Google Cloud

PROFESSIONAL SERVICES

Technical Infrastructure Design Document

Google Cloud Professional Services

Date: 20 December 2024

Authors: Google Cloud Team

Prepared for: Quest

Document type: Infrastructure Design Doc

Contents

1. Introduction

Google Cloud Platform offers flexibility and performance at scale for corporate- and public-facing applications. Integrating GCP infrastructure across an organization extends those benefits to all areas of the business, with the additional requirement in this project of establishing a HIPAA-compliant environment. Coordinated efforts across teams and projects are essential to building a secure and adaptable platform that meets both current and future project requirements. Key activities such as provisioning user accounts, enforcing access controls, auditing network security, and configuring chargebacks can be managed effectively when all relevant teams are engaged from the outset. This Technical Design Document outlines the decisions and strategies for addressing these design requirements, establishing a secure and adaptable foundation.

This Document is a living document. As of 12/21/2024 the ownership and authoring of the document has been taken over by Quest. Changes may be made with Track Changes on.

Comments may also be made but please provide solutions along with any concerns you have. The Architecture team will review and approve / accept changes periodically as we move forward. Because the project is still in heavy change and development we will work out of the project folder for now but eventually this document will move to the Knowledge Base when the change rate slows down likely some time in February 2025.

1.1 High-level Quest Overview

Quest is pursuing a multi-cloud environment, with a strategic focus on establishing Google Cloud Platform (GCP) as the data analytics and AI primary cloud within their organization. Google's goal is to enable the Quest team by providing guidance and tools necessary to begin developing on GCP. Google and Quest will collaborate throughout the program to document and set up a GCP environment that aligns with HIPAA compliance standards. This document represents the outcomes, design decisions, and architecture developed in accordance with Google's best practices during the Infrastructure Cloud planning phase.

2. Identity management

The purpose of identity management design is to reliably authenticate users' or services' identity and guard against loss of credentials and attempts at impersonation.

2.1 Cloud Identity

Google Cloud Identity is the product used for managing users, groups, and domain-wide security settings for Google Cloud Platform. Cloud Identity is tied to a unique DNS domain that needs to be enabled for receiving email (e.g., has an appropriate MX configured) so that users and groups configured with responsibilities in GCP can receive generated notifications. Cloud Identity configurations are made in the Admin Console. Existing Google Workspace Quest users can use their Google Admin Console for Cloud Identity. Quest users without an existing Google Workspace account, we can create a Cloud Identity in the "IAM" section of the GCP Cloud Console.

Background

Quest provisions all users through Microsoft Azure Active Directory (Azure AD), which supports SAML 2.0. This existing Azure AD infrastructure will be leveraged for authentication into GCP, providing seamless integration and centralized identity management across both environments.

The source of identities

The existing Active Directory (AD) Identity provider for the 'questdiagnostics.com' domain will be used as the source of truth for provisioning subset groups to Google Cloud Identity. The particular groups available in GCP will be a subset of those in Quest's AD and synchronized with Google Cloud Directory Sync (GCDS). This connection with Entra ID will sync over the selected groups. Sync will occur approximately every 40 minutes as long as the Microsoft Entra provisioning service is running. It is important for Quest to consider the processes involved when making changes to the enterprise application when teams need to onboard with GCP.

While the majority of users will be provisioned through automated mechanisms, there exists a cohort of approximately 125 users currently residing in Google Workspace as "unmanaged" accounts. These users require a transition to "managed" status to align with the organization's security and access control policies.

To complete their onboarding, these users must adhere to the standard authentication process established for all managed users. This process may involve steps such as:

- **Identity Verification:** Confirming their identity through multi-factor authentication.
- **Account Setup:** Configuring their profile, setting up security questions, and enrolling in any necessary security measures (e.g., single sign-on, device management).
- **Policy Acknowledgement:** Reviewing and acknowledging relevant organizational policies, including acceptable use policies and data security policies.

By following these recommendations, the organization can ensure a secure and streamlined onboarding experience for all users, including those transitioning from unmanaged to managed accounts.

Authentication

For GCP, user authentication will be implemented via SAML federation to the Quest's Microsoft Entra ID (Azure AD) system. This setup will delegate authentication to Entra ID, enforcing all configured policies, including Multi-Factor Authentication (MFA) if enabled. Enabling MFA is a requirement to enhance security across GCP resources, ensuring access control aligns with Quest's existing identity management policies.

Authorization

Once a user is authenticated in GCP, their access to resources is determined by their Azure AD group memberships and the IAM permissions assigned to those groups. For a user to access or interact with any GCP resource, they must have the appropriate IAM permissions. Without

membership in specific groups with assigned permissions, users can log in but will have no visibility or ability to perform actions.

Accounting

All user activity in GCP will be logged for accountability. This is discussed in more detail in the Monitoring and Security sections.

2.2 Users

Super Admins

The Google Cloud instance will be managed by the Quest Cloud Platform Team, which will use service accounts with the Super Admin role. These service accounts will be used for initial setup and then locked to prevent further access.

Super Admins have the authority to set permissions and establish Organization Admins in GCP. To maintain administrative continuity, Google recommends configuring more than one Super Admin, though these accounts will remain inaccessible after initial configuration to prevent unauthorized access.

Once initial items such as GCDS sync is setup, this account is typically never needed to be used. Auditing and alerting capabilities will be setup to monitor access by Super Admins.

Also of note is that Super Admins have the ability to impersonate any user in GCP by resetting their password, and thus accessing any GCP resources. The Admin Console provides an auditing and alerting capability to identify specifically if and when a Super Admin has performed a password change. In the Admin Console, this capability is configured under 'Reports' > 'Manage alerts', using the 'User's password changed' alert.

GCP Organizational Admins

Organizational administration for GCP is delegated through a hierarchy. This is covered in more detail in the Folder Hierarchy and Projects sections later on.

A brief overview is that there is:

1. **Root organization administrators** who can see over the **entire GCP environment** and are a **very restricted group**. In practice at Quest these members are also probably super admins.

2. **Root organization viewers** who can see **resources** and **billing** over the entire GCP environment.
3. **Master portfolio organization viewers** who can see **resources** and **billing** under their master portfolio folder.
4. **Portfolio organization viewers** who can see **resources** and **billing** under their portfolio.

Membership of these groups is handled through Azure AD and their permissions are defined through terraform.

Note: For getting started it is common to allow the company's core GCP team to have one or two people directly assigned Org-Admin privileges in order to gain expertise with provisioning the overall GCP organization/folder/project hierarchy with automation tools with infrastructure-as-code solutions such as terraform.

Onboarding

Quest will follow its established onboarding processes for employees and vendors, ensuring valid identities are maintained within Azure AD under the questdiagnostics.com domain. From there, assigning users to the appropriate Azure AD groups will enable authentication into Google Cloud. Additional Azure AD group memberships will grant further permissions within GCP based on role requirements.

Offboarding

Quest will follow existing offboarding processes for employees and vendors by removing or deactivating identities in Azure AD under the questdiagnostics.com domain. Additional steps include:

1. Suspending or removing the user in Azure AD.
2. Immediately deactivating or deleting the user in the GCP Admin Console.

These steps address two key cases:

A. GCDS runs on a scheduled basis and there will be a delay between when the user is offboarded in Azure AD and when those changes are synced to GCP.
B. The user in GCP could have authentication tokens checked out with a time expiry (say up to 24 hours). The second step ensures those tokens and any live sessions are revoked immediately

Conflict accounts (Unmanaged accounts)

Conflicting accounts may be created by the user provisioning process in the Admin Console. They occur when an email address used to provision a new user in a Google instance (managed by your organization) is the same email address as is used by a consumer (i.e., personal) Google account. Essentially, if an employee used their 'username@questdiagnostics.com' email address to set up a test or trial account at Google.

If a conflicting account is created during the provisioning process, the next time the user logs into their consumer account, they will be presented with additional information about how to resolve the conflicting account. The user will be directed to this Google Help Center article for further information: https://support.google.com/accounts/troubleshooter/1699308?hl=en

Finding conflict accounts: https://support.google.com/a/answer/6178640

Prevent conflicting accounts: https://knowledge.workspace.google.com/kb/how-to-prevent-unmanaged-google-account-creation-000006505

GCDS Integration

Quest's implementation of GCDS shall be deployed as described below.

Deployment

- Azure AD as Source: GCDS will synchronize directly with Azure AD. Ensure that GCDS has the necessary permissions to read from Azure AD for group synchronization.
- Connection Security: Verify outbound HTTPS (port 443) access to Google's GCP APIs, as GCDS will still need to communicate securely with Google Cloud.

Groups

- Include any groups and associated user membership where the group is prefixed with **'gcp-*'.** Here is the link to the recommended GCP groups to be created in AD.
- Groups can be nested but it is recommended not to nest more than two or possibly three.
- At this time there are no exclusion rules although one that is a negation of the inclusion rule could be applied as a check.

Organizations

- There are no sub organizations to synchronize. Typically this feature is used with G-Suite and not with GCP Cloud Identity.

Authentication and Password Management

- Password Policy: Since authentication will rely on SAML-based SSO, password synchronization remains disabled, and users will authenticate with SAML directly.

Additional Considerations

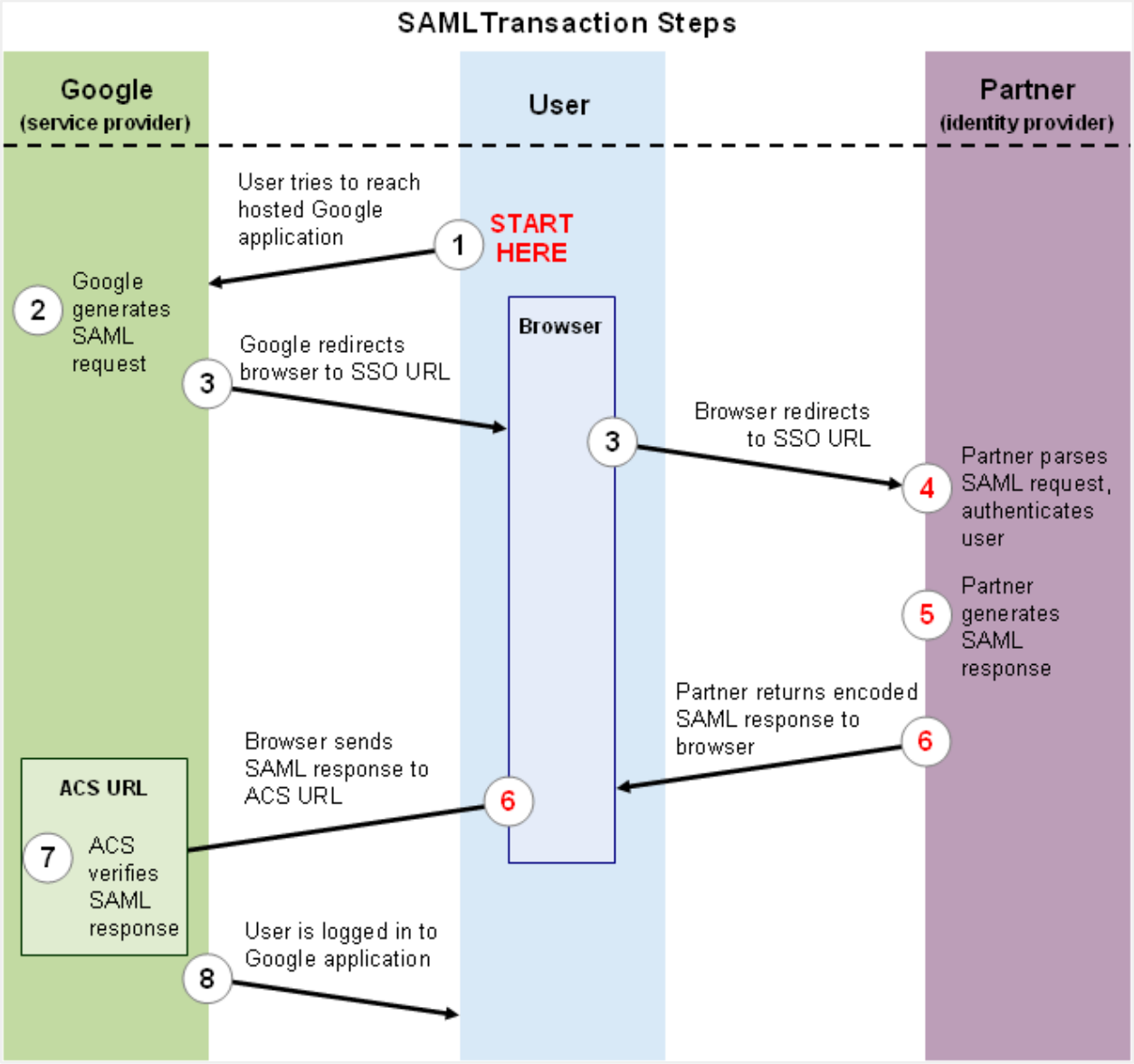
- Azure AD Integration Settings: Verify that GCDS can retrieve necessary group details from Azure AD, including group membership attributes, to match Google Cloud Identity configurations accurately.

GCDS Reference Material

- About Google Cloud Directory Sync
- Cloud Identity to use Microsoft Entra ID
- Best Practices for Enterprise Organizations
- What is GCP Cloud Identity

SAML Integration

Authentication for users after they are synchronized to Google Cloud Identity will be integrated with SAML via Azure Active Directory.



The above is taken from: [Partner Operated SAML / Single Sign On]

2.3 AD Groups

Google recommends using groups to grant roles and permissions to users. Managing access to GCP projects via groups is easier than managing access via individual users, as once project owners have properly associated groups with roles in a GCP project, granting and revoking a user's access to multiple projects can be managed centrally in the Admin Console. Additionally, group membership is auditable through the Admin Console.

For organizations that adhere to the principle of least privilege, groups can be created and associated with narrower levels of access to the services within a GCP project.

LDAP group membership will be managed in Quest's Azure Active Directory, with an emphasis on minimizing the number of actual LDAP groups while still supporting effective governance of GCP resources.

Note: For EOY 2024 we are in a change freeze so no changes can be made to the AD groups with SSO enabled. Further, until cleanup of the unmanaged users is complete SCIM can not be turned on. While those items are being worked and closed any groups created in GCP as per the below must also be created and populated in AD via SNOW ticket.

Group Naming Conventions

The naming convention for A/D group names shall be:

1. **gcp-syncdusers** - a special group for GCDS that contains all users that will be synchronized to google cloud. Membership of this group allows the user to login but does not grant them any privileges to do anything.
2. Global/Org Groups: **gcp-global-<functional-description>**

Examples:

- gcp-organization-admins
- gcp-billing-admins
- gcp-vpc-network-admins
- gcp-hybrid-connectivity-admins

Note that groups can be assigned permissions via roles at different levels in the hierarchy from organization wide to particular folders and individual projects. This is known as 'binding' and is managed within GCP, typically through an infrastructure-as-code tool such as terraform. The permissions assigned to each group are addressed further on.

Global Groups and Groups not affiliated with a particular application

Detailed permissions and where they are bound (org/folder/project) are discussed later under IAM Permissions.

A/D group name	Google Cloud Role	Group purpose
gcp-organization-admins	Organization Administrator	• Administering any resource that belongs to the organization.

