object Creator (1)	
Storage Object Viewer (1)	

Figure 21 - 1. GCS Logs Storage Bucket 2. Bucket Permissions

Healthcare Console: In the [Healthcare Console](#), the Healthcare Dataset created by the DPT template is listed. Permissions on the dataset can be seen by selecting the dataset (Figure 22).

The screenshot displays the Google Cloud Platform interface for the Healthcare console. The top navigation bar includes the Google Cloud logo, the project name 'md-platform-project', and a search bar. The main content area is divided into three sections:

- Left Panel:** A sidebar with 'Healthcare' selected, and sub-options for 'Browser' and 'Marketplace'.
- Center Panel:** Titled 'Datasets', it shows a table of datasets. A red box highlights the dataset 'md_project_healthcare-datasets' in the 'us-central1' region, with a count of '1'.
- Right Panel:** Titled 'Permissions', it shows options to 'ADD MEMBER' and 'Show inherited permissions'. Below this is a 'Filter tree' section with a table of roles and members. A red box highlights the following roles and their counts: Editor (4), Healthcare Dataset Viewer (1), Owner (6), Security Admin (1), Security Reviewer (2), and Viewer (1). The total count for these roles is '2'.

Name	Region	Count
md_project_healthcare-datasets	us-central1	1

Role / Member	Inheritance
Editor (4)	2
Healthcare Dataset Viewer (1)	
Owner (6)	
Security Admin (1)	
Security Reviewer (2)	
Viewer (1)	

Figure 22 - 1. Healthcare Dataset 2. Dataset Permissions

Select the dataset to see the *DICOM*, *FHIR* and *HL7-V2* datastores. Figure 23, Figure 24 and Figure 25 show the permissions for *DICOM*, *FHIR* and *HL7-V2* respectively.

The screenshot shows the Google Cloud Platform interface for a Healthcare dataset. The left sidebar contains 'Healthcare', 'Browser', and 'Marketplace'. The main area is titled 'Dataset' and shows a list of 'Data stores' under the 'DATA STORES' tab. The 'dicom-store' is selected and highlighted with a red box. The 'Permissions' panel on the right shows a list of roles and members, with 'Editor (4)' and 'Healthcare DICOM Editor (1)' highlighted with a red box.

Name	Type	Version	Cloud Pub/Sub Topic
<input checked="" type="checkbox"/> dicom-store	DICOM	1	-
<input type="checkbox"/> fhir-store	FHIR	STU3	-
<input type="checkbox"/> hl7-v2-store	HL7V2	-	-

Role / Member	Inheritance
▶ Editor (4)	2
▶ Healthcare DICOM Editor (1)	
▶ Healthcare DICOM Store Administrator (1)	
▶ Owner (6)	
▶ Security Admin (1)	
▶ Security Reviewer (2)	
▶ Viewer (1)	

Figure 23 - 1. DICOM Datastore 2. Datastore Permissions

The screenshot shows the Google Cloud Platform interface for a Healthcare dataset. The left sidebar contains 'Healthcare', 'Browser', and 'Marketplace'. The main area is titled 'Dataset' and shows a list of 'Data stores' under the 'DATA STORES' tab. The 'fhir-store' is selected and highlighted with a red box. The 'Permissions' panel on the right shows a list of roles and members, with 'Editor (4)' and 'Healthcare FHIR Resource Editor (1)' highlighted with a red box.

Name	Type	Version	Cloud Pub/Sub Topic
<input type="checkbox"/> dicom-store	DICOM	-	-
<input checked="" type="checkbox"/> fhir-store	FHIR	STU3	1
<input type="checkbox"/> hl7-v2-store	HL7V2	-	-

Role / Member	Inheritance
▶ Editor (4)	2
▶ Healthcare FHIR Resource Editor (1)	
▶ Healthcare FHIR Resource Reader (1)	
▶ Owner (6)	
▶ Security Admin (1)	
▶ Security Reviewer (2)	
▶ Viewer (1)	

Figure 24 - 1. FHIR Datastore 2. Datastore Permissions

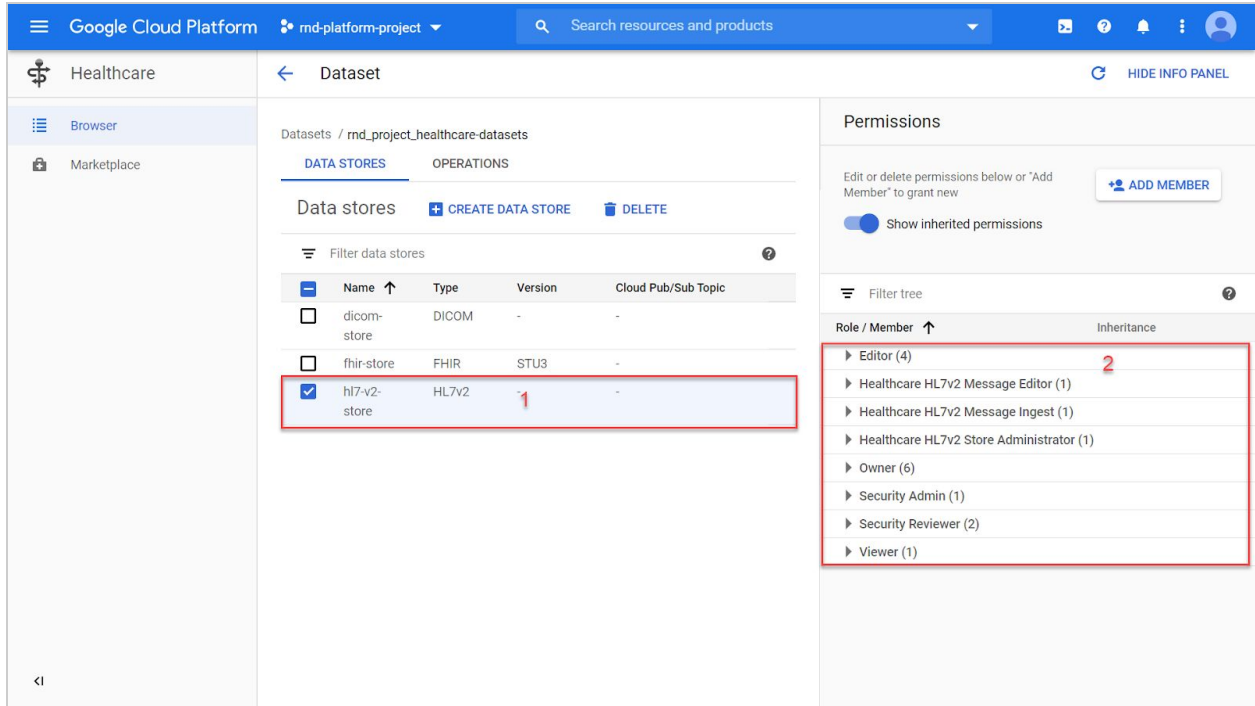


Figure 25 - 1. HL7-v2 Datastore 2. Datastore Permissions

Google Kubernetes Engine Console: On the [Kubernetes Engine Console](#), two Kubernetes clusters (GKE clusters) created by the DPT are listed, one for the analysts and the other for researchers (Figure 26).

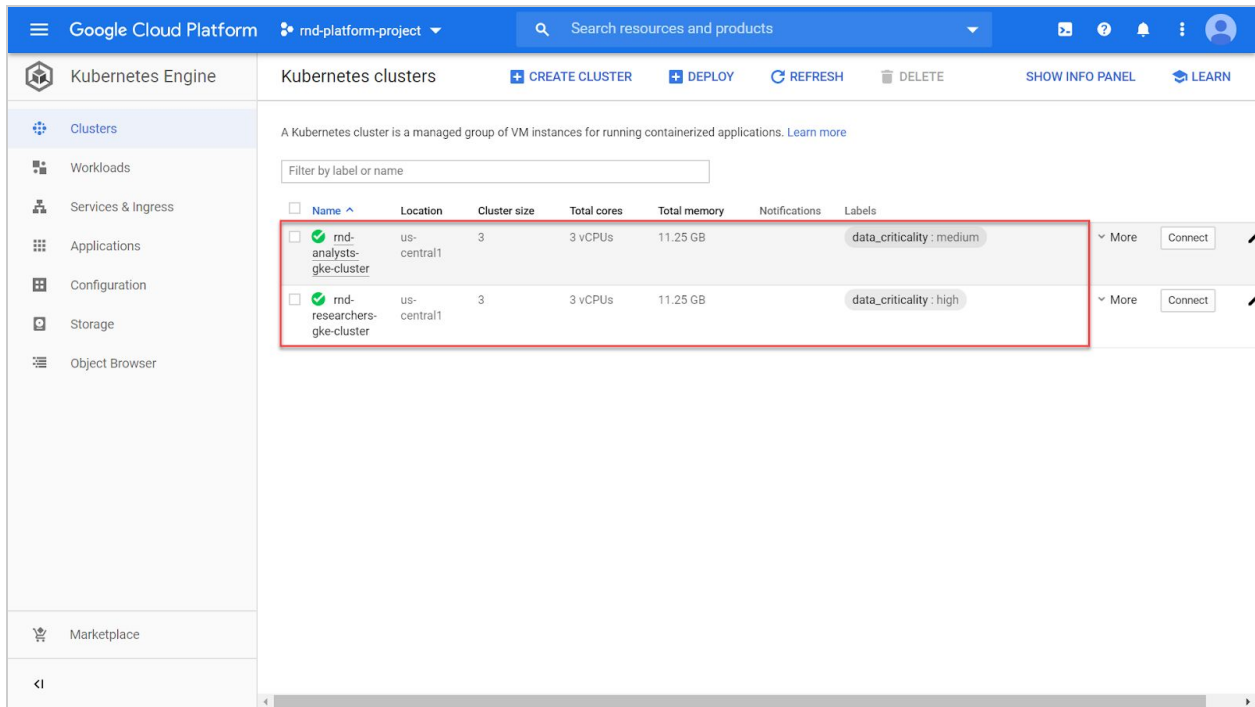


Figure 26 - Google Kubernetes Engine Console with clusters

Figure 27 lists all the configurations of the analysts GKE cluster and Figure 28 lists the nodes created for the cluster.