		• Assign this role sparingly; org admins have access to all of your Google Cloud resources.
gcp-billing-admins	Billing Account Administrator	• Setting up billing accounts and monitoring their usage.
gcp-vpc-network-admins	Compute Shared VPC Admin	• Creating networks, subnets, firewall rules, and network devices such as Cloud Router, Cloud VPN, and cloud load balancers.
gcp-hybrid-connectivity-admins	Compute Network Admin	• Hybrid Connectivity administrators are responsible for creating network devices such as Cloud VPN instances and cloud routers.
gcp-logging-admins	Security Reviewer	• Setting up centralized logging and monitoring for the entire organization. • Managing alerting and reporting policies, system availability and SLA/SLO requirements.
gcp-logging-viewers	Private Logs Viewer	• Read-Only access to logs for the entire organization.
gcp-monitoring-admins	Monitoring Admin	• Monitoring administrators have access to use and configure all features of Cloud Monitoring
gcp-security-admins	Organization Policy Administrator	• Establishing and managing security policies for the entire organization, including access management and organization constraint policies. See the Google Cloud security foundations guide for more information about planning your Google Cloud security infrastructure.
gcp-developers		• Developers are responsible for designing, coding, and testing applications
gcp-devops		• DevOps practitioners create or manage end-to-end pipelines that support continuous integration and delivery, monitoring, and system provisioning

Group Management in GCP: IAM Roles and Group-Based Permissions

This design synchronizes AD groups to GCP, allowing you to leverage existing AD group memberships for cloud access control. By assigning GCP IAM roles (like admin, editor, or viewer) to these synced groups, you can grant broad permissions while using deny rules and custom roles within GCP to achieve fine-grained restrictions. This approach centralizes identity management in AD, simplifies GCP permission administration, and offers flexibility to tailor access based on specific needs.

```
# **DISCLAIMER:**

# The below Terraform code blocks are provided as examples ONLY.

# They are intended to illustrate concepts and should NOT be copied and
pasted directly into your environment without careful review and
modification.

# Before deploying any Terraform code, ensure that you:

# * **Understand the code:** Thoroughly review and understand the code's
purpose, the resources it creates, and the permissions it grants.

# * **Customize values:** Replace all placeholder values (project IDs,
organization IDs, group names, user emails, etc.) with your specific
values.

# * **Validate configurations:** Double-check all configurations,
especially those related to IAM roles and permissions, to ensure they
align with your security policies and requirements.

# * **Test in a safe environment:** Deploy the code in a non-production
environment first to test its behavior and make any necessary adjustments.

# * **Use appropriate Terraform practices:** Follow best practices for
Terraform, such as using a remote backend for state management, organizing
your code into modules, and implementing proper version control.


# **Failure to carefully review and adapt this code to your specific
environment could result in unintended consequences, including security
vulnerabilities or data loss.**


# Proceed with caution and consult with your security and infrastructure
teams as needed.
```

1. Introduction

- **Purpose:** This section outlines the design and implementation of a flexible and secure IAM permissions model in GCP, leveraging predefined roles (admin, editor, viewer) and a group-based approach for managing access to resources.

- **Scope:** This section covers the following:
  - Definition of admin, editor, and viewer roles and their intended use cases.
  - Strategy for using groups to modify and fine-tune permissions.
  - Test cases to validate the expected behavior of the permission model.

2. Role Definitions

- **Admin (`roles/resourcemanager.organizationAdmin` or `roles/resourcemanager.folderAdmin`):**
  - Highest level of access.
  - Can manage all resources within the organization or folder, including granting and revoking permissions, creating and deleting projects, and setting organization policies.
  - Intended for users with complete administrative control.
- **Editor (`roles/editor` or service-specific editor roles):**
  - Can modify existing resources but typically cannot grant IAM permissions.
  - Provides a balance between administrative control and restricted access.
  - Intended for users who need to actively manage resources but not control permissions.
- **Viewer (`roles/viewer`):**
  - Read-only access to resources.
  - Can view configurations and data but cannot make any changes.
  - Intended for users who need to monitor or analyze resources without making modifications.

3. Group-Based Permission Strategy

- **Purpose:** To provide a flexible and manageable way to modify or fine-tune permissions beyond the capabilities of predefined roles.
- **Methodology:**
  - **Create groups with specific roles:** Define groups with specific IAM roles attached to them.

    - **Add users to groups:** Add users to these groups to grant them the corresponding permissions.
    - **Leverage deny rules (optional):** Use deny rules within groups to restrict specific permissions, even if those permissions are granted by other roles or groups.
- **Example:**
    - Create a "ReadOnly-ProjectA" group with the `Viewer` role on Project A.
    - Add users who need read-only access to Project A to this group.

4. Test Cases

- **Purpose:** To validate the expected behavior of the permission model and ensure that group memberships correctly modify individual permissions.

- **Test Case 1: Viewer Access to Editor Group**
    - **Setup:** A user with the `Viewer` role is added to a group with the `Editor` role on a specific project.
    - **Expected Result:** The user should have `Editor` level access to that project, demonstrating that group permissions add to individual permissions.
- **Test Case 2: Editor Access with Deny Rule**
    - **Setup:** A user with the `Editor` role is added to a group with a deny rule for deleting Compute Engine instances.
    - **Expected Result:** The user should be able to manage other resources in the project but should be prevented from deleting instances, demonstrating the effect of deny rules.
- **Test Case 3: Conflicting Permissions**
    - **Setup:** A user is a member of two groups: one with the `Editor` role on a project and another with the `Viewer` role on the same project.
    - **Expected Result:** The user should have `Editor` access, demonstrating that the most permissive access level prevails in case of conflicts.
-

5. Implementation

Deployment Example of above test cases:

Terraform

```
# Define the project

resource "google_project" "project" {
```

```
  project_id = "test-project-id"
  name       = "Test Project"
}
# Define the groups
resource "google_group" "editor_group" {
  email = "editor-group@yourdomain.com"
}
resource "google_group" "deny_delete_group" {
  email = "deny-delete-group@yourdomain.com"
}
resource "google_group" "viewer_group" {
  email = "viewer-group@yourdomain.com"
}
# Add the user to the groups (replace with your user's email)
resource "google_group_member" "editor_group_member" {
  group = google_group.editor_group.email
  email = "user@example.com"
}
resource "google_group_member" "deny_delete_group_member" {
  group = google_group.deny_delete_group.email
  email = "user@example.com"
}
resource "google_group_member" "viewer_group_member" {
  group = google_group.viewer_group.email
  email = "user@example.com"
}
# --- Test Case 1: Viewer Access to Editor Group ---

# Grant Viewer role to the user directly on the project
resource "google_project_iam_member" "project_viewer_member" {
  project = google_project.project.project_id
```

```
  role     = "roles/viewer"
  member   = "user:user@example.com"
}
# Grant Editor role to the editor group on the project
resource "google_project_iam_binding" "project_editor_binding" {
  project = google_project.project.project_id
  role     = "roles/editor"
  members = ["group:${google_group.editor_group.email}"]
}
# --- Test Case 2: Editor Access with Deny Rule ---
# Grant Editor role to the user directly on the project
resource "google_project_iam_member" "project_editor_member" {
  project = google_project.project.project_id
  role     = "roles/editor"
  member   = "user:user@example.com"
}
# Create a custom role with deny rule for deleting instances
resource "google_project_iam_custom_role" "deny_delete_role" {
  project      = google_project.project.project_id
  role_id      = "deny_delete_instances"
  title         = "Deny Delete Instances"
  description = "Denies permission to delete Compute Engine instances"
  permissions = ["compute.instances.list", "compute.instances.get"] # Allow listing and
getting, but not deleting
 included_permissions = ["compute.instances.list", "compute.instances.get"]
  stage                 = "GA"
}
# Grant the custom role to the deny_delete_group on the project
resource "google_project_iam_binding" "project_deny_delete_binding" {
  project = google_project.project.project_id
  role     = "projects/${google_project.project.project_id}/roles/deny_delete_instances"
  members = ["group:${google_group.deny_delete_group.email}"]
```

```
}
# --- Test Case 3: Conflicting Permissions ---


# Grant Editor role to the editor group on the project
resource "google_project_iam_binding" "project_editor_binding_2" {
  project = google_project.project.project_id
  role    = "roles/editor"
  members = ["group:${google_group.editor_group.email}"]
}
# Grant Viewer role to the viewer group on the project
resource "google_project_iam_binding" "project_viewer_binding" {
  project = google_project.project.project_id
  role    = "roles/viewer"
  members = ["group:${google_group.viewer_group.email}"]
}
```

Explanation:

- **Project:** This defines a project (`test-project-id`).
- **Groups:** This defines three groups: `editor_group`, `deny_delete_group`, and `viewer_group`.
- **Group Members:** This adds the user (`user@example.com`) to all three groups.
- **Test Case 1:**
  - `project_viewer_member`: Grants the `Viewer` role directly to the user.
  - `project_editor_binding`: Grants the `Editor` role to `editor_group`.
- **Test Case 2:**
  - `project_editor_member`: Grants the `Editor` role directly to the user.
  - `deny_delete_role`: Creates a custom role that allows listing and getting instances but denies deleting them.
  - `project_deny_delete_binding`: Grants the custom role to `deny_delete_group`.
- **Test Case 3:**
  - `project_editor_binding_2`: Grants the `Editor` role to `editor_group`.
  - `project_viewer_binding`: Grants the `Viewer` role to `viewer_group`.

Important Notes:

This Terraform code provides an EXAMPLE foundation for testing your IAM scenarios. You can modify and expand it to cover more complex cases and edge cases as needed.

6. Monitoring and Maintenance

- **Regularly audit permissions:** Periodically review user and group permissions to ensure they are still appropriate and aligned with security policies.
- **Monitor audit logs:** Track changes to IAM configurations and access attempts to detect any unauthorized activities.
- **Update documentation:** Keep this document and any related documentation up-to-date with any changes to the permission model.

Here's a breakdown of how AD groups map to GCP groups:

1. Synchronization

- **Not a direct mapping:** AD groups and GCP groups are not the same. They exist in different systems.
- **Synchronization is key:** You need a mechanism to synchronize AD groups to GCP. This typically involves tools or services that:
    - **Read group information from AD:** This includes group names, members, and potentially other attributes.[1]
    - **Create corresponding groups in GCP:** Groups with the same name and members are created in your Google Cloud organization or project.
- **Tools and services:**
    - **Google Cloud Directory Sync (GCDS):** This tool can synchronize users and groups from AD to Google Workspace (which then can be used for GCP authentication).

2. Permissions and IAM

- **GCP groups for IAM:** Once AD groups are synchronized to GCP, you can use those GCP groups in your IAM policies.
- **Assign roles to GCP groups:** You can grant IAM roles to the synchronized GCP groups, just like you would with any other GCP group. This allows you to manage permissions for your AD users in GCP.

3. Considerations

- **Synchronization frequency:** Determine how often you need to synchronize groups. This depends on how often group memberships change in your AD environment.

You can adjust the synchronization frequency for Google Groups with Active Directory by modifying the **Synchronization schedule** setting within your Google Cloud Directory Sync (GCDS) configuration.

To do this, you'll need to edit the GCDS configuration file and adjust the `changeDetectionInterval` value, which is specified in seconds. This setting dictates how often GCDS checks Active Directory for changes to user accounts and group memberships.

- **Attribute mapping:** Ensure that relevant attributes (like group names and members) are correctly mapped from AD to GCP.
- **Nested groups:** Consider how nested groups in AD will be handled in GCP. Some synchronization solutions might support nested groups, while others might flatten them.
- **Group naming conventions:** Align your AD group naming conventions with GCP's requirements and best practices for group names.

Example:

1. You have an AD group called "Developers" that contains all your developers.
2. You use GCDS to synchronize this group to GCP.
3. In GCP, you grant the "Compute Engine Instance Admin" role to the "Developers" group.
4. All the developers in your AD group will now have the ability to manage Compute Engine instances in your GCP project.

Key benefits:

- **Centralized management:** Manage user access and permissions from your existing AD infrastructure.
- **Simplified administration:** Avoid manually creating and managing groups in GCP.
- **Consistency:** Maintain consistent permissions across your on-premises and cloud environments.

By effectively mapping AD groups to GCP groups, you can streamline your identity and access management and ensure a smooth transition to the cloud.

Additive Permissions

- **Roles as the foundation:** Start with the individual roles assigned to a user or service account. These provide the initial set of permissions.
- **Groups as layers:** Each group membership adds another layer of permissions on top of the existing roles.

- **Combined permissions:** The user's effective permissions are the *combination* of all their roles and all the roles inherited from their group memberships.

Example

- User A has the `Viewer` role on a project (read-only access).
- User A is added to a group with the `Editor` role on that project.
- User A's effective permissions are now `Viewer` + `Editor`, giving them both read and write access.

Determining Authoritative Permissions

When there's a conflict between permissions (one role grants access, another denies it), GCP uses a hierarchy to determine which permission is authoritative:

1. **Deny rules:** Deny rules always take precedence. If a deny rule blocks a permission, even if other roles or groups would grant it, the permission is denied.

2. **Individual roles:** If there are no applicable deny rules, individual roles assigned directly to the user take precedence over group roles.

3. **Group roles:** If there are no conflicting individual roles, the most permissive role among the user's group memberships takes precedence.

Example

- User B has the `Editor` role on a project (allowing them to delete instances).
- User B is also a member of a group with a deny rule for `compute.instances.delete` (preventing instance deletion).
- The deny rule takes precedence, so User B cannot delete instances, even though their `Editor` role would normally allow it.

Visualizing the Hierarchy

Think of it like this:

```
Deny Rules  >  Individual Roles  >  Group Roles
(most authoritative)               (least authoritative)
```

Key Takeaways

- **Understand the additive nature:** Group memberships expand permissions, not replace them.
- **Prioritize deny rules:** Deny rules are powerful tools for fine-grained control and security.
- **Be mindful of conflicts:** Carefully consider how roles and group memberships interact to avoid unintended access.
- **Test thoroughly:** Always test your IAM configurations to ensure the resulting permissions are as expected.

Deployment Details:

1. AD Group Synchronization

- **Tooling:** As we discussed, Google Cloud Directory Sync (GCDS) is essential.
- **Scope:** Decide which AD groups need to be synced to GCP. You might not need *all* of them. Focus on groups that require access to GCP resources.
- **Frequency:** Configure how often GCDS or your IdP syncs. More frequent syncs keep GCP up-to-date but use more resources. Start with daily syncs and adjust as needed.
- **Nested Groups:** If you have nested groups in AD, check if your chosen tool supports syncing them as nested in GCP, or if they'll be flattened.

2. Group Mapping and Naming

- **Consistency:** Ideally, AD group names should map directly to GCP group names. This makes things easier to manage.
- **GCP limitations:** Remember GCP's group naming rules (up to 63 characters, lowercase, numbers, hyphens, and dots). Adjust AD group names if needed *before* syncing.
- **Prefixes/Suffixes:** If you need to distinguish synced groups in GCP, consider adding a prefix or suffix (e.g., `ad-developers` in GCP for the "Developers" group in AD).

3. Roles and Permissions

- **AD roles are NOT synced:** AD group memberships are synced, but the *roles* those groups have in AD don't directly translate to GCP roles.
- **Assign GCP roles:** Once groups are in GCP, you assign them GCP IAM roles (admin, editor, viewer, or custom roles) just like you would any other GCP group.
- **Example:** Your "AD-DataAnalysts" group in GCP might be granted the `BigQuery Data Viewer` role.

4. Admin, Editor, Viewer in AD vs. GCP

- **Conceptual overlap, but different implementation:** The *ideas* of admin, editor, and viewer are similar, but the *specifics* differ between AD and GCP:
  - **AD:** These often relate to permissions on domain resources (files, folders, printers, etc.).
  - **GCP:** These relate to cloud resources (Compute Engine, Cloud Storage, etc.).
- **Don't assume equivalence:** An "AD Admin" doesn't automatically become a "GCP Organization Admin." You need to explicitly grant the appropriate GCP roles.

5. Authoritative Source

- **AD is the source of truth for group membership:** Changes to groups in AD (adding/removing users) should be reflected in GCP after sync.
- **GCP is authoritative for roles:** You assign GCP IAM roles *in GCP*, not in AD.

6. Fine-Tuning with Deny Rules and Custom Roles

- **Deny rules in GCP:** Even if a user inherits broad access from an AD group, you can use *deny rules* in GCP to restrict specific permissions.
- **Custom roles:** If predefined GCP roles don't perfectly match your needs, create custom roles to tailor permissions for your AD-synced groups.

Example: Putting it all together

1. You have an AD group "AppDevs" with developers who need access to a GCP project.
2. GCDS syncs this to a GCP group "ad-appdevs."
3. In GCP, you grant "ad-appdevs" the `Editor` role on the project.
4. You *also* create a deny rule in GCP preventing "ad-appdevs" from deleting Cloud SQL instances.
5. Now, those developers can manage most project resources (due to the `Editor` role synced from AD) but CANNOT delete databases (due to the GCP deny rule).

This multi-layered approach gives you fine-grained control over how your AD groups interact with GCP permissions.

Terraform

```
# Define the folders
resource "google_folder" "folder1" {
  display_name = "Folder1"
  parent       = "organizations/YOUR_ORGANIZATION_ID"
}
```

```
resource "google_folder" "folder2" {
  display_name = "Folder2"
  parent       = "organizations/YOUR_ORGANIZATION_ID"
}

# Define the groups
resource "google_group" "admin_group" {
  email = "admin-group@yourdomain.com"
}

resource "google_group" "viewer_group" {
  email = "viewer-group@yourdomain.com"
}

# Add the user to the groups (replace with your user's email)
resource "google_group_member" "admin_group_member" {
  group = google_group.admin_group.email
  email = "user@example.com"
}

resource "google_group_member" "viewer_group_member" {
  group = google_group.viewer_group.email
  email = "user@example.com"
}

# Grant Folder Admin role to the admin group on Folder1
resource "google_folder_iam_binding" "folder1_admin_binding" {
  folder = google_folder.folder1.name
  role    = "roles/resourcemanager.folderAdmin"
  members = ["group:${google_group.admin_group.email}"]
}

# Grant Folder Viewer role to the viewer group on Folder2
resource "google_folder_iam_binding" "folder2_viewer_binding" {
  folder = google_folder.folder2.name
  role    = "roles/resourcemanager.folderViewer"
  members = ["group:${google_group.viewer_group.email}"]
}
```

Explanation:

- **Folders:** This defines two folders (`folder1` and `folder2`) within your organization. Replace `YOUR_ORGANIZATION_ID` with your actual organization ID.
- **Groups:** This defines two groups (`admin_group` and `viewer_group`). Replace `yourdomain.com` with your domain.
- **Group Members:** This adds the user (`user@example.com`) to both groups.
- **IAM Bindings:**

- `folder1_admin_binding`: Grants the `Folder Admin` role to `admin_group` on `folder1`.
- `folder2_viewer_binding`: Grants the `Folder Viewer` role to `viewer_group` on `folder2`.

How it works:

- The user (`user@example.com`) inherits the `Folder Admin` role on `folder1` by being a member of `admin_group`.
- The same user also inherits the `Folder Viewer` role on `folder2` by being a member of `viewer_group`.
- This effectively gives the user admin access to `folder1` and its contents, and read-only access to `folder2` and its contents.

Important Notes:

- **Replace placeholders:** Make sure to replace the example values (organization ID, domain, user email) with your actual values.
- **Terraform state:** Terraform will manage the state of these resources. Make sure you have a proper Terraform backend configured.
- **Dependencies:** Ensure that the groups are created before adding members or granting roles. Terraform should handle this automatically based on the resource dependencies.
- **AD synchronization:** This example focuses on the GCP side. You'll still need to configure your AD group synchronization (using GCDS or an IdP) to keep the groups in sync.

2.4 Organizational Policies

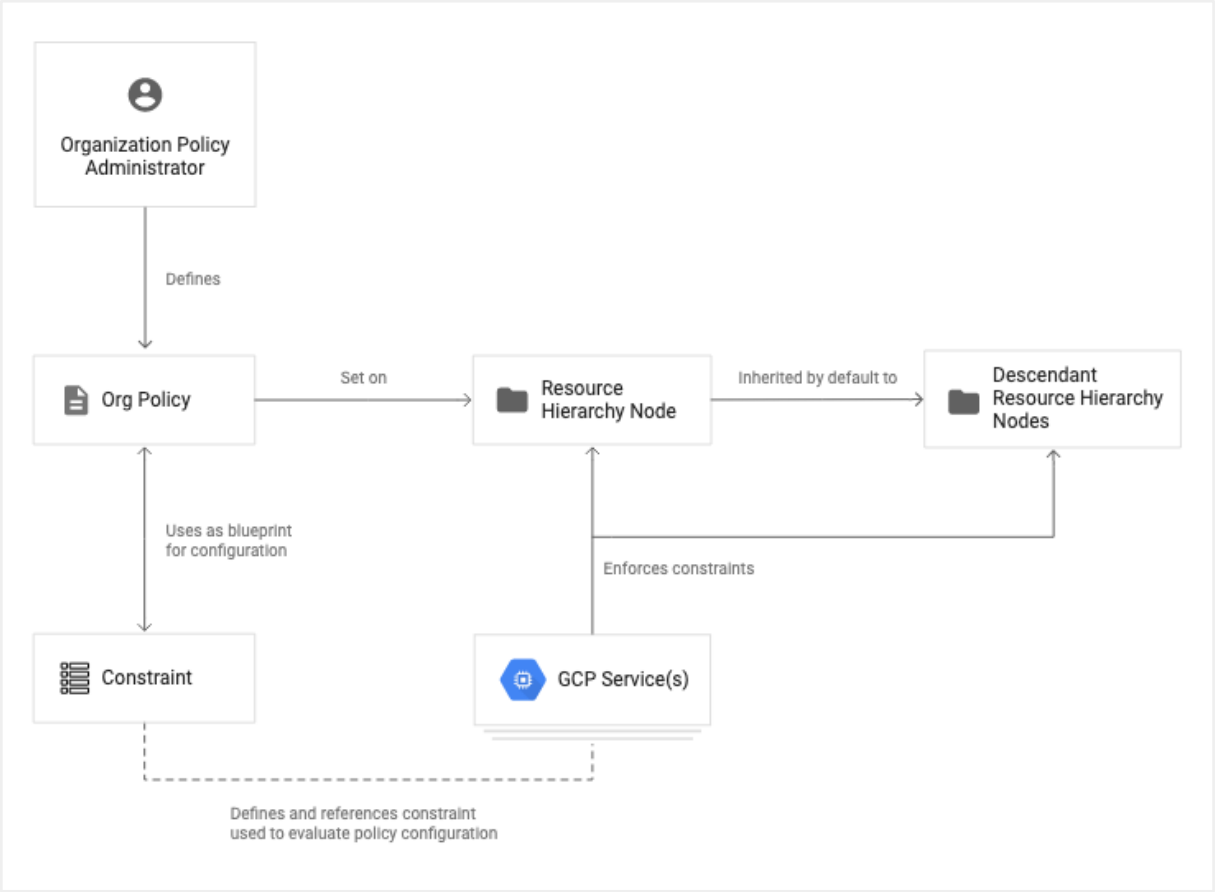
Quest Diagnostics is requesting a list of best practice organization policies that go beyond the default setup and enforce additional controls which are relevant to a HIPAA compliant environment.

With the secure-by-default organization policy enforcements, insecure postures are addressed with a bundle of organization policies that are enforced at the time of creation of an organization resource. Examples of these enforcements include disabling service account key creation and disabling service account key upload.

When an existing user creates an organization, the security posture for the new organization resource might be different from the existing organization resources. Secure-by-default organization policies are enforced for all organizations created on or after May 3, 2024.

These default organizational policies are best practice for HIPAA environment foundations, provided the additional relevant controls in the shared responsibility model are adhered to.

An organization policy is a configuration of constraints, which can be set on a resource hierarchy node (i.e. a project, a folder, or at the top/root level) to enforce the restrictions defined in the constraints on the node and its descendants.



Key organization policy concepts

Note that policies are **not** retroactive, meaning existing resources, even if they violate the new policies, will continue to run and function and manual intervention will be required.
If a new organization policy sets a restriction on an action or state that a service is already in, the policy is considered to be in violation, **but the service will not stop its original behavior**. You will need to **address this violation manually**. This prevents the risk of a new organization policy completely shutting down your business continuity.

You can find organization policy violation details in the Google Cloud Console on the **Monitoring** page within the **Compliance** section, and you can configure alerts for violations using **Cloud Monitoring**.

Organization policy vs Identity and Access Management

Organization policies relate to a different layer of management than IAM policies. Organization policies allow an administrator to set restrictions on how specific resources can be configured, IAM focuses on who can act on particular resources based on permissions.

Constraints

A constraint is a particular type of restriction against a Google Cloud service or a list of Google Cloud services. The Google Cloud service mapped to that constraint and associated with a resource hierarchy node will then enforce the restrictions configured within the organization policy.

A constraint has a type, either **list** or **boolean**:

- List constraints evaluate the constraint with a list of allowed or denied values that you provide, such as an allowlist of IP addresses that can connect to a virtual machine.
- Boolean constraints are either enforced or not enforced for a given resource, and govern a specific behavior, such as whether external service accounts can be created.

Recommended policies

The full list of currently available constraints can be found here. At this time, the recommendation is to enable the following organization policies. This is in addition to the organization policy constraints that are automatically enforced when you create an organization resource. More policies could be added/removed depending on the requirement as the journey progresses.

Constraint in IAM > Organization Policies	Resources (Org/Folder)	Allow / Deny	Status	Config.	Needed?
compute.vmExternalIpAccess	Org	Allow only a whitelist of compute resources to have an external IP	Allow All	Default setting	Enable
iam.allowedPolicyMemberDomains	Org	Defines the member domains that can be added to Cloud IAM policies	Allow All	Default setting	Enable
compute.skipDefaultNetworkCreation	Org	To skip the creation of the default network during project creation	Not Enforced	Default setting	Enable
gcp.resourceLocations	Org	Defines the set of locations where location-based GCP resources can be created.	Allow All	Default setting	Enable

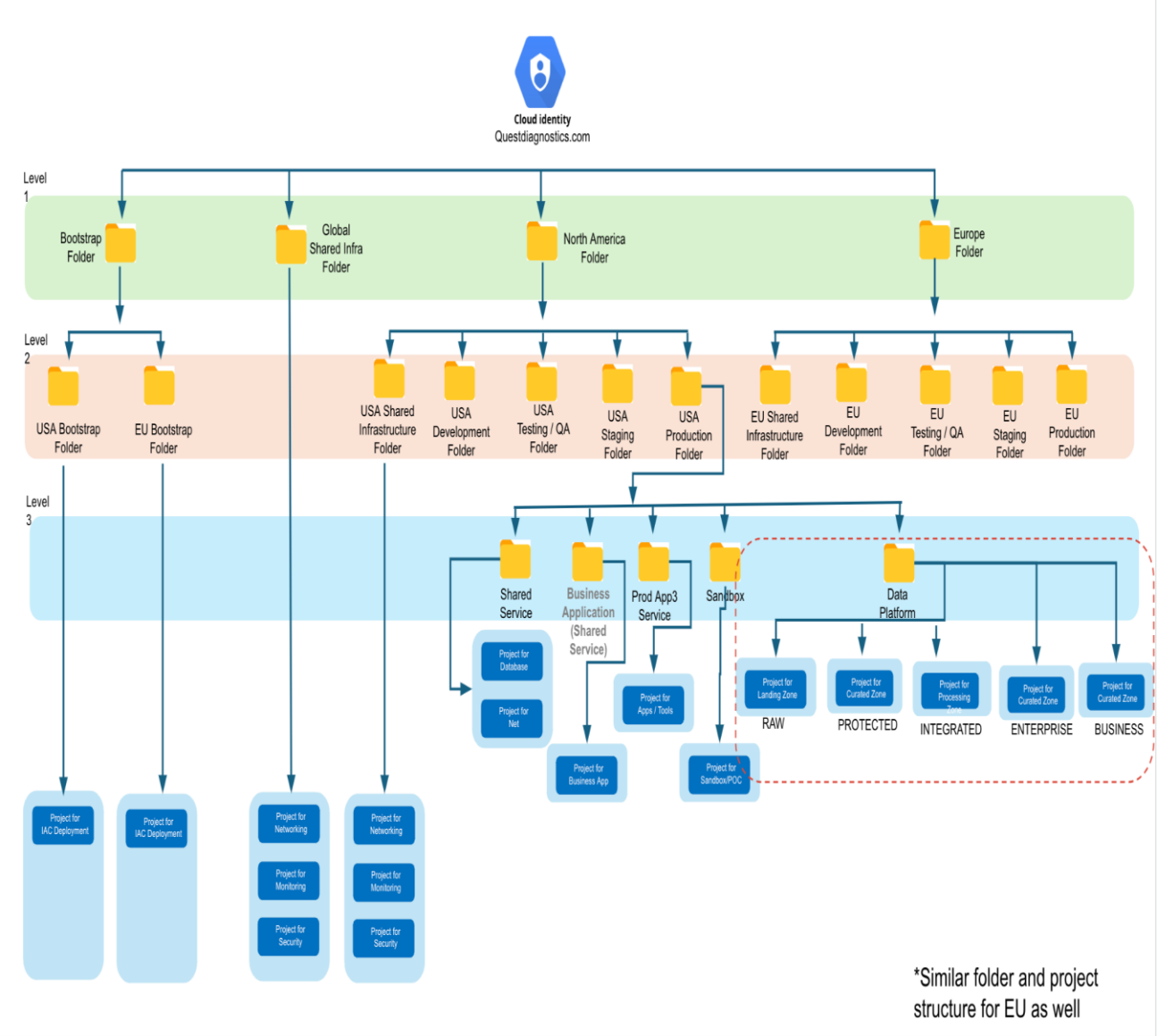
compute.trustedImageProjects	Org	Defines the set of projects that can be used for image storage and disk instantiation for Compute Engine	Allow All	Default setting	Enable
storage.uniformBucketLevelAccess	Org	Allow only IAM policies to grant access to objects in GCS buckets, not bucket ACLs	Not Enforced	Default setting	Enable
compute.restrictXpnProjectLienRemoval	Org	Restricts the set of users that can remove a Shared VPC project lien	Not Enforced	Default setting	Enable
compute.restrictDedicatedInterconnectUsage (or PartnerInterconnectUsage)	Org	Defines the set of VPCs that are allowed to use Dedicated or Partner Interconnect	Allow All	Default setting	Enable
compute.restrictSharedVpcHostProjects	Org	Defines the set of Shared VPC host projects that projects at or below this resource can attach to.	Allow All	Default setting	Enable
iam.automaticIamGrantsForDefaultServiceAccounts	Org	Prevents the App Engine & Compute Engine default service accounts from being automatically granted an IAM role on their project	Not Enforced	Default setting	Enable
sql.restrictPublicIp	Org	Prevent Cloud SQL from being configured with a public IP	Not Enforced	Default setting	Enable
constraints/compute.disableSerialPortAccess	Org	Disables serial port access to Compute Engine VMs belonging to the organization, project, or folder where this constraint is set to True.	Not Enforced	Default setting	Enable
gcp.restrictCmekCryptoKeyProjects	Org	This policy restructs Cloud KMS keys that you can use to protect a resource in a Filestore project.	Not Enforced	Optional	Additional HIPAA Policy
gcp.restrictNonCmekServices	Org	This list constraint defines which services require Customer-Managed Encryption Keys (CMEK). Setting this constraint to Deny (i.e. deny resource creation without CMEK) requires that, for the specified services, newly created resources must be protected by a CMEK key.	Not Enforced	Optional	Additional HIPAA Policy
compute.allowedVlanAttachmentEncryption	Org	This list constraint defines the allowed encryption settings for new VLAN Attachments. By default, VLAN Attachments are allowed to use any encryption settings. Set IPSEC as the allowed	Not Enforced	Optional	Additional HIPAA Policy

		value to enforce creating encrypted VLAN attachments only.			
essentialcontacts.disableProjectSecurityContacts	Org	This boolean constraint, when enforced, allows organization policy administrators to ensure that only contacts assigned at the organization or folder level can receive security notifications.	Not Enforced	Optional	Additional HIPAA Policy

3. Resource management

Resource management design aims to organize, name, and set quotas of cloud resources to ensure a structured, consistent, and controlled environment.