The screenshot shows the 'rnd-analysts-gke-cluster' configuration page in the Google Cloud Platform console. The 'Details' tab is selected, and a red box highlights the configuration table.

Property	Value	Action
Master version	1.14.10-gke.27	Upgrade available
Endpoint	172.16.1.34	Show credentials
Client certificate	Disabled	
Binary Authorization	Enabled	Configure
Kubernetes alpha features	Disabled	
Current total size	3	
Region	us-central1	
Node zones	us-central1-b us-central1-c us-central1-f	
Network	rnd-private-vpc-network	
Subnet	analysts-gke-cluster-subnet	
VPC-native (alias IP)	Enabled	
Pod address range	10.4.0.0/14	
Default maximum pods per node	8	
Service address range	10.0.32.0/20	
Intranode visibility	Disabled	
Kubernetes Engine Monitoring	System and workload logging and monitoring	
Private cluster	Enabled	
VPC peering	gke-n3aab5931dd3bac4052c-6ef5-81a6-peer	
Master address range	172.16.1.32/28	

Figure 27 - Analysts GKE Cluster Configurations

The screenshot shows the 'Nodes' tab for the 'rnd-analysts-gke-cluster'. A red box highlights a table listing the nodes.

Name	Status	CPU requested	CPU allocatable	Memory requested	Memory allocatable	Storage requested	Storage allocatable
gke-rnd-analysts-gke-analysts-gke-clu-04ba9d2e-9ncm	Ready	471 mCPU	940 mCPU	367 MB	2.77 GB	0 B	0 B
gke-rnd-analysts-gke-analysts-gke-clu-6efd9edb-360v	Ready	294 mCPU	940 mCPU	352.09 MB	2.77 GB	0 B	0 B
gke-rnd-analysts-gke-analysts-gke-clu-d11304d0-2fg0	Ready	592 mCPU	940 mCPU	687.64 MB	2.77 GB	0 B	0 B

Figure 28 - Analysts GKE Cluster Nodes

Figure 29 lists all the configurations of the researchers GKE cluster and Figure 30 lists the nodes created for the cluster.

The screenshot shows the Google Cloud Platform interface for the 'rnd-researchers-gke-cluster'. The 'Details' tab is selected, displaying the following configuration parameters:

Parameter	Value	Action
Master version	1.14.10-gke.27	Upgrade available
Endpoint	172.16.0.34	Show credentials
Client certificate	Disabled	
Binary Authorization	Enabled	Configure
Kubernetes alpha features	Disabled	
Current total size	3	
Region	us-central1	
Node zones	us-central1-b us-central1-f us-central1-a	
Network	rnd-private-vpc-network	
Subnet	researchers-gke-cluster-subnet	
VPC-native (alias IP)	Enabled	
Pod address range	10.8.0.0/14	
Default maximum pods per node	8	
Service address range	10.1.32.0/20	
Intranode visibility	Disabled	
Kubernetes Engine Monitoring	System and workload logging and monitoring	
Private cluster	Enabled	
VPC peering	gke-n3aab5931dd3bac4052c-6ef5-81a6-peer	
Master address range	172.16.0.32/28	

Figure 29 - Researchers GKE Cluster Configurations

The screenshot shows the Google Cloud Platform interface for the 'rnd-researchers-gke-cluster', specifically the 'Nodes' tab. A table lists the nodes with their status and resource requirements:

Name	Status	CPU requested	CPU allocatable	Memory requested	Memory allocatable	Storage requested	Storage allocatable
gke-rnd-researchers-researchers-gke-82f75832-1cmr	Ready	415 mCPU	940 mCPU	704.18 MB	2.77 GB	0 B	0 B
gke-rnd-researchers-researchers-gke-9e54ad94-9211	Ready	481 mCPU	940 mCPU	356.52 MB	2.77 GB	0 B	0 B
gke-rnd-researchers-researchers-gke-a28d2107-0mnt	Ready	461 mCPU	940 mCPU	346.03 MB	2.77 GB	0 B	0 B

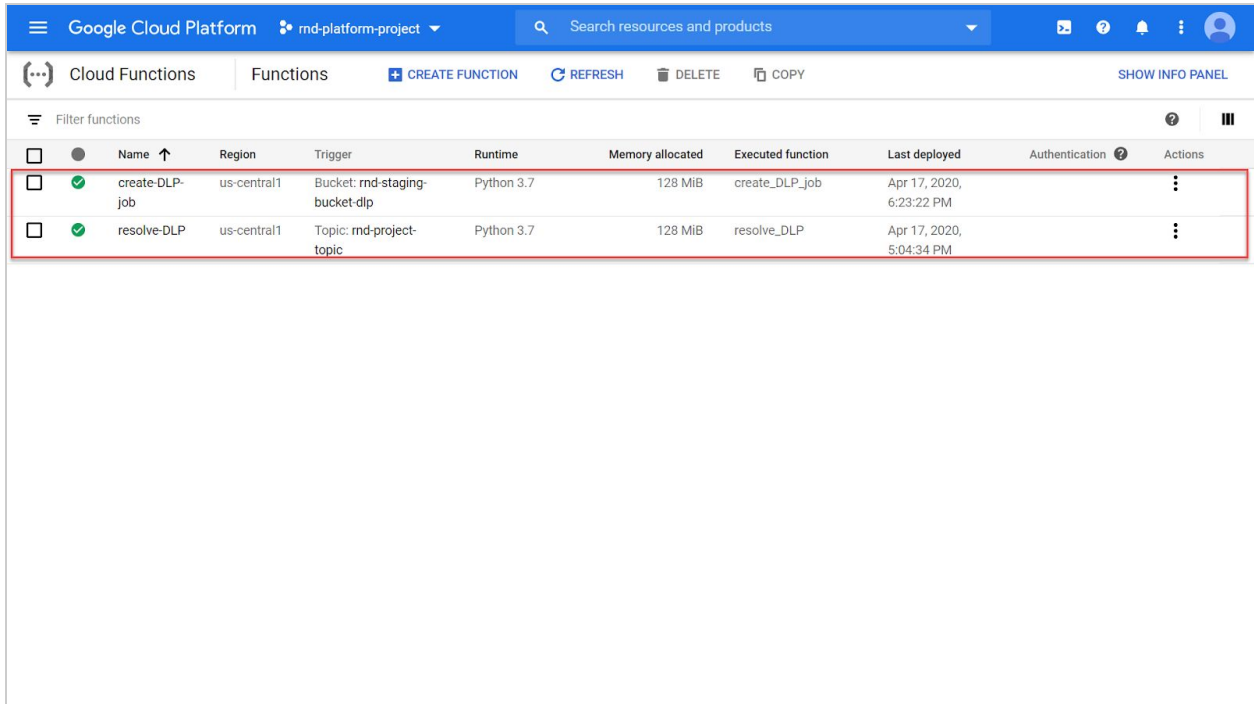
Figure 30 - Researchers GKE Cluster Nodes

Compute Engine: On the [Compute Engine Console](#), all the GKE clusters' nodes from Figure 28 and Figure 30 are listed. These nodes are compute VM instances (Figure 31).

Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input type="checkbox"/> gke-rnd-analysts-gke-analysts-gke-clu-04ba9d2e-9ncm	us-central1-b		gke-rnd-analysts-gke-analysts-gke-clu-04ba9d2e-grp	192.168.0.10 (nic0)	None	SSH
<input type="checkbox"/> gke-rnd-analysts-gke-analysts-gke-clu-6efd9edb-360v	us-central1-c		gke-rnd-analysts-gke-analysts-gke-clu-6efd9edb-grp	192.168.0.9 (nic0)	None	SSH
<input type="checkbox"/> gke-rnd-analysts-gke-analysts-gke-clu-d11304d0-2fg0	us-central1-f		gke-rnd-analysts-gke-analysts-gke-clu-d11304d0-grp	192.168.0.5 (nic0)	None	SSH
<input type="checkbox"/> gke-rnd-researchers-researchers-gke-82175832-tcmr	us-central1-f		gke-rnd-researchers-researchers-gke-82175832-grp	10.0.0.6 (nic0)	None	SSH
<input type="checkbox"/> gke-rnd-researchers-researchers-gke-9e54ad94-9211	us-central1-b		gke-rnd-researchers-researchers-gke-9e54ad94-grp	10.0.0.10 (nic0)	None	SSH
<input type="checkbox"/> gke-rnd-researchers-researchers-gke-a28d2107-0mnt	us-central1-a		gke-rnd-researchers-researchers-gke-a28d2107-grp	10.0.0.7 (nic0)	None	SSH

Figure 31 - GKE Clusters' Nodes

Cloud Functions Console: On the [Cloud Functions Console](#), two Cloud Functions - one performing the DLP classification task and the other performing sorting of staging storage bucket objects into sensitive and non-sensitive data storage buckets - are listed (Figure 32).



The screenshot shows the Google Cloud Platform interface for the 'Cloud Functions' console. The page title is 'Google Cloud Platform' and the project is 'md-platform-project'. The search bar contains 'Search resources and products'. The main heading is 'Cloud Functions' with a sub-heading 'Functions'. There are buttons for 'CREATE FUNCTION', 'REFRESH', 'DELETE', and 'COPY'. A 'SHOW INFO PANEL' link is also present. Below the heading is a 'Filter functions' section. The main content is a table with the following columns: Name, Region, Trigger, Runtime, Memory allocated, Executed function, Last deployed, Authentication, and Actions. Two functions are listed and highlighted with a red border:

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Name ↑	Region	Trigger	Runtime	Memory allocated	Executed function	Last deployed	Authentication	Actions
<input type="checkbox"/>	<input checked="" type="checkbox"/>	create-DLP-job	us-central1	Bucket: rmd-staging-bucket-dlp	Python 3.7	128 MiB	create_DLP_job	Apr 17, 2020, 6:23:22 PM		⋮
<input type="checkbox"/>	<input checked="" type="checkbox"/>	resolve-DLP	us-central1	Topic: rmd-project-topic	Python 3.7	128 MiB	resolve_DLP	Apr 17, 2020, 5:04:34 PM		⋮

Figure 32 - Cloud Functions Console

Figure 33 and Figure 34 point out the triggers for both the Cloud Functions - staging storage bucket for DLP classification function and a Pub/Sub topic for the object classification function.

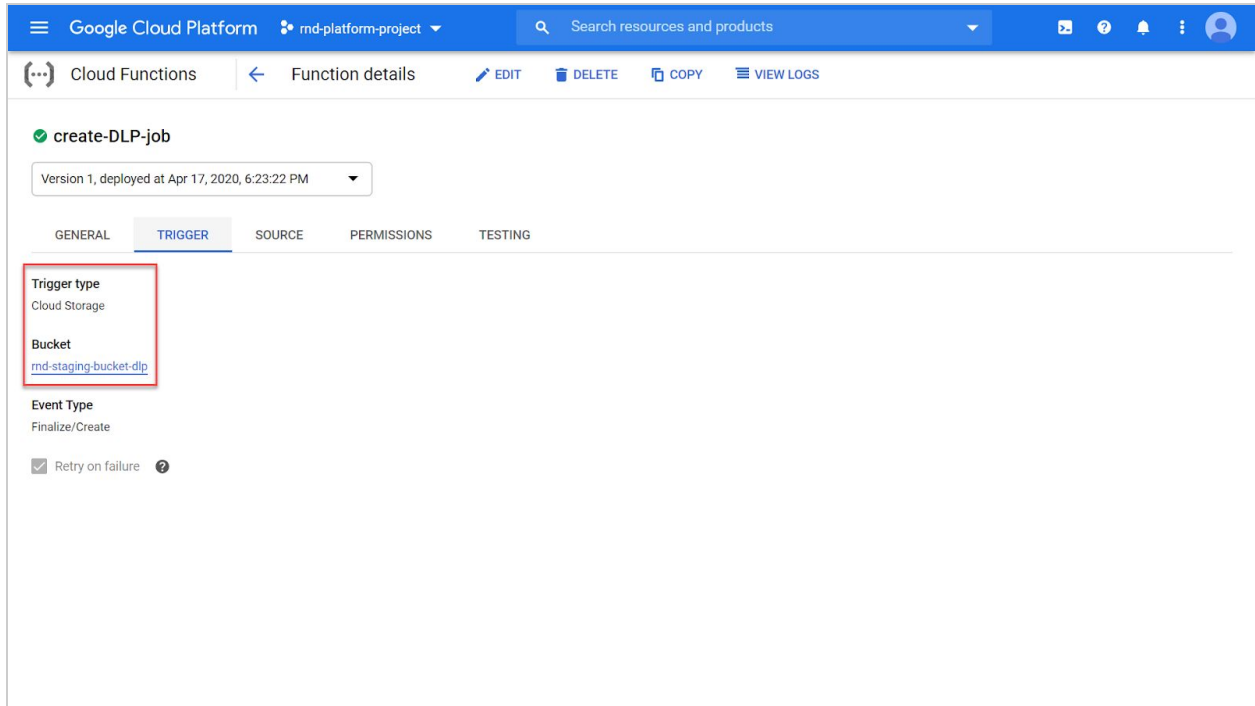


Figure 33 - Trigger for DLP classification Cloud Function

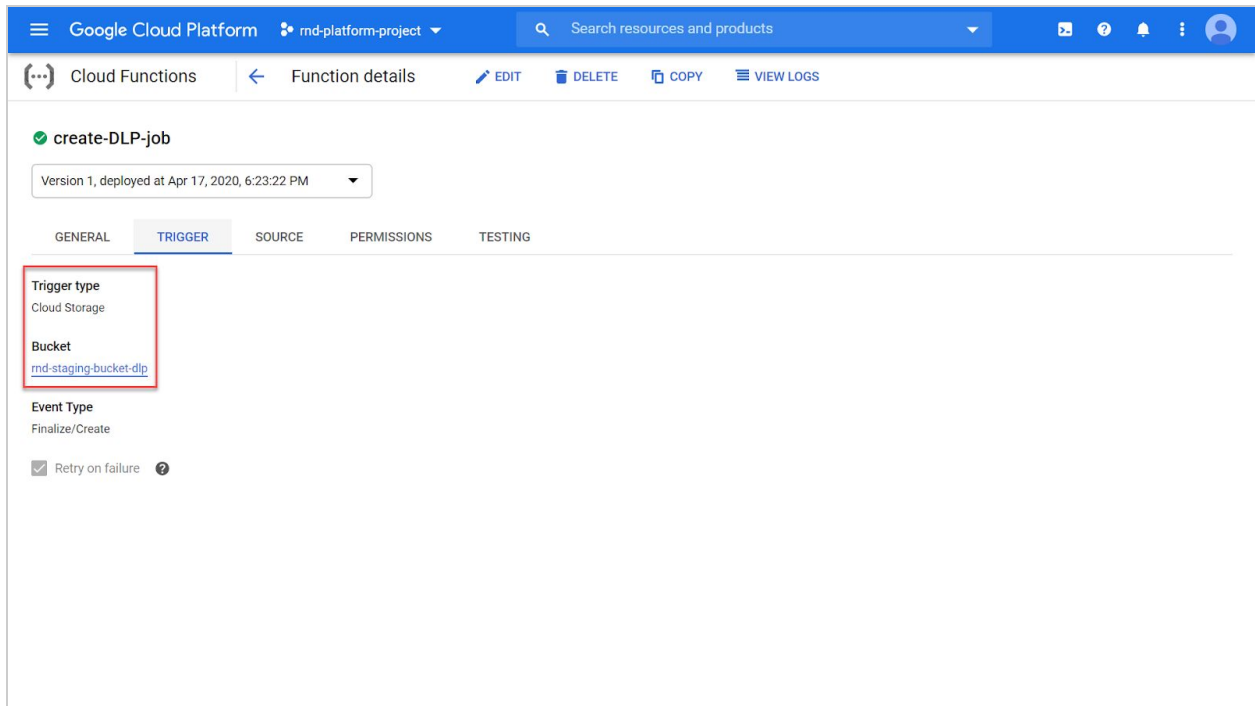


Figure 34 - Trigger for object sorting Cloud Function

Pub/Sub Console: On the [Pub/Sub Console](#), two Pub/Sub topics - one for fetching labels from the DLP classification task, and the other one which is a dead-letter topic for the previous topic - are listed (Figure 35).