3.1 Organization hierarchy



Questdiagnostics.com (Organization)

├── **Bootstrap** (gcp-boot) **(TF State files, Projects IaC deployment)**

```
| └── USA Bootstrap Folder (gcp-boot-us) ( State files) Service Account(SA) ONLY for USA
      └── (prj-boot-iac-us-4000)(Projects for IaC deployment, scripts, and documentation)
| └── EU Bootstrap Folder (gcp-boot-us) ( State files) SA ONLY for EU
      └── (prj-boot-iac-eu-4300)(Projects for IaC deployment, scripts, and documentation)
├── Global Shared Infrastructure (gcp-shrd) (Infra Globally (i.e. AD,DNS, NET) and Region) specific)
| ├── Networking (prj-shrd-ntwk-4311)(Project for VPCs, Firewalls, etc.)
| ├── Monitoring (prj-shrd-mntr-4312)(Project for centralized logging and monitoring)
| └── Security  (prj-shrd-secu-4313)(Project for security tools, SIEM, etc.)
|
├── North America Folder (gcp-us) (Region/Business Unit/Subsidiary)
| ├── USA Shared Infrastructure (gcp-shrd-infra-us) (Infra Globally (i.e. AD,DNS, NET) and Region) specific)
| ├── Networking (prj-shrd-ntwk-us-4113) (Project for VPCs, Firewalls, etc.)
| ├── Monitoring (prj-shrd-mntr-us-4613) (Project for centralized logging and monitoring)
| └── Security (prj-shrd-secu-us-4313)(Project for security tools, SIEM, etc.)
| ├── Development Folder (gcp-dev-us)
| | ├── Shared Dev-Infra Services Folder (gcp-d-shrd-serv-us)(Dev-specific DB, Net)
| | ├── Busines Application Shared Services Folder (gcp-d-buapp-shrd-serv-us)
| | | └── Dev-BA-app-A shared project (prj-d-shrd-serv-baapp-us-5432) (i.e GKE Shared Cluster)
| | | └── Dev-App1 (Namespace 1)
| | | └── Dev-App2 (Namespace 2)
| | └── Dev-App3 Service Folder (apps/tools) (gcp-d-app3-shrd-serv-us)
| | | └── Dev-App3 project (Resources) (prj-d-app3-sql31-us-5402)
| | └── Sandbox Folder (gcp-d-sdb-us)
```

```
│ │       └── Development (Sandbox Project for development)  (prj-d-sdb-test-us-5552)
│ ├── Testing/QA Folder
│ │ ├── Shared Infra Services Folder (Test-specific i.e. DB, Net)
│ │ ├── Busines Application Shared Services Folder
│ │ │ └── Test-BA-app-A shared project (i.e GKE Shared Cluster)
│ │ │ └── Test-App1 (Namespace 1)
│ │ │ └── Test-App2 (Namespace 2)
│ │ └── Test-App3 Service Folder (apps/tools)
│ │ │ └── Test-App3 project (Resources)
│ │ └── Sandbox Folder
│ │ └── Development (Sandbox Project for development)
│ ├── Stage Folder
│ │ ├── Shared Services Stage Folder(Project Stage-specific i.e DB, Net)
│ │ ├── Busines Application Shared Services Folder

│ │ │ └── Stage-BA-app-A shared project (i.e GKE Shared Cluster)
│ │ │ └── Stage-App1 (Namespace 1)
│ │ │ └── Stage-App2 (Namespace 2)
│ │ └── Stage-App3 Service Folder (apps/tools)
│ │ │ └── Stage-App3 project (Resources)
│ │ └── Stage-Sandbox/POC Folder ( Exception process)
│ │ └── Development (Sandbox Project for development)
│ ├── Production Folder
│ │ ├── Shared Services Prod Folder (Project Prod-specific i.e. DB, Net)
│ │ ├── Busines Application Shared Services Project
│ │ │ └── Prod-BA-app-A shared project (i.e GKE Shared Cluster)
│ │ │ └── Prod-App1 (Namespace 1)
│ │ │ └── Prod-App2 (Namespace 2)
│ │ └── Prod-App3 Service Folder (apps/tools)
│ │ │ └── Prod-App3 project (Resources)
```

```
│ │ └── Prod-Sandbox/POC Folder ( Exception process)
│ │ └── Prod-Sandbox/POC Project (Sandbox Project for development)
│ │
├── EU Folder (Region/Business Unit/Subsidiary)
├── EU Shared Infrastructure (Infra Globally (i.e. AD,DNS, NET) and Region)
specific)
│ ├── Networking (Project for VPCs, Firewalls, etc.)
│ ├── Monitoring (Project for centralized logging and monitoring)
│ └── Security (Project for security tools, SIEM, etc.)
│ ├── Development Folder
│ │ ├── Shared Infra Services Folder(Dev-specific DB, Net)
│ │ ├── Busines Application Shared Services Folder
│ │ │ └── Dev-BA-app-A shared project (i.e GKE Shared Cluster)
│ │ │ └── Dev-App1 (Namespace 1)
│ │ │ └── Dev-App2 (Namespace 2)
│ │ └── Dev-App3 Service Folder (apps/tools)
│ │ │ └── Dev-App3 project (Resources)
│ │ └── Sandbox Folder
│ │ └── Development Project (Sandbox Project for development)
│ ├── Testing/QA Folder
│ │ ├── Shared Infra Services Folder (Test-specific i.e. DB, Net)
│ │ ├── Busines Application Shared Services Folder
│ │ │ └── Test-BA-app-A shared project (i.e GKE Shared Cluster)
│ │ │ └── Test-App1 (Namespace 1)
│ │ │ └── Test-App2 (Namespace 2)
│ │ └── Test-App3 Service Folder (apps/tools)
│ │ │ └── Test-App3 project (Resources)
│ │ └── Sandbox Folder
│ │ └── Development Project (Sandbox Project for development)
│ ├── Stage Folder
```

```
│ │ ├── Shared Services Stage Folder(Project Stage-specific i.e DB, Net)

│ │ ├── Busines Application Shared Services Folder
│ │ │ └── Stage-BA-app-A shared project (i.e GKE Shared Cluster)
│ │ │ └── Stage-App1 (Namespace 1)
│ │ │ └── Stage-App2 (Namespace 2)
│ │ └── Stage-App3 Service Folder (apps/tools)
│ │ │ └── Stage-App3 project (Resources)
│ │ └── Stage-Sandbox/POC Folder ( Exception process)
│ │ └── Development Project (Sandbox Project for development)
│ ├── Production Folder
│ │ ├── Shared Services Prod Folder (Project Prod-specific i.e. DB, Net)
│ │ ├── Busines Application Shared Services Project
│ │ │ └── Prod-BA-app-A shared project (i.e GKE Shared Cluster)
│ │ │ └── Prod-App1 (Namespace 1)
│ │ │ └── Prod-App2 (Namespace 2)
│ │ └── Prod-App3 Service Folder (apps/tools)
│ │ │ └── Prod-App3 project (Resources)
│ │ └── Prod-Sandbox/POC Folder ( Exception process)
│ │ └── Prod-Sandbox/POC Project (Sandbox Project for development)
```

Explanation and Key Considerations:

1. **questdiagnostics.com (Organization):** This is the root node representing your company in GCP.
2. **Bootstrap**: A dedicated folder for IaC. This ensures a clean and controlled environment setup using Terraform.
3. **Shared Infrastructure:** Centralized services used across all environments. This includes networking, monitoring, and security tools.
4. **Regions (North America, Europe):** Separate folders for each region, ensuring geographic isolation for compliance, performance, and localized management. This enables adherence to local regulations (e.g., GDPR) and optimizes resource placement for end-users in each region.

5. **Environment-Based Folders (Development, Testing, Production):** Distinct folders for each environment within regions, ensuring isolation and enforcing environment-specific policies and IAM roles for security and compliance.
6. **Shared Services within Environments:** This allows for environment-specific shared services, like test databases or development tools.
7. **Department/Team Structure**: Folders mirroring Quest Diagnostics' organization (Marketing, Engineering) with sub-folders for teams (Social Media, Web Development). This promotes clear ownership and responsibility.
8. **Project Placement**: Projects are created within the folders relevant to their function and environment.

3.2 Folder Naming

The folder naming convention is flexible, but it is recommended to use names that are concise yet descriptive. Folder names can be reused within the tree. The syntax should be all lowercase with hyphen-separated fields as appropriate.

Naming Convention: gcp-<environment(if any)>-<foldername>-<region>
Examples:

- Development Folder: gcp-dev-us
- Business Application Shared Services Folder: gcp-d-buapp-shrd-serv-us
- North America Folder: gcp-us

Explanation:

For higher-level folders that do not specify an environment or business unit, use a generic format such as **gcp-us** to represent all USA-related folders and projects.For nested folders, include the environment and business unit in the name. For example, **gcp-d-buapp-shrd-serv-us** indicates a folder for the Development environment (d) and Business Application Shared Services (buapp-shrd-serv).This structure ensures clarity and scalability while maintaining consistency across the folder hierarchy.

3.3 Project Naming

Projects generally fall into two categories:

- Global projects that have functionality across all of GCP.
- Product projects that are within a specific business portfolio.

There are some constraints in regards to project names, in particular:

- Project name space is global within the organization. In particular, two projects can not have the same name even if they are under different folders.

- It is convenient for the project ID to be similar to the project name - typically with a suffix of random numbers. That in turn involves the constraints that they must be 6 to 30 lowercase letters, digits, or hyphens.
- Also see: REST Resource: projects

Naming convention: "prj-business-code-environment-code{-project-label}-index
Example: prj-acde-p-shared-base-1"

4. Access management

Access management design ensures that only the right people/services are authorized to perform the right actions on the right resources.

IAM policies are available for configuration in the Google Cloud Console or through IaC. IAM roles are available for users, groups of users, and service accounts that allow granular control of permissions to access resources. The organization resource provides a way to unify all projects under a single organization with permission inheritance across the organization. Project organization and hierarchy choices are key to a successful program.

4.1 Service accounts

A service account is utilized by applications and virtual machines to authenticate their access to Google Cloud APIs. This specialized account has its own identity, and applications leverage its credentials to authorize their interactions with a designated set of APIs. The actions that these applications can perform are governed by the permissions that have been specifically assigned to the service account.

Quest will be using service accounts extensively and will follow the general best practices that:

- Roles should be by least privileges and scopes should be narrow.
- This is discussed in more detail here Next 2017 video.

Generally service accounts are used in three ways:

- Used by applications directly that need the service account's privileges.
- Used by tooling such as terraform or spinnaker in order to provision resources or deploy applications onto those resources.

4.1.2 Default service accounts

When you enable or use some Google Cloud services, they create user-managed service accounts that enable the service to deploy jobs that access other Google Cloud resources.

These Service Accounts are known as default service accounts and primarily exist for new, inexperienced users of the platform who are just beginning their GCP journey.

The following table lists the services that create default service accounts:

Service	Service Account Name	Email
App Engine, and any Google Cloud service that uses App Engine	App Engine Default Service Account	{project-id}@appspot.gserviceaccount.com
Compute Engine, and any Google Cloud service that uses Compute Engine (GKE, Dataflow, etc)	Compute Engine Default Service Account	{project-number}-compute@developer.gserviceaccount.com

Default Service Accounts

When a default service account is created, it is automatically granted the Editor role (roles/editor) on your project. This role includes a very large number of permissions. To follow the principle of least privilege, Google strongly recommends that Quest Diagnostic's disable the automatic role grant by adding the Disable Automatic IAM Grants for Default Service Accounts constraint to the organization policy. This Organization Policy will ensure that the Editor role is not automatically granted the editor role upon API enablement.

4.1.3 Service accounts key types

Each service account is associated with two sets of public/private RSA key pairs that are used to authenticate to Google:

- Google-managed keys
- User-managed keys

When service accounts are used inside GCP, their keys don't need to be downloaded, and they are managed by GCP (Google managed keys). When service accounts are used outside of GCP, their keys must be downloaded and managed externally (user managed keys). It is recommended to use Google-managed keys wherever possible to reduce the operational burden and associated risk with managing long term credentials. When a user-managed key is required, we recommend implementing a process to generate dynamic credentials and automate the rotation of the credential.

Service Account Key Type	Description
Google Managed Keys	● Google stores both the public and private portion of the key. ● Key rotation is handled by Google (signing a maximum of two weeks per key). ● Google holds private keys in escrow (never directly accessible). ● IAM provides APIs to use these keys to sign on behalf of the service account.
User Managed Keys	● Quest Diagnostic's owns both the public and private portions of a key pair. ● Quest Diagnostics can initiate the generation of one or more user-managed key pairs (also known as "external" keys) that can be used from outside of Google Cloud. ● Google only stores the public portion of a user-managed key. ● Quest Diagnostics is responsible for the security of the private key and other management operations such as key rotation (up to 10 service account keys per service account to facilitate key rotation). Audit logs will show the specific key that is used for authentication as detailed in the following documentation. ● Service account keys expire in the year 9999. ● User-managed keys can be managed by the IAM API, gcloud command-line tool, or the service accounts page in the Google Cloud Console.

Quest Diagnostics have decided that for some developers, they will restrict the use of persistent service account keys with the following security approaches:

- To prevent developers from creating and downloading persistent credentials, they will configure the organization policy constraint constraints/iam.disableServiceAccountKeyCreation on certain projects and folders

4.1.4 Service account use cases

Google Cloud services can be accessed in a number of ways. However, service account keys are not always necessary. In fact, there are often more secure authentication methods

available. It is recommended exploring alternatives to service account keys whenever possible.

If you're using the Google Cloud CLI, Cloud Client Libraries, Terraform, or REST requests to interact with Google Cloud services, refer to the following diagram to select the most appropriate authentication method for your needs.