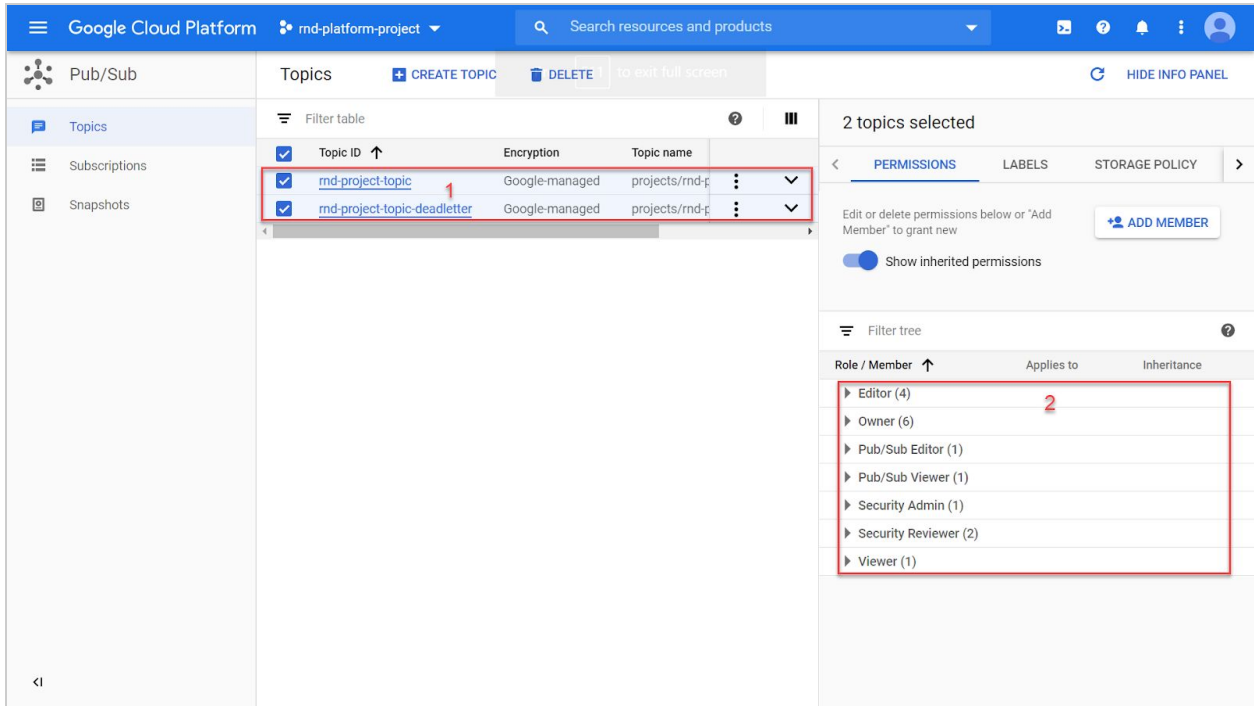


Figure 35 - 1. Pub/Sub Topics 2. Permissions for the topics

The subscriptions attached to the topics listed in Figure 35 are listed in Figure 36 along with the permissions.

The screenshot displays the Google Cloud Platform interface for Pub/Sub Subscriptions. The main table lists three subscriptions, with the first row highlighted by a red box and labeled '1'. The permissions panel on the right shows a list of roles and member counts, with the 'Owner (6)' role highlighted by a red box and labeled '2'.

Subscription ID	Delivery type	Topic name
gcf-resolve-DLP-us-central1-rnd-project-topic	Push	projects/rnd-plat...
md-project-subscription	Pull	projects/rnd-plat...
md-project-subscription-deadletter	Pull	projects/rnd-plat...

Role / Member	Applies to	Inheritance
Editor (4)		
Owner (6)		
Pub/Sub Editor (1)		
Pub/Sub Subscriber (1)		
Security Admin (1)		
Security Reviewer (2)		
Viewer (1)		

Figure 36 - 1. Pub/Sub Subscriptions 2. Permissions for the subscriptions

Virtual Private Cloud Networks: On the [VPC network Console](#), the custom created VPC is listed with the corresponding subnetwork. Figure 37 shows the custom created *rnd-private-vpc-network* network and its subnetworks created for both the GKE clusters.

VPC networks						
europa-west3	default		10.156.0.0/20	10.156.0.1		Off
southamerica-east1	default		10.158.0.0/20	10.158.0.1		Off
asia-south1	default		10.160.0.0/20	10.160.0.1		Off
northamerica-northeast1	default		10.162.0.0/20	10.162.0.1		Off
europa-west4	default		10.164.0.0/20	10.164.0.1		Off
europa-north1	default		10.166.0.0/20	10.166.0.1		Off
us-west2	default		10.168.0.0/20	10.168.0.1		Off
asia-east2	default		10.170.0.0/20	10.170.0.1		Off
europa-west6	default		10.172.0.0/20	10.172.0.1		Off
asia-northeast2	default		10.174.0.0/20	10.174.0.1		Off
asia-northeast3	default		10.178.0.0/20	10.178.0.1		Off
us-west3	default		10.180.0.0/20	10.180.0.1		Off
us-west4	default		10.182.0.0/20	10.182.0.1		Off
rnd-private-vpc-network	2	Custom			6	Off
us-central1	analysts-gke-cluster-subnet		192.168.0.0/20, 2 more	192.168.0.1		Off
us-central1	researchers-gke-cluster-subnet		10.0.0.0/20, 2 more	10.0.0.1		Off

Figure 37 - VPC Network

For more details, select the custom created VPC. Figure 38 and Figure 39 show the details page with permissions highlighted for subnetworks of the VPC network.

The screenshot shows the Google Cloud Platform interface for a VPC network named 'rnd-private-vpc-network'. The left sidebar lists various VPC-related services. The main content area shows the details for the 'analysts-gke-cluster-subnet'. A table lists subnets, with 'analysts-gke-cluster-subnet' selected and highlighted with a red box. A red '1' is placed next to this row. To the right, the 'PERMISSIONS' section is also highlighted with a red box, showing a list of roles and their member counts: Owner (6 members) with a red '2', Editor (4 members), Viewer (1 member), Compute Network Viewer (1 member), Dataproc Service Agent (1 member), Security Admin (1 member), and Security Reviewer (2 members).

Name	Region	IP address ranges	Gateway	Private Google access
<input checked="" type="checkbox"/> analysts-gke-cluster-subnet	us-central1	192.168.0.0/20, 2 more	192.168.0.1	On
<input type="checkbox"/> researchers-gke-cluster-subnet	us-central1	10.0.0.0/20, 2 more	10.0.0.1	On

Figure 38 - 1. Analysts GKE Cluster Subnetwork 2. Permissions

This screenshot is similar to Figure 38, but the 'researchers-gke-cluster-subnet' is selected and highlighted with a red box. A red '1' is placed next to this row. The 'PERMISSIONS' section on the right is also highlighted with a red box, showing the same list of roles and member counts as in Figure 38.

Name	Region	IP address ranges	Gateway	Private Google access
<input type="checkbox"/> analysts-gke-cluster-subnet	us-central1	192.168.0.0/20, 2 more	192.168.0.1	On
<input checked="" type="checkbox"/> researchers-gke-cluster-subnet	us-central1	10.0.0.0/20, 2 more	10.0.0.1	On

Figure 39 - 1. Researchers GKE Cluster Subnetwork 2. Permissions

Service Accounts: On the [Service Accounts Console](#), default service accounts and four custom service accounts for two GKE clusters and two BigQuery datasets are listed (Figure 40).

The screenshot shows the Google Cloud Platform interface for the project 'rnd-platform-project'. The left sidebar contains navigation options like IAM, Identity & Organization, and Service Accounts. The main content area displays 'Service accounts for project "rnd-platform-project"'. A table lists several service accounts, with five highlighted by a red border. The 'Permissions' panel on the right provides information on how to manage service account access.

Email	Status	Actions
<input type="checkbox"/> researchers-gke-cluster-sa@rnd-platform-project.iam.gserviceaccount.com	✓	⋮
<input type="checkbox"/> rnd-non-sensitive-data-bq-sa@rnd-platform-project.iam.gserviceaccount.com	✓	⋮
<input type="checkbox"/> analysts-gke-cluster-sa@rnd-platform-project.iam.gserviceaccount.com	✓	⋮
<input type="checkbox"/> audit-bq-sa@rnd-platform-project.iam.gserviceaccount.com	✓	⋮
<input type="checkbox"/> rnd-platform-project@appspot.gserviceaccount.com	✓	⋮
<input type="checkbox"/> 43120061059-compute@developer.gserviceaccount.com	✓	⋮

Figure 40 - Service Accounts

API & Services Console: Figure 41 below lists all the APIs enabled as a part of the data hosting project.

Name	Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
Compute Engine API	24,537	8.681	198.568	293.784
Cloud Monitoring API	15,681	0	32.006	77.636
Cloud Logging API	4,046	0.321	93.006	140.045
Cloud Healthcare API	93	0	90.112	241.712
Cloud Functions API	55	0	307.515	808.277
Service Networking API	2	0	393.216	511.181
BigQuery API				
BigQuery Storage API				
Cloud Data Loss Prevention (DLP) API				
Cloud Dataproc API				
Cloud Dataproc Control API PRIVATE ?				
Cloud Life Sciences API				
Cloud Natural Language API				
Cloud OS Login API				
Cloud Pub/Sub API				
Cloud Resource Manager API				
Cloud Speech-to-Text API				
Cloud Storage				
Cloud Translation API				
Cloud Video Intelligence API				

Figure 41 - API & Services Console

4. Extended Product Guidance

The section outlines the GxP-aligned security configurations for other products and services that are not part of the Life Sciences R&D platform deployment, but which can be included based on customer specific requirements. The instructions for deploying an example project using DPT is explained in [Section 3.5](#).

For the following products and services, Section 4 outlines the default out-of-the-box configurations, including user-configurable settings, that are available.

- Google Cloud Spanner
- Google Cloud Bigtable
- Google Cloud Firewall
- Google Cloud SQL
- Deep Learning Virtual Machines
- Google Cloud Dataproc
- Google Cloud Datalab

Note:

1. *Google Cloud products and services not included in the list above, but which are supported by Terraform can be integrated by adding the respective script(s) within the terraform_deployments section in the Life Sciences R&D Platform template shown in Section 3. The list of Google Cloud products and services supported by Terraform can be found [here](#).*
2. *Requirements for the creation of customized groups with specific access-level permissions using Cloud IAM and conditional access varies across organizations and can be customized in DPT.*

4.1 Google Cloud Spanner

Google Cloud Spanner is a fully managed, horizontally scalable, and highly available relational database solution offering consistency, scalability, and availability at the Cloud level.

Cloud Spanner can handle high volume real-time data at a very low cost. This further enhances its usability in real-time decision making, along with a very low latency. Cloud Spanner offers global consistency alongside SQL in applications requiring seamless integration across multiple regions.

To learn more about Cloud Spanner and the parameters discussed below, refer to the [Cloud Spanner documentation](#) and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Cloud Spanner

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Identity and Access Management	Users in the owners' group of the project are allowed to perform any action allowed by the project's organization policies.	Use G Suite to create users and groups. Additional role-member bindings can be added in the data section of the template as suggested to control access to the spanner instance.

Copy of Records	Default Configurations	User-Controlled Configurations (ex. via DPT)
Infrastructure Security	<p>Cloud Spanner is replicated by default at the byte level from the underlying distributed file system that it's built on.</p> <p>Besides that, Cloud Spanner uses a concept called splits which replicates the data in the instance and provides additional compute power for each copy of replication. This ensures high availability of data and also the service in the event of a physical or technical incident.</p>	<p>A node provides up to 2TB of storage and the template here is configured to use two nodes per instance which can be scaled</p> <p>The number of nodes can be configured in the template based on the requirement.</p>

DPT Template Configuration for Google Cloud Spanner

```

Template (Please refer to accompanying *.yaml template for detailed configuration)
terraform_deployments:
  resources:
    config:
      resource:
        - google_spanner_instance:
            - main:
                # Code Block 4.1.a
                - config: regional-europe-west1
                  display_name: main-instance
                  num_nodes: '2'
        - google_spanner_database:
            - database:
                - ddl:

```

<pre> - CREATE TABLE t1 (t1 INT64 NOT NULL,) PRIMARY KEY(t1) - CREATE TABLE t2 (t2 INT64 NOT NULL,) PRIMARY KEY(t2) instance: "\${google_spanner_instance.main.name}" name: my-database # the bindings and the corresponding roles which the IAM policy refers to are in the data section, and instance, database, and policy are in the resource section. # add roles in the data section under binding # Code Block 4.1.b - google_spanner_instance_iam_member: - owner: - instance: "\${google_spanner_instance.main.name}" role: roles/owner member: user:user@domain reviewer: - instance: "\${google_spanner_instance.main.name}" role: roles/iam.securityReviewer member: user:user@domain </pre>		
Infrastructure Security	Asset Management	<p>Refer to code block 4.1.a</p> <p>config - The name of the instance's configuration (similar but not quite the same as a region), which defines the geographic placement and replication of databases in this instance. It determines where the data is stored.</p>
Identity and Access Management	User access control	<p>Refer to code block 4.1.b</p> <p>google_spanner_instance_iam_.binding - Member role for the user. G Suite users/groups and Cloud IAM roles can be used to control access.</p>

4.2 Google Cloud Bigtable

Google Cloud Bigtable is Google Cloud's NoSQL database service. Bigtable can quickly scale to support billions of rows and columns and can store petabytes of data. It supports high read and write throughput at low latency, and it is an ideal data source for MapReduce operations.

Google Cloud encrypts all data stored in the Cloud Bigtable by default, but row-level or cell-level security restrictions are not offered by Cloud Bigtable.

To learn more about Cloud Bigtable and the parameters discussed below, refer to the [Cloud Bigtable documentation](#) and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Bigtable

Copy of Records	Default Configurations	User-Controlled Configurations (ex. via DPT)
Infrastructure Security	Replication in different zones of the same region typically reduces the replication latency between the clusters.	The template here can be configured to have up to four clusters. Considering the higher replication latency, these clusters can be created in zones spanning across different regions. Further, each cluster can also be configured to have any number of nodes (1 or more) to restrict or enhance its performance as per the business requirement.

DPT Template Configuration for Google Cloud Bigtable

Note: For customizable parameter options in the template below, please refer to Terraform documentation for [Bigtable Instance](#) and [Bigtable Table](#). The configurable values in the template below are indicative only. Please modify it to match specific requirements in the context of usage.

```

Template (Please refer to accompanying *.yaml template for detailed configuration)
terraform_deployments:
  resources:
    config:
      resource:
        - google_bigtable_instance:
            - instance:
                - cluster:
                    # appropriate id must be chosen based on its purpose
                    - cluster_id: tf-instance-cluster
                    # Code Block 4.2.a
                    num_nodes: 3
                    storage_type: HDD
                    zone: europe-west1-b
                    name: tf-instance
                # Deploying table under bigtable instance
                # This section must be changed as per the data
            - google_bigtable_table:
                - table:
                    - instance_name: ${google_bigtable_instance.instance.name}
  
```

<pre> name: tf-table split_keys: - a - b - c </pre>		
Infrastructure Security	Disaster Recovery	Refer to code block 4.2.a num_nodes - The number of nodes in the Cloud Bigtable cluster.

4.3 Google Cloud Firewall

Google Cloud Firewall is used to enforce network rules and to control which IP's are allowed to access resources on Google Cloud. Cloud Firewall rules can be used to limit access to SQL and Google Cloud Storage from external networks and to also enforce network segmentation and control internal access within the VPC's and between VPC's.

To learn more about Google Cloud Firewall and the parameters discussed below, refer to the [Cloud Firewall documentation](#) and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Cloud Firewall

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Infrastructure Security	This template here is configured to use a private VPC network and a firewall allowing access through some ports using protocols like TCP and ICMP.	<p>The firewall rules can be customized to block or allow traffic on ports or by protocol.</p> <p>The firewall rules can also be applied based on an IP address that belongs to either a source or a destination range, or an IP Address which has a particular tag or even an IP address with a particular.</p>

DPT Template Configuration for Google Cloud Firewall

Template (Please refer to accompanying *.yaml template for detailed configuration)

```

compute_firewalls:
- name: test-firewall
  network: example-network
  # Code Block 4.3.a
  allow:
  - protocol: icmp
  - protocol: tcp
    ports:
    - '80'
    - '8080'
    - 1000-2000
  source_tags:
  - web
  # For EGRESS traffic, it is NOT supported to specify source_ranges OR
source_tags.
  # Code Block 4.3.b
  direction: INGRESS
  # Uncomment the following attributes as per the requirements. For more
information on the attributes, refer to
"https://www.terraform.io/docs/providers/google/r/compute\_firewall.html".
  # Code Block 4.3.c
  # destination_ranges:
  # Code Block 4.3.d
  # source_ranges:
  # Code Block 4.3.e
  # source_service_accounts:
  # Code Block 4.3.f
  # source_tags:
  # Enable logging for a particular firewall rule. If logging is enabled,
logs will be exported to Stackdriver
  # Code Block 4.3.g
  enable_logging: true

```

<p>Infrastructure Security</p>	<p>Network Security</p>	<p>Refer to Code Block 4.3.a</p> <p>allow - The list of ALLOW rules specified by this firewall. Each tuple of the allow block represents a PERMITTED connection. It has 2 fields:</p> <ul style="list-style-type: none"> • protocol - The IP Protocol to which this rule will apply. Example - TCP, UDP, ICMP, etc. • ports - List of ports to which this rule applies.
---------------------------------------	--------------------------------	--

		<p>Refer to Code Block 4.3.b direction - The direction in which traffic flows. Accepted values are INGRESS (incoming) or EGRESS (outgoing). The default value is INGRESS.</p> <p>Refer to Code Block 4.3.c destination_ranges - Ensures that the firewall rule(s) will apply only to traffic that has a destination IP address in these ranges. These ranges must be expressed in CIDR format. Only IPv4 is supported</p> <p>Refer to Code Block 4.3.d source_ranges - Ensures that the firewall rule(s) will apply only to traffic that has source IP addresses in these ranges. These ranges must be expressed in CIDR format. Only IPv4 is supported.</p> <p>Refer to Code Block 4.3.e source_service_accounts - The firewall rule(s) will apply only to traffic originating from an instance with a service account in this list.</p> <p>Refer to Code Block 4.3.f source_tags - The firewall rule(s) will apply only to traffic with source IP that belongs to a tag listed in source tags.</p> <p>Refer to Code Block 4.3.g enable_logging - This field denotes whether to enable logging for a particular firewall rule.</p>
--	--	--

4.4 Google Cloud SQL

Google Cloud SQL is a fully-managed relational database compatible with MySQL, PostgreSQL, and SQL Server. Cloud SQL ensures reliability and security through regular security updates and by providing options for high availability, automated backups, etc.