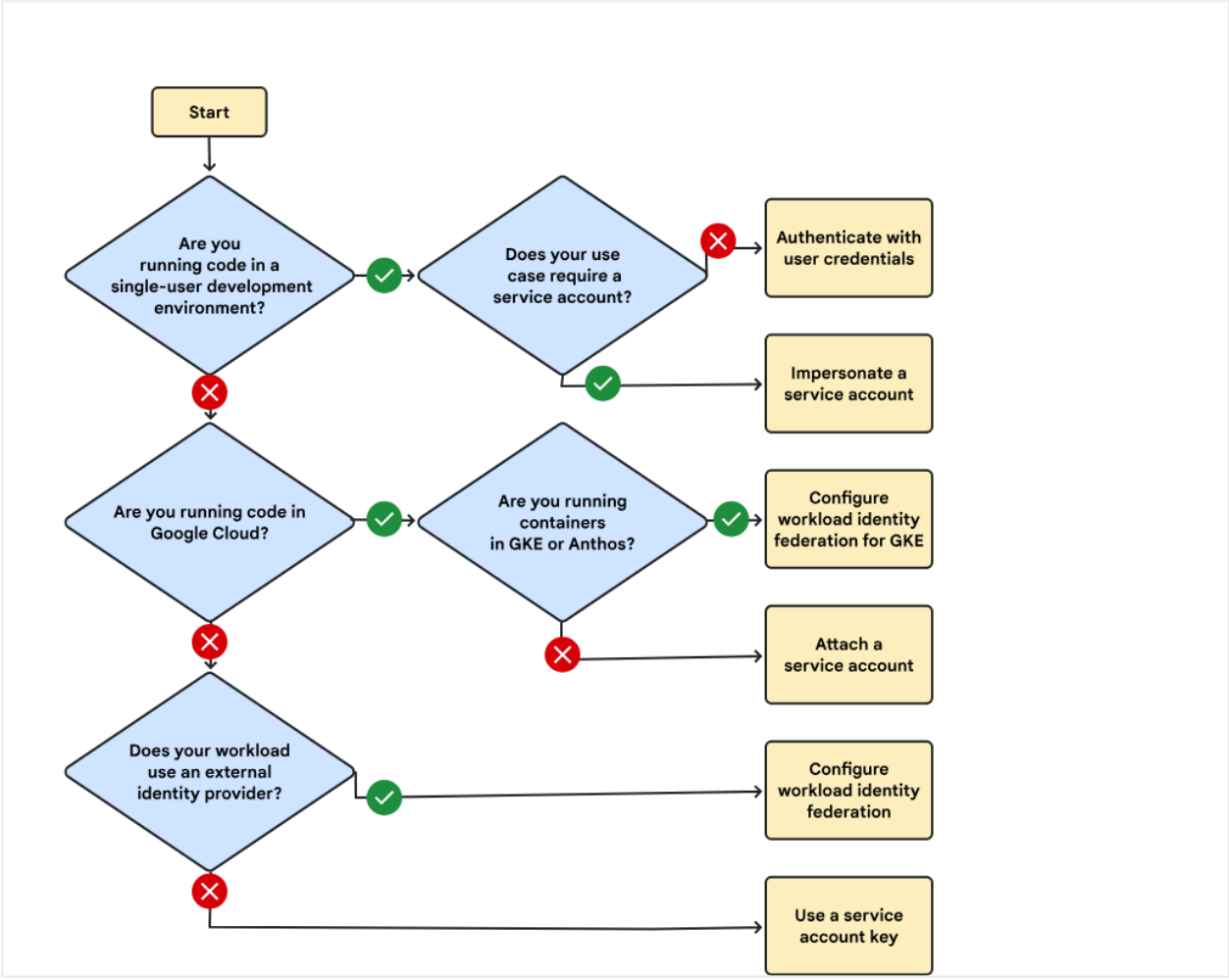
Service account Rotation & Monitoring Best Practices:

Service account keys should not be used for check in and check out access due to their long-lived nature and security risks. We recommend a multi-layered approach to service account creation & rotation security leveraging organization policies. Workload Identity Federation can also be used to establish trust between your identity provider and Google Cloud, enabling workloads to obtain temporary credentials. If this isn't feasible, use short-lived credentials or a secrets management solution as suggested in the above sections.

To alert service account owners to rotation changes, we suggest the use of Cloud Monitoring and Cloud Logging. Create custom metrics and alerts in Cloud Monitoring to track key rotation and expiration. Log key rotation events and failed attempts in Cloud Logging, triggering alerts based on specific criteria. Automate key rotation and owner notifications using tools like Cloud Functions.

Regularly audit service accounts and keys using IAM Recommender. IAM Recommender analyzes IAM configurations, Forseti Security identifies IAM policy violations, and Security Command Center provides security health analytics.

Utilize the serviceAccount.keys.list method to programmatically check key metadata and the Logs Viewer to filter logs for key activity.



How to use service accounts decision tree

Service Account Naming and Documentation Convention

To help track the association between a service and an application or resource, follow a naming convention when creating new service accounts:

- Add a prefix to the service account email address that identifies how the account is used. For example:
  - vm- for service accounts attached to a VM instance.
  - wlifgke- for service accounts used by Workload Identity Federation for GKE.
  - wlif- for service accounts used by Workload Identity Federation.
  - onprem- for service accounts used by on-premises applications.
- Embed the name of the application in the service account email address, for example: vm-travelexpenses@ if the VM runs a travel expenses application.

- Use the description field to add a contact person, links to relevant documentation, or other notes.
- Have a centralized repository for service accounts, they can reside in a single project so you have centralized control over them, they can have access to other projects, resources but the management will be centralized.
- SA keys should never be embedded in code or committed to any external source repository and also scan external repositories for keys.
- Use Workload Identity Federation instead of SA keys where possible.
- Rotate user-managed service account keys frequently (15, 30 or even 60 days depending on your Org security policy). Rotating keys will reduce the window of opportunity for a key that is associated with a compromised account to be used. Keys should be rotated to ensure that data cannot be accessed with an old key which might have been lost, cracked, or stolen. Even after the owner's precaution, keys can be easily leaked by common development malpractices like checking keys into the source code or leaving them in Downloads directory, or accidentally leaving them on support blogs/channels.
- Regularly audit service accounts and keys via IAM tools and usage metrics.
- Only allow service account permissions like impersonation, creation etc. through IAM roles to authorized principals, Anyone who has access to the keys will be able to access resources through the service account.
- Use dedicated, custom service accounts for running VMs, with minimum permissions required. In general, the service accounts should be mapped to specific application components (for example, fronted, application server, database) and per environment (Prod, QA, Dev), to limit lateral movement in case a service account gets compromised.
- Use service accounts to apply firewall rules, where possible (for example, for VM based applications running on GCP).
- Do not embed secrets related to authentication in source code, such as service account credentials, API keys, OAuth tokens. You can use an environment variable pointing to credentials outside of the application's source code. That environment variable is **NOT** the credential itself, but a reference to point to the credential itself.
- As a general guidance, service accounts should have no more than 2 active user managed keys.
- GCP-managed keys are used by Cloud Platform services such as App Engine and Compute Engine. These keys cannot be downloaded. Google will keep the keys and automatically rotate them on an approximately weekly basis.
- User-managed keys are created, downloadable, and managed by users. For user-managed keys, User have to take ownership of key management activities which includes:
    - Key storage
    - Key distribution

- Key revocation
- Key rotation
- Protecting the keys from unauthorized users
- Key recovery

4.3 IAM Groups, roles and scopes

Quest Diagnostics has already created several groups under their GCP organization that will be used for segmentation of tasks in the cloud including org-admins, network-admins, etc. Remember that the purpose of these groups are to align to the principle of least privilege, where we are only granting the appropriate permissions that people need to have to do their jobs and no more permissions than that.
The best practice is to have a focused dedicated document that has all the group names and the IAM roles assigned to each one of those groups at different levels such as Org, Folder or Project, in order to have documented control over all this information and make it easy for people to audit all these permissions in the cloud directly from one document that serves as a single source of truth for IAM.

In case you want to add roles to your different principals including service accounts, groups, etc. please remember that the following page describes IAM roles and lists the predefined roles that you can grant to your principals.

- Understanding roles

You can also enter the IAM roles available for each service, for example:

- IAM roles and permissions for Compute Engine
- IAM roles and permissions for Cloud Storage

4.4 Access Auditing Tools

There are several tools available for auditing IAM, the main tools are:

- Policy Troubleshooter for IAM permissions
- Review and apply role recommendations
- Policy Analyzer for IAM
- Asset Inventory
    - Introduction to Cloud Asset Inventory
    - Google Cloud Asset Inventory video

- Policy Simulator:
  - Policy Simulator overview
  - Test role changes with Policy Simulator
- Policy Intelligence Tools
  - Policy Intelligence tools
  - Using Policy Intelligence to achieve least privilege access

SCC Event Threat Detection produces security findings by matching events in the Cloud Logging log streams to known indicators of compromise (IoC). IoCs, developed by internal Google security sources, identify potential vulnerabilities and attacks. Event Threat Detection also detects threats by identifying known adversarial tactics, techniques, and procedures in your logging stream, and by detecting deviations from past behavior of Quest's organization or projects. If you Security Command Center Premium tier at the organization level, Event Threat Detection can also scan Google Workspace logs.

IAM Policy Analyzer relies on the Cloud Asset API to function. Because Cloud Asset Inventory utilizes and therefore enables the Cloud Asset API, enabling Cloud Asset Inventory will automatically provide access to IAM Policy Analyzer. You can consider IAM Policy Analyzer a downstream feature of Cloud Asset Inventory, as its functionality is dependent on the API that Cloud Asset Inventory enables.

5. Networking

The purpose of networking design is to enable communications between Google Cloud resources, Quest's on-premises data centers and third-party clouds. Networking design includes both connection and segregation/protection of resources.

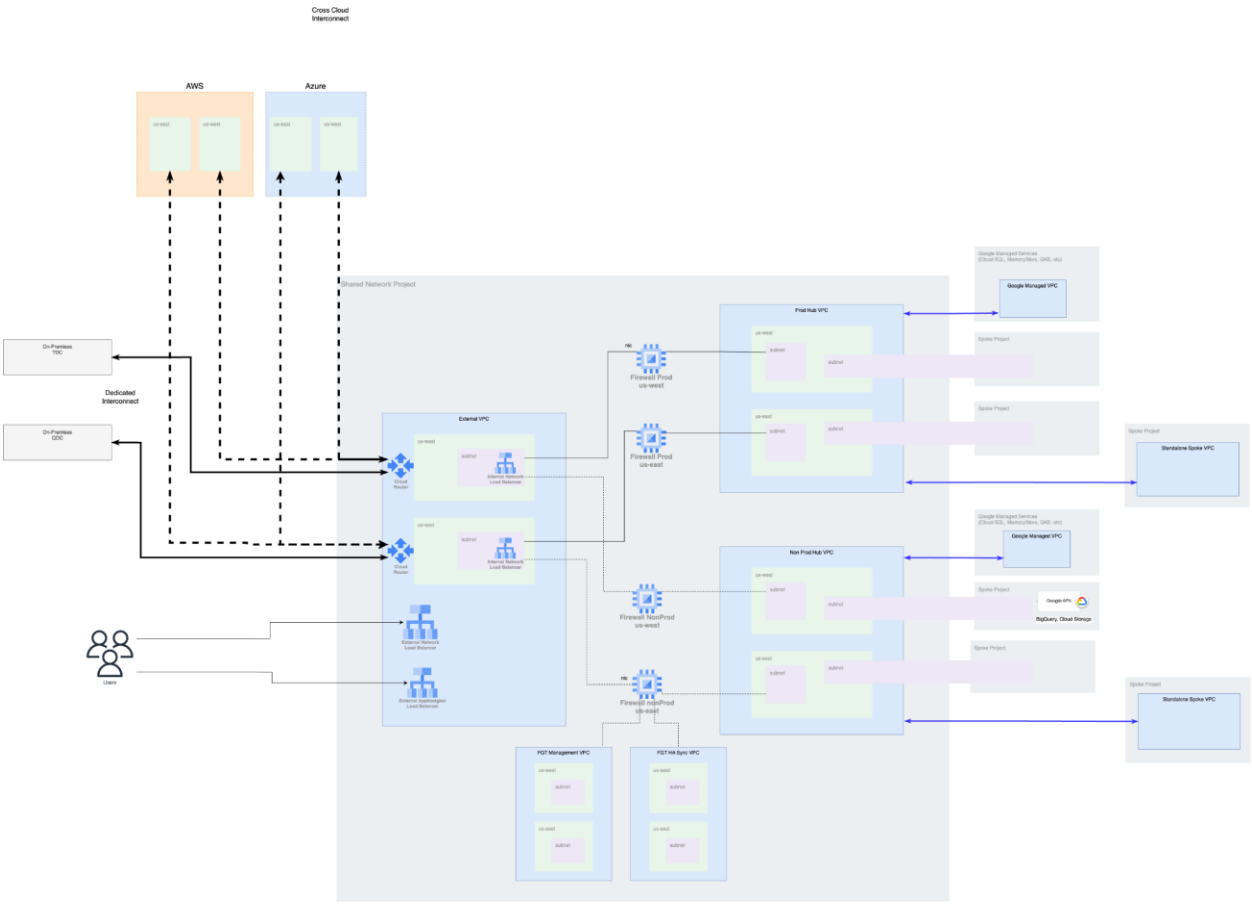
5.1 Requirements

Quest is building its Google Cloud Foundations and presented the following network and communication flow requirements:

- Support for Google Cloud Resources in two US regions, and possibility in the future of using a region in Europe
- Dedicated high capacity and low latency connection from Google Cloud to its two data Centers in US
- High capacity connections from Google Cloud to Quest's environments in AWS and Azure
- All connections between Google Cloud and external environments to be secure and encrypted
- Use of Fortinet NGFW appliances for traffic inspection
    - Inspect traffic between OnPrem and Google Cloud
    - Inspect traffic between other Cloud providers and Google Cloud
    - Inspect traffic between Internet and Google Cloud
    - Inspect traffic between non production and production Google Cloud environments
    - Possibility of having traffic inspection between network segments within Google Cloud
- Internal DNS naming resolution between all environments
- All the network traffic within Google Cloud should be logged
- Applications exposed externally should be protected by a WAF native solution
- CDN solutions should used to serve content closer to end users

5.2 High Level Design

Quest is looking to connect its Google Cloud environment network to all other environments: AWS, Azure and On-Premises. The following picture presents a high level overview of the proposed network design.



Source:
[GCP_Foundations_Visio.vsdx]
[Quest-VPC Standalone.vsdx]
[QuestNetworkDiagram.pdf]

At the center of this design sits the dedicated Fortinet Firewall appliance that will inspect all traffic traversing VPCs and external Quest environments (on-prem, other clouds and public Internet). External Load Balancers are ure used

For the initial deployment, Quest's Landing Zone will support two regions in North America. Support for additional regions can be added later. Traffic will be split in Production and Non Production instances of the Firewall.

Quest workloads can be deployed in two types of environments: environments that don't require internal traffic inspection and environments that require internal traffic inspection.

As it will be detailed below, Interconnect links will be used to connect Quest On-Prems to Google Cloud and Cross-Cloud Interconnect will be to connect to AWS and Azure.