SQL Instances, similar to Cloud Storage buckets, are used to store raw data imported from various healthcare systems. Cloud SQL is used as intermediate storage for holding data and normalizing it before importing it to BigQuery. Compared to BigQuery which serves both as a sink and a source, Cloud SQL is an application datastore handling large amounts of data transactions requiring consistency.

To learn more about Cloud SQL and the parameters discussed below, refer to the [Cloud SQL documentation](#) and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Cloud SQL

Copy of Records	Default Configurations	User-Controlled Configurations (ex. via DPT)
Data Security	Data is encrypted at-rest and in-transit, with Google managed encryption keys.	Custom encryption keys can be used instead of Google Cloud managed keys. Refer Customer-managed Encryption Keys for more information. See this whitepaper for more information on how Google encrypts data at rest by default and to understand the key management options.

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Infrastructure Security	Cloud SQL instances are exposed by default to the internet and are placed in a default VPC.	Instances can either be confined to a private network or made publicly accessible by modifying the network configurations in the template.
Resilience	A read/read-only replica is not created by default for Cloud SQL.	A failover replica is configured for the main SQL instance which ensures the availability of data in case of a physical or/and a technical incident.

DPT Template Configuration for Google Cloud SQL

Note: For options for the customizable parameters in the template below, please refer to [Cloud SQL guidance](#) for Terraform. The configurable values in the below template are indicative only. Please modify it to match specific requirements in the context of usage.

Template (Please refer to accompanying *.yaml template for detailed configuration)

```
terraform_deployments:
  resources:
    config:
      resource:
        - google_sql_database_instance:
            instance:
              name: {{.MASTER_CLOUD_SQL_NAME}}
              region: {{.REGION}}
              depends_on:
                - google_service_networking_connection.private_vpc_connection
                # Code Block 4.4.a
                # encryption_key_name: # {full path to the encryption key used
for the CMEK disk encryption}
              settings:
                availability_type: ZONAL
                tier: db-f1-micro
                # Code Block 4.4.b
                ip_configuration:
                  ipv4_enabled: false
                  private_network:
"$google_compute_network.private_network.self_link"
                # Code Block 4.4.c
                backup_configuration:
                  binary_log_enabled: true
                  enabled: true
            replica-instance:
              name: {{.REPLICA_CLOUD_SQL_NAME}}
              region: {{.REGION}}
              master_instance_name:
"$google_sql_database_instance.instance.name"
              depends_on:
                - google_service_networking_connection.private_vpc_connection
              settings:
                availability_type: ZONAL
                tier: db-f1-micro
                ip_configuration:
                  ipv4_enabled: false
                  private_network:
"$google_compute_network.private_network.self_link"
                # Code Block 4.4.d
                replica_configuration:
```

<pre>failover_target: true master_heartbeat_period: 60000</pre>		
Data Security	Encryption and Key Management	<p>Refer to Code Block 4.4.a</p> <p>default_encryption_configuration - The default encryption key for all tables in the dataset. Once this property is set, all newly created partitioned tables in the dataset will have an encryption key set to this value, unless table creation request (or query) overrides the key. It has the following parameter:</p> <p>kms_key_name - Google Cloud KMS encryption key that will be used to protect the destination BigQuery table. If a custom KMS key is not specified, Google Cloud default encryption will be used for the data at rest. See this whitepaper for more information on how Google encrypts data at rest by default and to understand the key management options.</p>
Infrastructure Security	Network Security	<p>Refer to Code Block 4.4.b</p> <p>settings.ip_configuration - Configuration for the IP traffic directed to the database instance. It is critical to have the best settings for the traffic to ensure security. It has the following parameters:</p> <ul style="list-style-type: none"> • ipv4_enabled - Whether this Cloud SQL instance should be assigned a public IPV4 address. Either <code>ipv4_enabled</code> must be enabled, or a <code>private_network</code> must be configured. • private_network - The VPC network from which the Cloud SQL instance is accessible

		<p>for private IP. For example, projects/myProject/global/networks/default. Specifying a network enables private IP. Either ipv4_enabled must be enabled, or a private_network must be configured. This setting can be updated, but it cannot be removed after it is set.</p>
<p>Resilience</p>	<p>Disaster Recovery</p>	<p>Refer to Code Block 4.4.c settings.backup_configuration - Configuration for backup of the database for added reliability. It has the following parameters:</p> <ul style="list-style-type: none"> ● binary_log_enabled - True if binary logging is enabled. ● enabled - (Optional) True if backup configuration is enabled. ● start_time - HH:MM format time indicating when backup configuration starts. <hr/> <p>Refer to Code Block 4.4.d replica_configuration - Configuration for the replication process and associated operations for the database. It has the following parameters:-</p> <ul style="list-style-type: none"> ● failover_target - Specifies if the replica is the failover target, this means if the master database instance fails due to some reason, the replica will take over as the new master instance. ● master_heartbeat_period - Time in milliseconds between replication heartbeats.

4.5 Deep Learning Virtual Machines

Deep learning VMs mentioned in this architecture are a set of preconfigured Debian 9-based Compute Engine virtual machine images optimized for data science and machine learning tasks. All images come with key ML frameworks and tools pre-installed with integrated support for JupyterLab, and can be used out of the box on instances with GPUs to accelerate the data processing tasks.

Access to the VM instance should be restricted to secure shell (SSH) or remote desktop protocol (RDP) via secure key-based access. To support resilience, snapshots of the Compute Engine instances should be enabled to back up data periodically.

To learn more about Compute Engine, Deep Learning VMs and the parameters discussed below, refer to the [Compute Engine documentation](#), [Deep Learning VM documentation](#), and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Deep Learning VM Instance

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Identity and Access Management	Users in the owners' group of the project are allowed to perform any action allowed by the project's organization policies.	Use G Suite to create users and groups. Additional role-member bindings can be added as mentioned in the template to control access to the compute instance.
Data Security	Data is encrypted at-rest and in-transit, with Google managing the security keys.	Custom encryption keys can be used for encryption of data stored on Google Compute Engine disks. See Customer-managed Encryption Keys for more information. See this whitepaper for more information on how Google encrypts data at rest by default and to understand the key management options.
Infrastructure Security	The instance can either be confined to use a private network or just a subnet of that network. Also, the instance may be assigned a public IP.	The VM instance is created under a subnet of a user-defined VPC network as configured in the template and public access can be restricted as required.

Copy of Records	Default Configurations	User-Controlled Configurations (ex. via DPT)
Resilience	A default disk is attached to the compute instance. The compute instance can use any number of disks as required.	The template can be configured to attach a compute disk to the compute instance and enable its backup by a compute snapshot to recover data in case of a technical incident.

DPT Template Configuration for Deep Learning VM Instance

```

Template (Please refer to accompanying *.yaml template for detailed configuration)
compute_instances:
- name: {{.DEEP_LEARNING_VM_NAME}}
  zone: {{.ZONE}}
  machine_type: n1-standard-1
  # Code Block 4.5.a
  _iam_members:
  - role: roles/editor
    member: {{.DEEP_LEARNING_VM_EDITOR_ROLE_MEMBER}}
  - role: roles/viewer
    member: {{.DEEP_LEARNING_VM_VIEWER_ROLE_MEMBER}}
  # Code Block 4.5.b
  attached_disk:
    source: ${google_compute_disk.deep-learning-vm-disk.self_link}
    mode: READ_ONLY
    # A 256-bit customer supported key stored with them.
    # disk_encryption_key_raw:
    # A key stored on Google Cloud KMS.
    # kms_key_self_link: (ex.{google_kms_crypto_key.gcs.self_link})
  # Code Block 4.5.c
  deletion_protection: false
  Boot_disk:
    # Code Block 4.5.d
    auto_delete: false
    mode: READ_WRITE # can be changed to READ_WRITE or READ_ONLY
    initialize_params:
    # deep learning vm containing tensorflow pre-installed as example. Use
    "gcloud compute images list --project deeplearning-platform-release
    --no-standard-images" to find a list of more VMs users can use.
    # using VMs with GPU requires specific configurations. Check the deep
    learning VM Google documentation for the same.
    image:
    https://www.googleapis.com/compute/v1/projects/deeplearning-platform-release
    /global/images/tf2-latest-cpu-20200227
    # A 256-bit customer supported key stored with them.
  
```

```
# disk_encryption_key_raw:
# A key stored on Google Cloud KMS.
#   kms_key_self_link: (ex.{google_kms_crypto_key.gcs.self_link})
# Code Block 4.5.e
labels:
  data_criticality: {{.DEEP_LEARNING_VM_DATA_CRITICALITY_LABEL}}
  datatype: {{.DEEP_LEARNING_VM_DATA_TYPE_LABEL}}
  project: {{.PROJECT_ID}}
# Code Block 4.5.f
network_interface:
  subnetwork:
    ${google_compute_subnetwork.deep-learning-vm-subnetwork.self_link}
  service_account:
    email: ${google_service_account.deepvmsa.email}
    scopes:
      - bigquery
      - sql-admin
      - userinfo-email
      - compute-ro
      - storage-ro
# allow_stopping_for_update: true / false
# Enables an instance to be stopped for an updation purpose)
# enable_secure_boot: true

terraform_deployments:
  resources:
    config:
      resource:
        - google_compute_snapshot:
            deep-learning-vm-snapshot:
              name: deep-learning-vm-snapshot
              source_disk: ${google_compute_disk.deep-learning-vm-disk.name}
              zone: {{.ZONE}}
              labels:
                project: {{.PROJECT_ID}}
              connected_disk:
                ${google_compute_disk.deep-learning-vm-disk.name}
                # Customer Managed Keys must be configured here
                # snapshot_encryption_key:
                #   raw_key:
                #   sha256:
        - google_compute_disk:
            deep-learning-vm-disk:
              name: deep-learning-vm-disk
              type: pd-ssd
              zone: {{.ZONE}}
              labels:
```

<pre> project: {{.PROJECT_ID}} connected_instance: {{.DEEP_LEARNING_VM_NAME}} physical_block_size_bytes: 4096 # Customer Managed Keys must be configured here # disk_encryption_key: # raw_key: # sha256: # kms_key_self_link: (ex. {google_kms_crypto_key.gcs.self_link}) - google_compute_subnetwork: - deep-learning-vm-subnetwork: - name: deep-learning-vm-subnet network: \${google_compute_network.private_network.self_link} region: {{.REGION}} ip_cidr_range: 10.3.0.0/16 </pre>		
Identity and Access Management	User access control	<p>Refer to Code Block 4.5.a</p> <p>iam_members - Member role for the user. G Suite users/groups and Cloud IAM roles can be used to control access.</p>
Resilience	Disaster Recovery	<p>Refer to Code Block 4.5.b</p> <p>attached_disk - An array of objects where each object defines a disk attached to the compute instance.</p>
Data Security	Information Lifecycle Management	<p>Refer to Code Block 4.5.c</p> <p>deletion_protection - This feature ensures that the compute instance cannot be deleted and defaults to false. Refer to this document for more information.</p> <p>auto_delete - Whether the disk attached to the compute instance is automatically deleted after the instance is deleted or not. Defaults to true.</p>
	Encryption & Key Management	<p>Refer to Code Block 4.5.d</p> <p>disk_encryption_key_raw - A 256-bit custom managed encryption key to encrypt this disk. If a custom key is not specified, Google Cloud default encryption will be used for the data</p>

		<p>at rest. See this whitepaper for more information on how Google encrypts data at rest by default and to understand the key management options.</p> <p>kms_key_self_link - A key stored on Google Cloud KMS. If a custom KMS key is not specified, Google Cloud default encryption will be used for the data at rest. See this whitepaper for more information on how Google encrypts data at rest by default and to understand the key management options.</p>
	<p>Labels</p>	<p>Refer to Code Block 4.5.e</p> <p>Labels are used to identify assets based on their classification. This can be used to identify the appropriate types of data stored within the service.</p>
<p>Infrastructure Security</p>	<p>Network Security</p>	<p>Refer to Code Block 4.5.f</p> <p>network_interface.network - The name or self_link of the network to attach this instance to. The network must exist in the same region this instance will be created in. Either network or subnetwork must be provided.</p> <p>network_interface.subnetwork - The name or self_link of the subnetwork to attach this instance to. The subnetwork must exist in the same region this instance will be created in. Either network or subnetwork must be provided.</p>

4.6 Google Cloud Dataproc

Google Cloud Dataproc is a managed Apache Spark and Apache Hadoop service that lets you take advantage of open source data tools for batch processing, querying, streaming, and machine learning. Dataproc automation helps you create clusters quickly, manage them easily, and save money by turning clusters off when you don't need them. Dataproc uses image templates to bundle operating system, big data components (Hadoop, Spark, Hive, and Pig), and Google Cloud Platform connectors into a package deployed on a cluster.

A Dataproc cluster is deployed as a part of the architecture which can be used to execute various big data jobs such as Hadoop, Spark, Hive and Pig jobs as necessary.

To learn more about Cloud Dataproc and the parameters discussed below, refer to the [Cloud Dataproc Documentation](#) and [resource configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Cloud Dataproc

Validation	Default Configurations	User-Controlled Configurations (ex. via DPT)
Data Security	Data is encrypted at-rest and in-transit, with Google managing the security keys.	Custom encryption keys can be used for encryption of data stored on Google Compute Engine disks. See Customer-managed Encryption Keys for more information.
Infrastructure Security	Auto-scaling is available but not enabled by default for Cloud Dataproc.	The auto-scaling policy attached to the cluster can be edited for scaling boundaries, frequency, and aggressiveness to provide fine-grained control over cluster resources throughout cluster lifetime.
Identity and Access Management	Users in the owners' group of the project are allowed to perform any action allowed by the project's organization policies.	Use G Suite to create users and groups. Additional role-member bindings can be added as mentioned in the template to control access to the Dataproc cluster.

DPT Template Configuration for Google Cloud Dataproc

Note: For options for the customizable parameters in the template below, please refer to Terraform documentation for [Dataproc Cluster](#) and [Dataproc Job](#). The configurable values in the

below template are indicative only. Please modify it to match specific requirements in the context of usage.

Template (Please refer to accompanying *.yaml template for detailed configuration)

```
terraform_deployments:
  resources:
    config:
      resource:
        # Dataproc cluster
        - google_dataproc_cluster:
            mycluster:
              name: {{.DATAPROC_CLUSTER_NAME}}
              region: {{.REGION}}
              labels:
                project: {{.PROJECT_ID}}
            cluster_config:
              master_config:
                num_instances: 1
                machine_type: n1-standard-1
                disk_config:
                  boot_disk_type: pd-ssd
                  boot_disk_size_gb: 15
              worker_config:
                num_instances: 2
                machine_type: n1-standard-1
                min_cpu_platform: Intel Skylake
                disk_config:
                  boot_disk_size_gb: 15
                  num_local_ssds: 1
              preemptible_worker_config:
                num_instances: 0
              software_config:
                image_version: 1.3.7-deb9
                override_properties:
                  dataproc:dataproc.allow.zero.workers: 'true'
            # KMS keys configuration can be enabled. Refer
            # https://www.terraform.io/docs/providers/google/r/dataproc\_cluster.html
            # Code Block 4.6.a
            # security_config:
            #   kerberos_config:
            #     kms_key_uri: (ex. {google_kms_crypto_key.gcs.self_link})
            #     root_principal_password_uri: bucketId/o/objectId
            # Code Block 4.6.b
            autoscaling_config:
              policy_uri: "${google_dataproc_autoscaling_policy.asp.name}"
            gce_cluster_config:
```

```

    service_account:
{{.DATAPROC_SERVICE_ACCOUNT_NAME}}@{{.PROJECT_ID}}.iam.gserviceaccount.com
  - google_dataproc_autoscaling_policy:
    asp:
      policy_id: dataproc-policies
      location: {{.REGION}}
      worker_config:
        min_instances: 2
        max_instances: 5
      basic_algorithm:
        yarn_config:
          graceful_decommission_timeout: 30s
          scale_up_factor: 0.5
          scale_down_factor: 0.5
# IAM bindings for dataproc
# Code Block 4.6.c
  - google_dataproc_cluster_iam_binding:
    - editor:
      - cluster: ${google_dataproc_cluster.mycluster.name}
        members:
        - {{.DATAPROC_CLUSTER_EDITOR_ROLE_MEMBER}}
        role: roles/editor
        region: {{.REGION}}
    - viewer:
      - cluster: ${google_dataproc_cluster.mycluster.name}
        members:
        - {{.DATAPROC_CLUSTER_VIEWER_ROLE_MEMBER}}
        role: roles/viewer
        region: {{.REGION}}
  - google_service_account:
    dataprocsa:
      account_id: {{.DATAPROC_SERVICE_ACCOUNT_NAME}}

```

Data Security

Encryption & Key Management

Refer to Code Block 4.6.a

security_config.kerberos_config.km
s_key_uri - The URI of the KMS key used to encrypt various sensitive files. It is a key stored on Google Cloud KMS. It is an attribute mentioned under the security_config block of the template. If a custom KMS key is not specified, Google Cloud default encryption will be used for the data at rest. See [this](#) [whitepaper](#) for more information on

		how Google encrypts data at rest by default and to understand the key management options.
Infrastructure Security	Capacity Management	Refer to Code Block 4.6.b cluster_config.autoscaling_config.policy_uri - The autoscaling policy config associated with the cluster.
Identity and Access Management	User access control	Refer to Code Block 4.6.a google_dataproc_cluster_iam_bindings - It is a list of role-member bindings. Each item in a list is populated with a role and a list of members that are granted the role.

4.7 Google Cloud Datalab

Cloud Datalab is a powerful interactive tool, built on Jupyter, to explore, analyze, transform, and visualize data and build machine learning models on Google Cloud Platform. It runs on Compute Engine and connects to multiple cloud services easily so you can focus on your data science tasks.