Two additional VPCs are required for the management interface and for the HA sync interface of the Fortinet VMs. All Fortigate VMs will have one interface in management and HA VPCs.

5.2.1 Google Cloud Regions

Quest requires two regions in the US to deploy their Google Cloud resources. Additionally, more regions might be needed in Europe.

The choice of regions should be based on the following requirements:

- User proximity: users in the US east and west. Regions that minimize latency to users.
- Data Center proximity: data centers near Philadelphia and Dallas.
- Resource availability and region health: availability of underlying resources to avoid situations of insufficient capacity, support for high demand of resources.
- Release of features: regions where new features and services are available faster, for example, Interconnect links with support for MACSec.
- Services: Quest is looking mainly for Data and AI services: BigQuery, Vertex AI, Looker, HPC compute, etc.
- Cost for running resources and services.

Based on these requirements, the following two regions are proposed:

Region	Google Cloud Regions
US East	us-east4 (Virginia)
US West	us-west1 (Oregon)

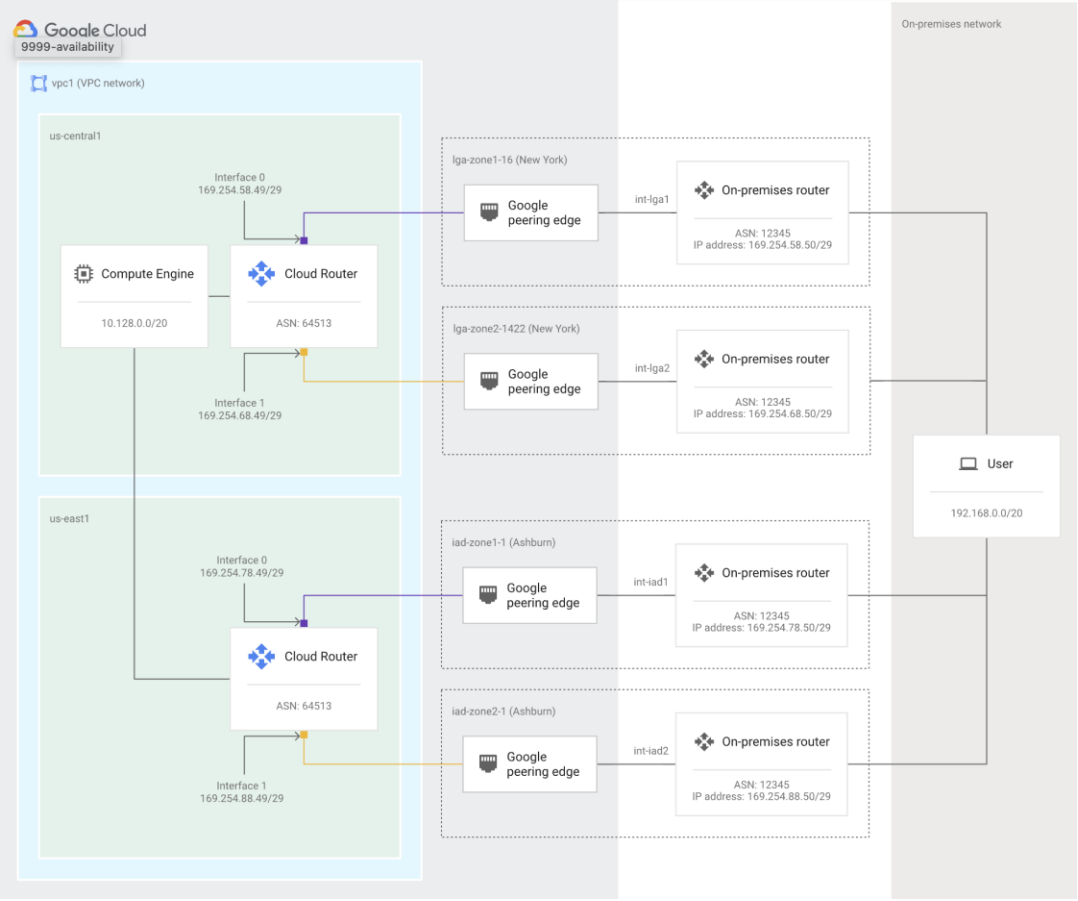
5.3 Connection to Google

5.3.1 To On-Premises

Quest requires dedicated connectivity to both QDC (Philadelphia) and TDC (Dallas) datacenters. To achieve this, Quest will use Dedicated Interconnect to provide connectivity between GCP and their on-premise infrastructure over RFC1918 address space. Dedicated Interconnect provides a consistent and reliable connection that may feature low latency and high bandwidth. Dedicated Interconnect enables transfer of large amounts of data between networks, which can be more cost effective than purchasing additional bandwidth over the public Internet or VPN tunnels.

All Interconnect links (to on-premises and other cloud providers) will be secured with MACSec encryption. MACsec for Cloud Interconnect uses IEEE standard 802.1AE Media Access Control Security (MACsec) to encrypt traffic between your on-premises router and Google's edge routers. A GCM-AES-256 connectivity association key (CAK) and a connectivity association key name (CKN) values will be created. The remote router should be configured to use the CAK and CKN values to configure MACsec. After MACsec is enabled on the remote router and in Cloud Interconnect, MACsec encrypts your traffic between your on-premises router and Google's peering edge router.

For the highest level availability, Google recommends the configuration for 99.99% availability for production-level applications, such as mission-critical operations that have a low tolerance for downtime. The following diagram depicts the recommended architecture with 4 Dedicated Interconnect links:



At least four Dedicated Interconnect connections, two connections in one metropolitan area and two connections in another metro. Connections that are in the same metro must be placed in different edge availability domains (metro availability zones) to achieve 99.99% availability. Placing Dedicated Interconnect connections in two separate edge availability domains within the same metro is important because maintenance windows are coordinated across edge availability domains within a metro, while maintenance windows are not coordinated across metros.

One Cloud Router will be created in each region, where the VLAN attachments terminate the Interconnect links. Cloud Routers will advertise routes to subnets outside of the Cloud Router's region with a lower priority compared to subnets that are in the Cloud Router's region.

An active / active configuration is preferred, since during normal operation traffic is evenly split between all Interconnects. This configuration is better for most applications for several reasons:

- If an Interconnect fails, only a portion of traffic needs to fail over to the other Interconnect(s). This results in less overall application impact.

- Running real traffic is the best way to ensure that all of the Interconnects are working correctly. It is unfortunate to first discover a misconfiguration during an actual outage.
- Due to the way Google's network is configured, running Interconnects at 50% of their max capacity is significantly more resilient to packet loss from congestion in the network.

The following table summarizes the Onprem locations that Quest wants to connect, the closest colocations and service providers available.

Quest	Address	Nearest Colocation	Service Providers
QDC	400 Egypt Rd West Norristown PA 19403	Equinix PH1 - Philadelphia	Lumen, Verizon
TDC	6431 Longhorn Dr. | Irving, TX 75063 USA	Equinix Dallas (DA1)	Lumen, Zayo

The following four Interconnect links will be created to QDC and TDC:

Quest	Metro Area	Colocation	Facility Provider	Google Cloud Region
QDC	Philadelphia	phl-zone1-146	Equinix PH1 - Philadelphia	us-east4
	Philadelphia	phl-zone2-146	Equinix PH1 - Philadelphia	us-east4
TDC	Dallas	dfw-zone1-4	Equinix Dallas (DA1)	us-west1
	Dallas	dfw-zone2-4	Equinix Dallas (DA1)	us-west1

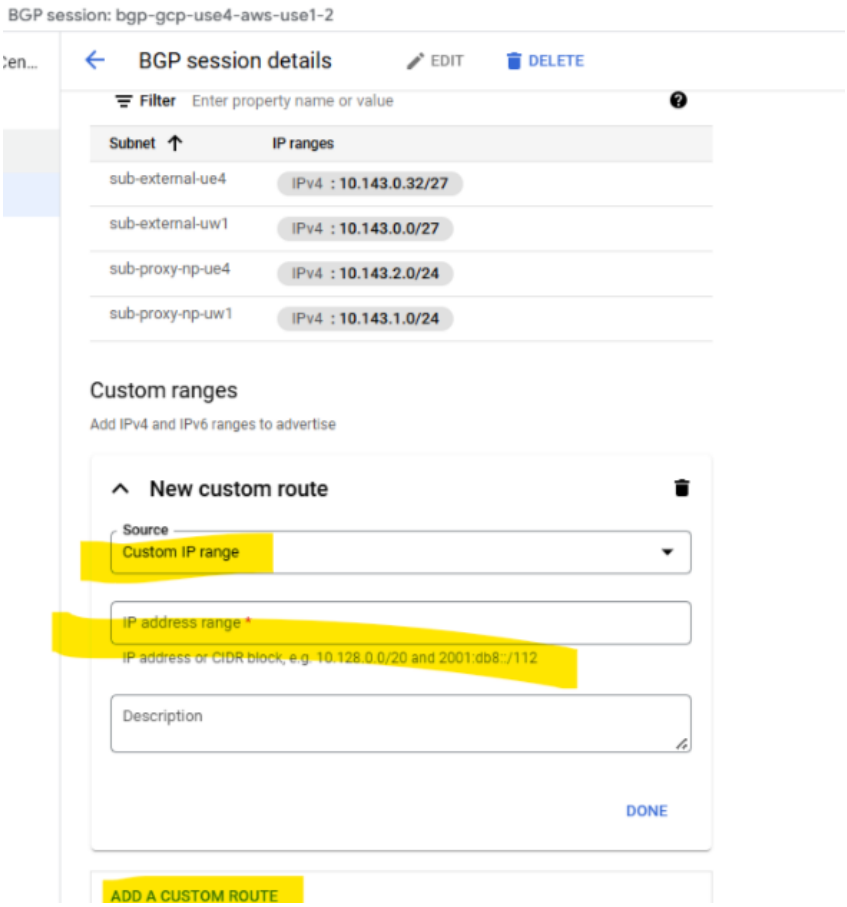
A Google Cloud project should be created to host all the Interconnect links.

5.3.1.2 Custom-Route BGP Configuration

The cloud router which terminates the dedicated interconnects to OnPrem is hosted on the VPC External. When BGP session is established between onprem Router to the GCP Cloud router for dedicated interconnects, the cloud router will only advertise the local subnets

available in the VPC external to OnPrem. To advertise any other IP ranges which are not part of "VPC external", "Custom route" option must be used. In future, if new IP Range/supernets are allocated for subnet expansion in other VPC for additional requirements, the same supernets must be included in the "Custom route" option to advertise those supernets over the BGP session to peers.
Refer screenshot provided below for example.

Reference URL: https://cloud.google.com/network-connectivity/docs/router/how-to/advertising-custom-ip



5.3.2 To AWS and Azure

Similarly, to connect to AWS and Azure, Cross-Cloud Interconnect will be used to establish a high-bandwidth dedicated connectivity. With Cross-Cloud Interconnect Google provisions a

dedicated physical connection between the Google network and that of another cloud service provider.
As with Dedicated Interconnect, for critical applications and high availability, Google recommends a 99,99% topology with two pairs of connections (one pair per region).

Quest is hosting resources in AWS in us-east-1 (N. Virginia) and us-west-2 (Oregon) regions. The following colocation facilities can be used for the Cross-Cloud Interconnect links:

Google Cloud Region	AWS Region	Metro area	Google Remote Location	AWS Remote Location
us-east4	us-east-1	Washington DC	aws-eqdc2	EqDC2
us-west1	us-west-2	Oregon	aws-ecpo1	ECPO1

Cross-Cloud Interconnect links will terminate in a Direct Connect Gateway on the AWS side. For each link, a Virtual Private Interface is required. Finally, a Virtual Private Gateway will provide access to an AWS VPC. Refer section 5.3.1.2 for the Note regarding Custom Route that must be used in the BGP Session.

As for Azure links, the following will be used:

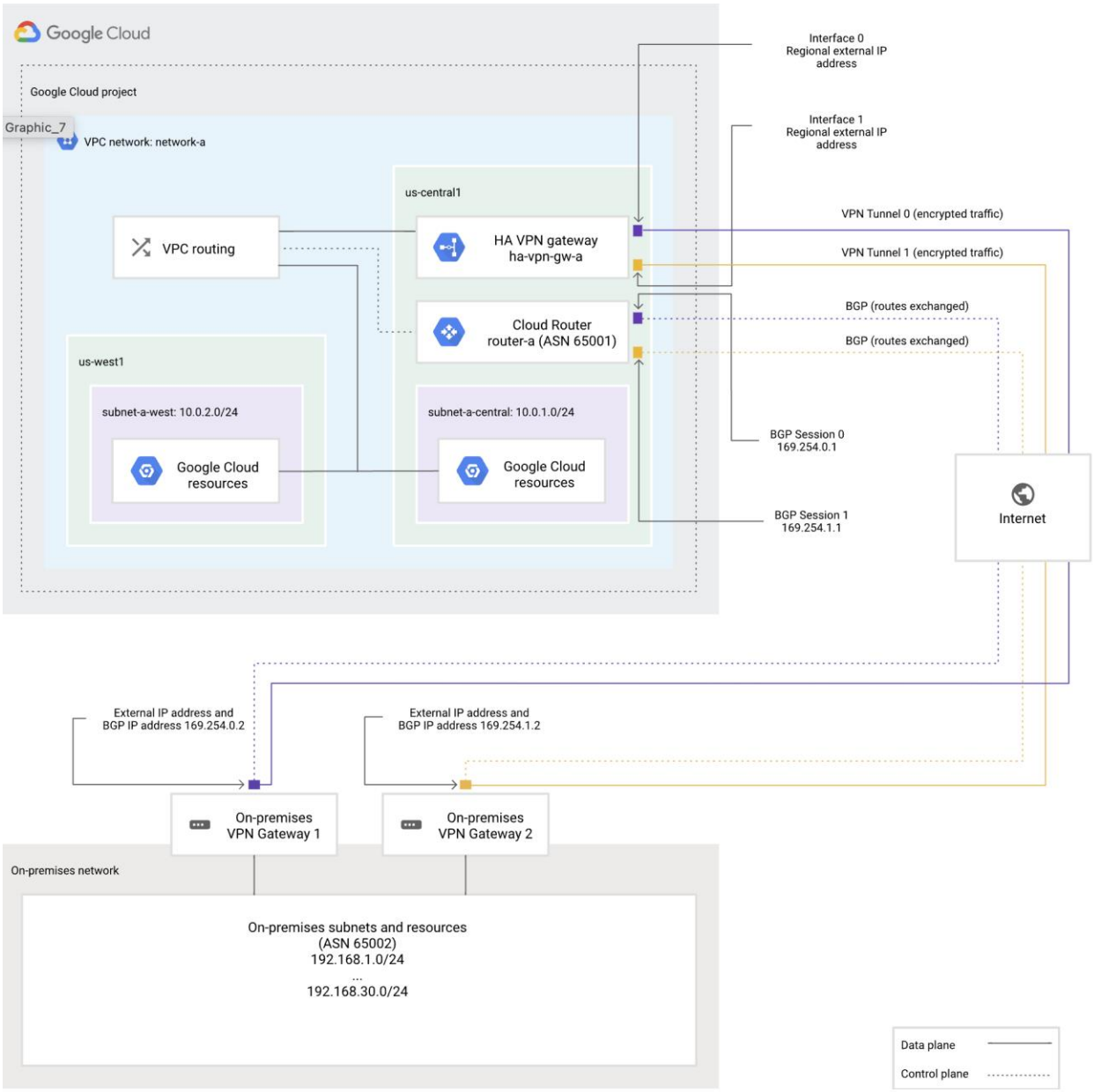
Google Cloud Region	Azure Region	Metro area	Google Remote Location	Azure Remote Location
us-east4	US-East-1 (N. Virginia)	Washington D.C.	azure-equinix-ashburn-dc2	Equinix-Ashburn-DC2
us-west1	US-West2 (California)	San Francisco	azure-coresite-santa-clara-sv7	CoreSite-Santa-Clara-SV7

In the Azure side of the links, a VNet with a subnet will be required in the regions where the links terminate. Express Route Circuits will be required to terminate the Interconnect links, with private BGP peering configured. Finally, a Virtual Network Gateway will connect the Azure VNet to the Google network. Refer section 5.3.1.2 for the Note regarding Custom Route that must be used in the BGP Session.