To learn more about Cloud Datalab and parameters discussed below, refer to the [Cloud Datalab Documentation](#) and [module configuration](#) respectively.

Title 21 CFR Part 11 Alignment for Google Cloud Datalab

<i>Copy of Records</i>	<i>Default Configurations</i>	<i>User-Controlled Configurations (ex. via DPT)</i>
Data Security	The datalab module has a persistent disk provisioned by default and its contents are automatically backed-up into Cloud Storage.	NA - This is enabled by default when provisioning Datalab. This can be disabled but is not recommended. DPT retains the default configuration as enabled.

<i>Validation</i>	<i>Default Configurations</i>	<i>User-Controlled Configurations (ex. via DPT)</i>
Infrastructure Security	The datalab module is not attached to the network by default and is available via external IP.	The module can either be confined to use a private network or just a subnet of that network. Absence of

		configuration for the network would fail the deployment.
--	--	--

DPT Template Configuration for Google Cloud Datalab

Note: For options for the customizable parameters in the template below, please refer to Terraform documentation for [Datalab](#). The configurable values in the template below are indicative only. Please modify it to match specific requirements in the context of usage.

Template (Please refer to accompanying *.yaml template for detailed configuration)

```

terraform_deployments:
  resources:
    config:
      module:
        # Refer
        "https://github.com/terraform-google-modules/terraform-google-datalab" for
        more information.
        - datalab:
          - datalab_user_email: user@domain
            # adding gpu requires a quota assigned. Refer to link
            "https://github.com/terraform-google-modules/terraform-google-datalab/tree/m
            aster/examples/basic" for further information.
            project_id: gcprd-project-name
            source: terraform-google-modules/datalab/google//modules/instance
            # Code Block 4.7.a
            network_name: ${google_compute_network.private_network.name}
            subnet_name: ${google_compute_subnetwork.datalab-subnetwork.name}
            version: "~> 1.0"
            zone: europe-west3-c
            # custom service account with the deployer given
            serviceAccountUser role
            service_account:
            ${google_service_account.datalab-service-account.email}
            datalab_enable_swap: true
            # Code Block 4.7.b
            create_disk: true
            datalab_enable_backup: true
            datalab_console_log_level: "warn"
            datalab_idle_timeout: "60m"
          resource:
            - google_compute_subnetwork:
              - datalab-subnetwork:
                - name: datalab-subnet
  
```

<pre> network: \${google_compute_network.private_network.self_link} region: europe-west3 ip_cidr_range: 10.2.0.0/16 # Custom created service account service_accounts: - account_id: datalab-service-account </pre>		
Infrastructure Security	Network Security	<p>Refer to code block 4.7.a</p> <p>network_name - The VPC network under which the Cloud Datalab instance is created. This setting must always be configured.</p> <p>subnet_name - The subnet under the previously mentioned VPC network under which the Cloud Datalab instance is created. This setting must always be configured.</p>
Data Security	Information Lifecycle Management	<p>Refer to code block 4.7.b</p> <p>create_disk - Created a persistent data disk and attaches it to the Datalab module.</p> <p>datalab_enable_backup - Automatically does backup of the disk's contents to Cloud Storage</p>

A. Appendix

A.1 GxP, CFR 21 Part 11, and Google Cloud

As an industry-leading cloud service provider (CSP), *Google Cloud Platform (GCP)* enables healthcare organizations to realize the benefits of cloud computing and stay on track with their GxP efforts. Additionally, Google Cloud can help align customers with Title CFR 21 Part 11 regulations if Google Cloud is used in a specified manner to handle data and processing. While Google has cloud resources that are ready for many Title CFR 21 Part 11 compliant workloads, the overall compliance depends on the configuration used by the customer.

Google Cloud customers own their data and control how it is used. Google Cloud acts as a data processor and only processes data in order to provide the stated services to customers. It is crucial to remember that enterprises and individuals utilizing Google Cloud are responsible for understanding GxP and its implications concerning use of Google Cloud products and services hosting applications and services. Some of these aspects are listed below.

- Applicability of the provisions and requirements of GxP across applications, platforms, and Google Cloud Infrastructure
- Classification and inventory of data, particularly personal data, along with the business and information systems that process this data
- Alignment of current controls, policies, and processes for managing and protecting healthcare data with GxP requirements
- Understanding of the existing data protection features on Google Cloud for meeting GxP requirements
- Review and acceptance of Google Cloud's data processing terms for the G Suite and Cloud Identity Data Processing Amendment ([link](#)) and for the Google Cloud Data Processing and Security Terms ([link](#)).
- Monitoring the deployment and code management processes and technology for security and compliance for improving compliance of service deployed.

Please refer to the following documents carefully before proceeding:

- Title 21 CFR Part 11 Overview Guide: <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application>
- Keys for 21 CFR Compliance (Gartner): <https://www.gartner.com/en/documents/357755/the-keys-to-21-cfr-part-11-compliance>
- GCP Terms of Service: <https://cloud.google.com/terms>

A.2 Compliance

Security and Privacy on the Cloud is a shared responsibility. Google Cloud is responsible for the security of the cloud and Google Cloud's customers are responsible for their security in the cloud. Google Cloud's focus on data security, privacy, and transparency has provided a foundation towards achieving GxP compliance for Google Cloud. Google Cloud offers data privacy, data portability, and threat protection products and features that can support GxP compliance efforts. These capabilities can be leveraged not only to prevent abuse or unauthorized access to personal data but also to maintain security of data and meet GxP requirements.

G Suite and Google Cloud Platform are regularly tested, assessed, and evaluated for the effectiveness of technical and organizational security and privacy measures via third-party audits and certifications as listed below.

1. [ISO 27001](#) for information security management systems
2. [ISO 27017](#) for cloud security controls
3. [ISO 27018](#) for protection of personally identifiable information (PII) in public clouds acting as PII processors
4. [SOC 2](#) and [SOC 3](#) for evaluating systems' and controls' security, availability, processing integrity, and confidentiality or privacy

Google Cloud Platform is also certified under the [HITRUST CSF](#) security framework and has attained a [CSA Star SOC2+ report](#).

To learn more about security and compliance for the Google Cloud Platform, refer to [Cloud Compliance & Regulations Resources](#).

A.3 Google Cloud Shared Responsibility Model

The shared responsibility model depends on the particular service model. This starts from the bottom of the stack and moves upwards, from the infrastructure as a service (IaaS) layer where only the hardware, storage, and network are Google’s responsibility, up to software as a service (SaaS) where most components of the stack except the content (i.e., data) and access policies are up to the provider.

To learn more about Google Cloud's Shared Responsibility Model, refer to the [Google Infrastructure Security Design Overview](#).



Figure 42 - Google Cloud Shared Responsibility Model

In general, Google is responsible for the security of the underlying infrastructure, including hardware, firmware, kernel, OS, storage, network, and more. This includes encrypting data at rest by default, encrypting data in transit, using custom-designed hardware, laying private network cables, protecting data centers from physical access, and following secure software development practices.

Conversely, customer responsibility for security and compliance in the cloud is listed in Appendix A.4.

A.4 Customer Responsibilities

The following are typical examples of security and compliance capabilities for which the customer is responsible. This is not an exhaustive list.

A.4.1 Identity and Access Management

1. User access provisioning
2. Custom roles to control access
3. Monitoring to detect unusual activity by users and administrators
4. Role-based access controls and separation of duties
5. Multi-factor authentication and 2-step verification for access critical environments and sensitive data
6. Periodic cadence for reviewing access lists

A.4.2 Governance, Risk and Compliance

1. Governance and implementation of organization-specific security policies and standards
2. Definition of security-specific key performance indicators (KPIs) and key risk indicators (KRIs)
3. Security awareness training and secure coding practices training and reporting
4. Background verification checks by authorized parties prior to granting access

A.4.3 Data Security

1. Consent collection, logging, tracking, and monitoring by end-users for organizational access to their PII, PHI, or other sensitive data
2. Data governance lifecycle and data management strategy
3. Data classification, labels, and handling as per regulatory requirements, service level agreements and operational continuity requirements
4. Policies and procedures for reuse, disposal, and deletion of resources (e.g., data, equipment, and digital media)
5. Data destruction, obfuscation, and archival standards, including supporting tools and technologies
6. Key management, if customers choose to use the Customer-Managed Encryption Keys or Customer-Supplied Encryption Keys (CSEK) capabilities of specific products.
7. Communication of incidents and notifications about data breaches, including reporting to regulatory authorities and customers

A.4.4 Infrastructure Security

1. Configuration and validation of firewall rules for both egress and ingress traffic
2. Penetration testing, black-box testing, and red teaming exercises

A.4.5 Security Operations

1. Audit logs for security events, faults, exceptions, and data access violations
2. Enabling of data-read and data-write logs for all critical environments and projects, which are reviewed on a periodic basis
3. Event logs stored and backed up at a centralized storage location and which are protected against tampering and unauthorized access
4. Clocks for all resources are in sync with the approved time sources like the network time protocol (NTP) servers/domain controllers.