5.3.3 VPN Temporary Solution

As a temporary solution, while Quest waits for the setup up of Interconnect, VPN links will be set up to connect Google Cloud environment to Quest's on-premises and other cloud environments.

For this, a Google Cloud VPN managed solution will be used to establish these VPN links. The recommended solution is to set up an HA VPN topology that guarantees a 99.99% availability SLA. This is achieved with two VPN tunnels as presented in the following diagram.



This High Availability VPN topology should be set up in all regions. One pair of tunnels in us-east region and another pair in us-west region.

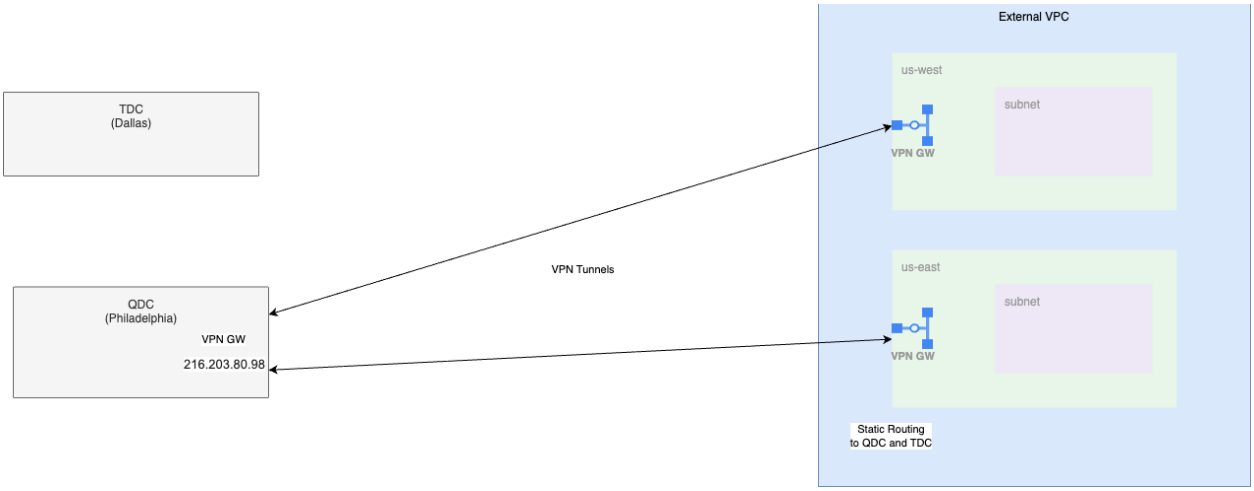
Each VPN tunnel will achieve a max bandwidth of 3 Gbps. To increase the bandwidth more HA VPN tunnels can be added.

Quest's VPN Gateways on their on-premises environments don't support Dynamic Routing (BGP). To overcome this limitation, Classic VPN will be used to connect to these environments. HA VPN will be used to connect to AWS and Azure.

5.3.4 Classic VPN to On-Premises

With Classic VPN, Quest will create IPsec VPN tunnels between an QDC VPN Gateway and the External VPC in Google Cloud. To achieve redundancy, two VPN tunnels will be created. One to a Cloud VPN GW in us-east and one to us-west.

The same routes will be configured for the first and the second tunnel. To set one tunnel to be primary, set a lower priority on that tunnel. If you want both tunnels to balance traffic, set their route priorities to be the same.



Additional tunnels will scale the bandwidth of the VPN. These will require a combination of either scaling VPN GWs on-premises or in Google.

The following connections will be established initially. Two tunnels in each region. VPN tunnels in us-west1 will connect to QDC. And us-east4 connects to TDC. These VPN links will be configured with Route-based routing .

VPC	Region	Datacenter	Google VPN GW IP	Tunnel 1 Remote IP	Remote Ranges
External	us-west1	QDC	Reserved Static Public IP (Update after Creation)	216.203.80.98	TBD
External	us-east4	QDC	Reserved Static Public IP (Update after Creation)	216.203.80.98	TBD

Static routes must be created in the External VPC for the subnets on-premises, with next hop set to the VPN tunnels.

The following Google Cloud Ranges should have a static route setup on-premises: 10.141.0.0/16, 10.142.0.0/16, 10.143.0.0/16.

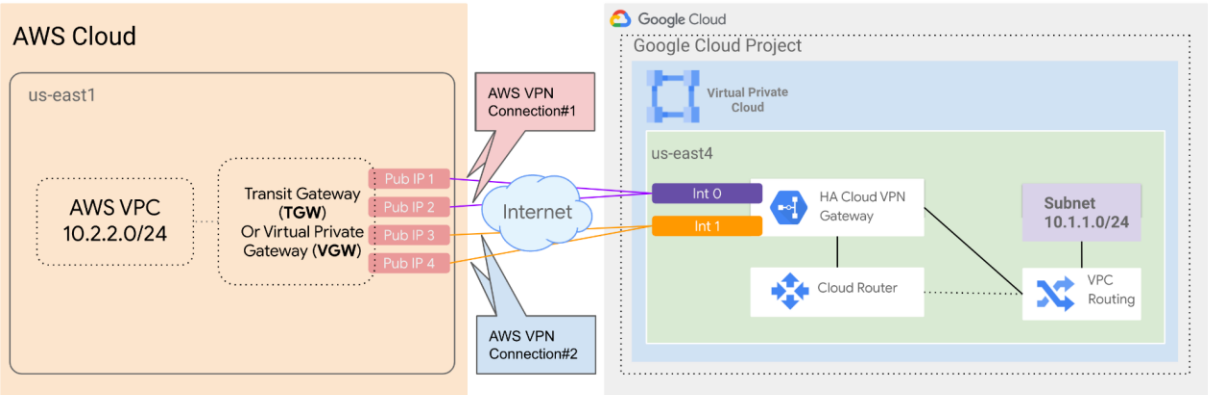
Firewall rules must be set on the External VPN and on the remote network to allow traffic to flow.

5.3.5 HA VPN to AWS and Azure

To connect to AWS and Azure, an HA VPN configuration will be used. To set up HA VPN, for each VPN link and region, Quest will have one Cloud Router and one VPN Gateway deployed in Google Cloud side. These resources will be used to terminate the two VPN tunnels on Google's side. The regional Cloud router will advertise all the Google Cloud IP ranges from the specific region back to the other side of the VPN (On-Premises, AWS and Azure). The same will happen when the Cloud Router will be used to terminate the Interconnect links.

To AWS

The following tutorial, guides on how to set up a HA VPN connection to AWS. Two links will be set up: one from us-west1 in GCP to us-west-2 in AWS, and one from us-east4 in GCP to us-east-1 in AWS. In AWS, the VPN has to terminate either in a Transit Gateway or in a Virtual Private Gateway.



The following Google Cloud HA VPN Gateways and Cloud Routers should be created. Additionally, 4 VPN Gateways should be created on the AWS side of Quest.

VPC	Region	AWS Region	Google HA VPN GW IP	Google Cloud Router ASN	AWS VPN GW IPs
External	us-west1	us-west-2	IP 1 (update after creating IP 2	TBD	TBD (4 IPs)
External	us-east4	us-east-1	IP 1 (update after creating IP 2	TBD	TBD (4 IPs)

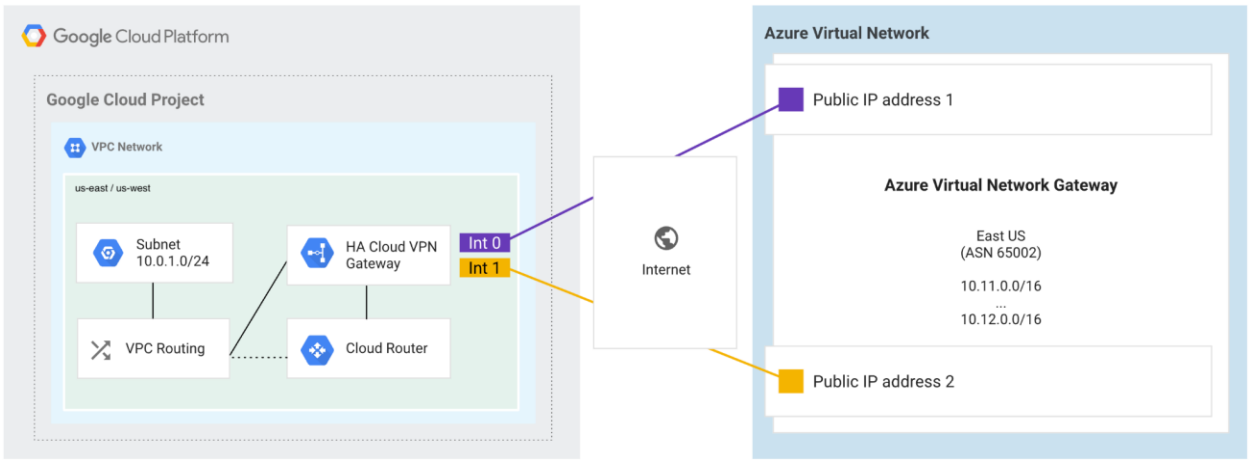
Google Cloud Routers will advertise all the Google Cloud ranges.

Tunnels Configurations:

Each Google Cloud VPN GW will terminate 4 VPN tunnels. And each Cloud Router must have 4 BGP interfaces with a private IP from the range *169.254.0.0/16*.

To Azure

Similarly, an HA VPN configuration will be used to connect to both Azure regions. An VPN Gateway must be configured in each region in Azure with 2 Public IPs and will terminate in the External VPC in Google Cloud. The following guide outlines the set up step by step.



VPC	Region	Azure Region	Google HA VPN GW IP	Google Cloud Router ASN	Azure VPN GW IPs
External	us-west1	US-West-2	IP 1 (update after creating IP 2	TBD	TBD (2 IPs)
External	us-east4	US-East-1	IP 1 (update after creating IP 2	TBD	TBD (2 IPs)

Tunnel Configurations:

Because the allowed ranges for Azure APIPA BGP peering internal IP addresses are 169.254.21.* and 169.254.22.*, you must select an available IP address in the /30 CIDR of those ranges for your Cloud Router BGP peering IP addresses.

5.4 Virtual Private Clouds (VPC)

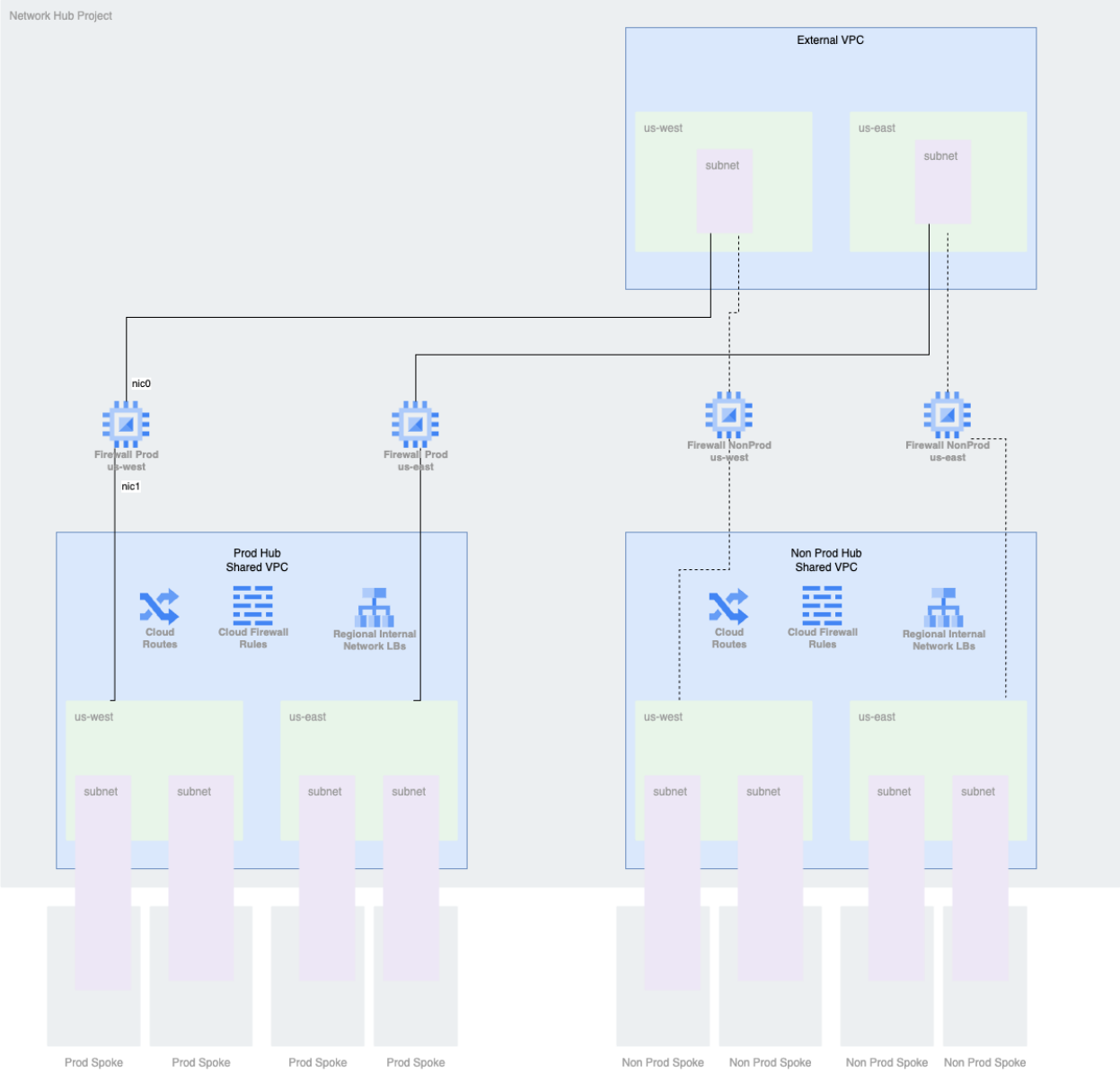
Quest should be able to create two types of networked environments in their Google Cloud landing zone: one where traffic between Google Cloud resources/applications doesn't require to be filtered by the Fortigate Firewall (no East-West traffic inspection) and environments where all traffic must be inspected by Fortigate (East-West traffic inspection).

5.4.1 Shared VPC

For environments where there's no requirement of traffic inspection, the use of Shared VPC is proposed.

Shared VPC allows an organization to connect resources from multiple projects to a common Virtual Private Cloud (VPC) network, so that they can communicate with each other securely and efficiently using internal IPs from that network. When you use Shared VPC, you designate a project as a host project and attach one or more other service projects to it. The VPC networks in the host project are called Shared VPC networks. Eligible resources from service projects can use subnets in the Shared VPC network. A project that participates in Shared VPC is either a host project or a service project.

Two Shared VPCs will be created: one non prod and one prod. This will guarantee network isolation between non production and production environments.



The Fortigate VMs are configured in a multi-nic setup, with a network interface in External VPC and in the Shared VPCs. For simplicity purposes, the Fortigate management and HA sync interfaces are not represented in the diagram.

Every time an environment is needed, a subnet is created in the corresponding Shared VPC. This subnet is linked to a service project (spoke). Spoke members will be able to use this su3+bnet to create resources.

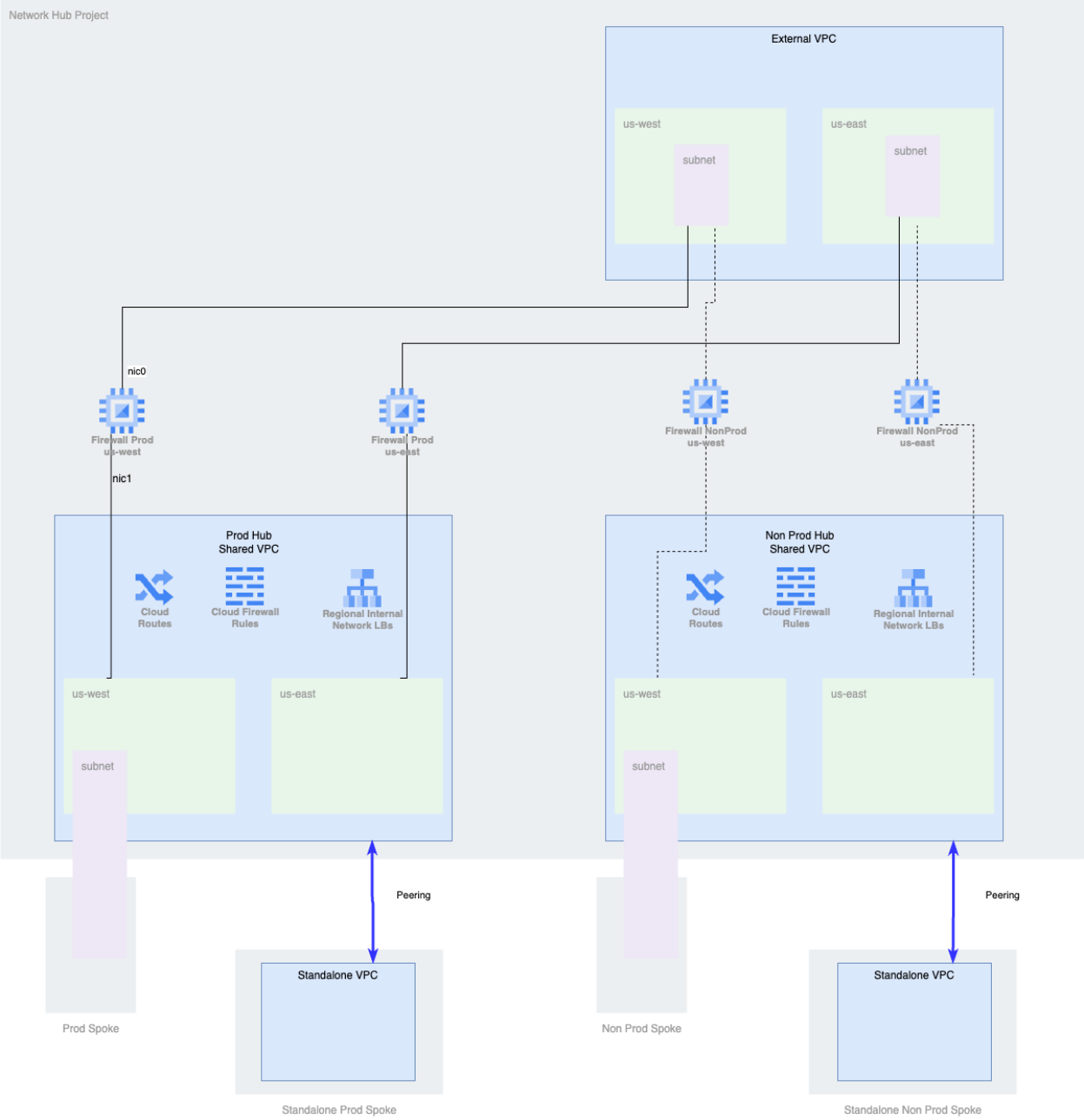
This setup guarantees that:

- Intra-VPC is not filtered by the Firewall (including inter-region traffic inside the VPC)
- Inter-VPC traffic goes over firewall

Each VPC can and should be secure with VPC Firewall rules. These are stateful rules to filter VM traffic.

5.4.2 Standalone VPCs

Whenever applications require environments with East-West traffic inspection, a Standalone VPC should be created. Standalone VPCs will be used to host workloads when there's a requirement to inspect all traffic. Standalone VPCs will be connected to the Hub VPCs via Peering or VPN.



Ingress and egress traffic from Standalone Spoke VPCs will be inspected by the centralized firewall appliance.

5.4.3 Sandbox VPCs

Quest wants to have the possibility of creating Sandbox environments to experiment new services, run PoCs, and for early development of applications that could later evolve to production. These Sandbox environments might need connectivity to Quests networks.

Whenever a Sandbox environment is needed, it should be created in a corresponding Sandbox project (according to Organization hierarchy defined above). If networked resources are needed for this environment, a completely isolated VPC should be created. This VPC won't be connected to Quest's network. This will give the possibility for the network team to assign any CIDR range to the VPC since there's no problems with overlapping ranges.

If connectivity to Quest's network is needed, the sandbox VPC should be treated as a normal connected Non Production VPC.

5.4.4 Provision new environments

Whenever a new application/environment needs to be provisioned, a decision needs to be made on what kind of networking environment is needed, based on security and traffic inspected aspects defined above.

Subnet in a Shared VPC

Whenever an application is deployed in a subnet in the shared VPC, the following tasks must be performed:

1. Create a project to host the application resources
2. Create a subnet in the Shared VPC (Nonprod or Prod), and specify region and IP range of the subnet (according to table in section 5.8). Share the sub with the application project.
3. The Shared VPC already has the necessary routes for applications deployed in the subnet to communicate
4. Google Cloud Firewall rules should be created for the application (see section 5.10) in the subnet. These initial rules only allow communication within this subnet (VMs of the application). Based on the requirements of the application, additional rules need to be added.
5. Tag application components (VMs, GKE nodes) with required application and regional tags (see section 5.9 and 5.10)
6. If the app requires communication outside of the VPC, add required Fortigate Firewall policies.

Standalone VPC

Whenever an application is deployed in standalone, the following tasks must be performed:

1. Create a project for the application
2. In this project create a VPC and peer it with Hub VPC (nonprod or prod)
3. Add routes according to section 5.9
4. Create required subnets or subnets for the applications.
5. Google Cloud Firewall rules should be created for the application (see section 5.10) in the subnet. These initial rules only allow communication within this subnet (VMs of the application). Based on the requirements of the application, additional rules need to be added.
6. Tag application components (VMs, GKE nodes) with required application and regional tags (see section 5.9 and 5.10)
7. If the app requires communication outside of the VPC, add required Fortigate Firewall policies.

Sandbox

Sandbox VPC are completely isolated from Quest network, so networking can be more customized. The following steps are required:

1. Create project for sandbox applications
2. Create VPC and customize it as needed
    a. Subnets, regions, CIDR ranges as required
    b. Desired routes to Internet or not
    c. Google Cloud Firewall rules according to application needs

5.4.5 Limits and Scalability

The scalability of a Shared VPC architecture is limited by quota limits, particularly those for VPCs, such as the number of instances per VPC network, firewall rules, and internal TCP/UDP load balancer forwarding rules per VPC network. When a limit is reached, another Shared VPC needs to be created and connected to the Firewall with a nic.

5.4.6 Subnets

Initially, subnets will be created for the two US regions. The CIDR ranges for these subnets will be taken from a CIDR range that Quest will define for Google Landing Zone.

The following subnets will be created to host the Fortinet Firewall and terminate the Interconnect/VPN links:

VPC	Subnet	Region	IP range
External	External US West	us-west1	10.143.0.0/27
External	External US East	us-east4	10.143.0.32/27
External	Proxy only West	us-west1	10.143.2.0/24
External	Proxy only West	us-east4	10.143.1.0/24
Management	Management US West	us-west1	10.143.0.64/28
Management	Management US East	us-east4	10.143.0.80/28
HA Sync	HA Sync US West	us-west1	10.143.0.96/28
HA Sync	HA Sync US East	us-east4	10.143.0.112/28
Non Prod Hub	Non Prod Hub US West	us-west1	10.142.128.0/28
Non Prod Hub	Non Prod Hub US East	us-east4	10.141.128.0/28
Prod Hub	Prod Hub US West	us-west1	10.142.0.0/28
Prod Hub	Prod Hub US East	us-east4	10.141.0.0/28

Whenever Standalone VPC are required, CIDR ranges will be taken from the corresponding environment/region range.

All subnets must have Private Google Access and VPC Flow Logs.

5.4.7 Traffic Logging

VPC Flow Logs records a sample of packets sent from and received by VMs, including instances used as Google Kubernetes Engine nodes, and packets sent through VLAN attachments for Cloud Interconnect and Cloud VPN tunnels.

Flow logs are aggregated by IP connection (5-tuple). These logs can be used for network monitoring, forensics, security analysis, and expense optimization.

You can view flow logs in Cloud Logging, and you can export logs to any destination that Cloud Logging export supports.

VPC Flow logs must be enabled in all Subnets created in the Google Cloud Landing Zone. After being enabled, these logs will be stored in Cloud Logging for 30 days (default retention). If you want to keep logs longer than that this can be configured.

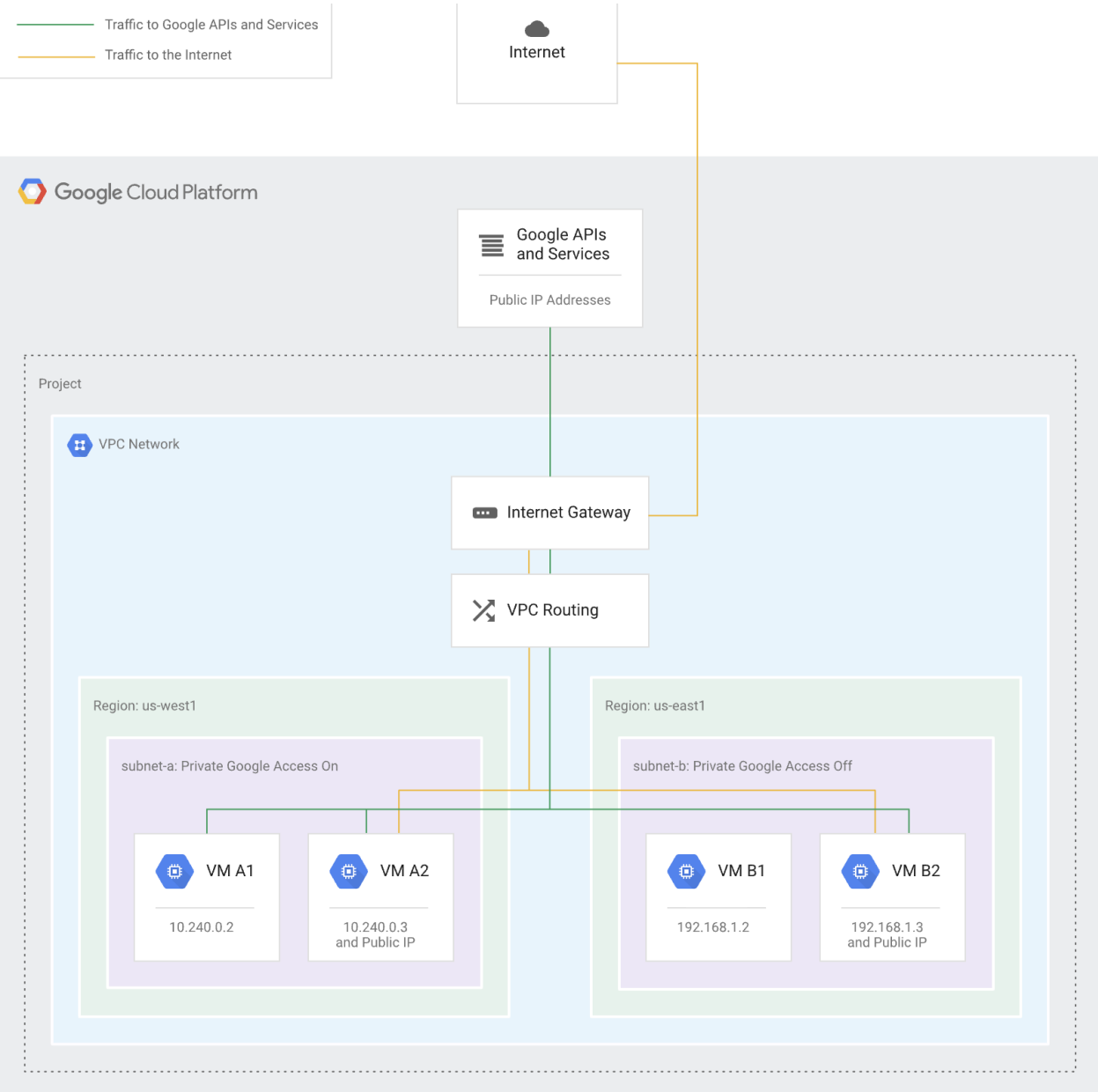
5.5 Private access to Google APIs and Google Services

5.5.1 Private Google Access

Instances with external IP addresses can access Google APIs and services through their assigned public IP address. However, resources with only internal IP addresses will need to use Private Google Access (PGA).

This feature provides instances created with only internal IP addresses the ability to send requests to Google APIs and services not through the Internet. Those requests stay within Google's network.

Quest will enable PGA at the subnet level. When enabled, instances in the subnet that only have a private IP address can send traffic to Google APIs and services through the default route with a next hop to the default internet gateway. The diagram below illustrates how Private Google Access works from within VPC.



This will provide internal traffic when accessing Google APIs like Storage and BigQuery. For Internal VPCs without a default route to the Internet Gateway, a route must be added to CIDR Range "199.36.153.8/30" and "34.126.0.0/18" with next hop as Default Internet Gateway. Additionally, a private DNS zone must be created for the domain "*.googleapis.com" and CNAME as "private.googleapis.com" or "restricted.googleapis.com" depending on whether the API being used is protected with VPC Service Controls, as detailed here. This should be set on the Prod and NonProd Hub VPCs and in the Standalone VPCs that require private connectivity to Google APIs.

The following DNS configuration should be created:

Record Type	Zone	Record Name	Record Value
CNAME	googleapis.com	*.googleapis.com	private.googleapis.com or restricted.googleapis.com
A	googleapis.com	private.googleapis.com	199.36.153.8 - 199.36.153.11

On-premises hosts can reach privately to Google APIs and services by using Cloud VPN or Cloud Interconnect from your on-premises network to Google Cloud. To enable Private Google Access for on-premises hosts, you must configure DNS, firewall rules, and routes in your on-premises and VPC networks.

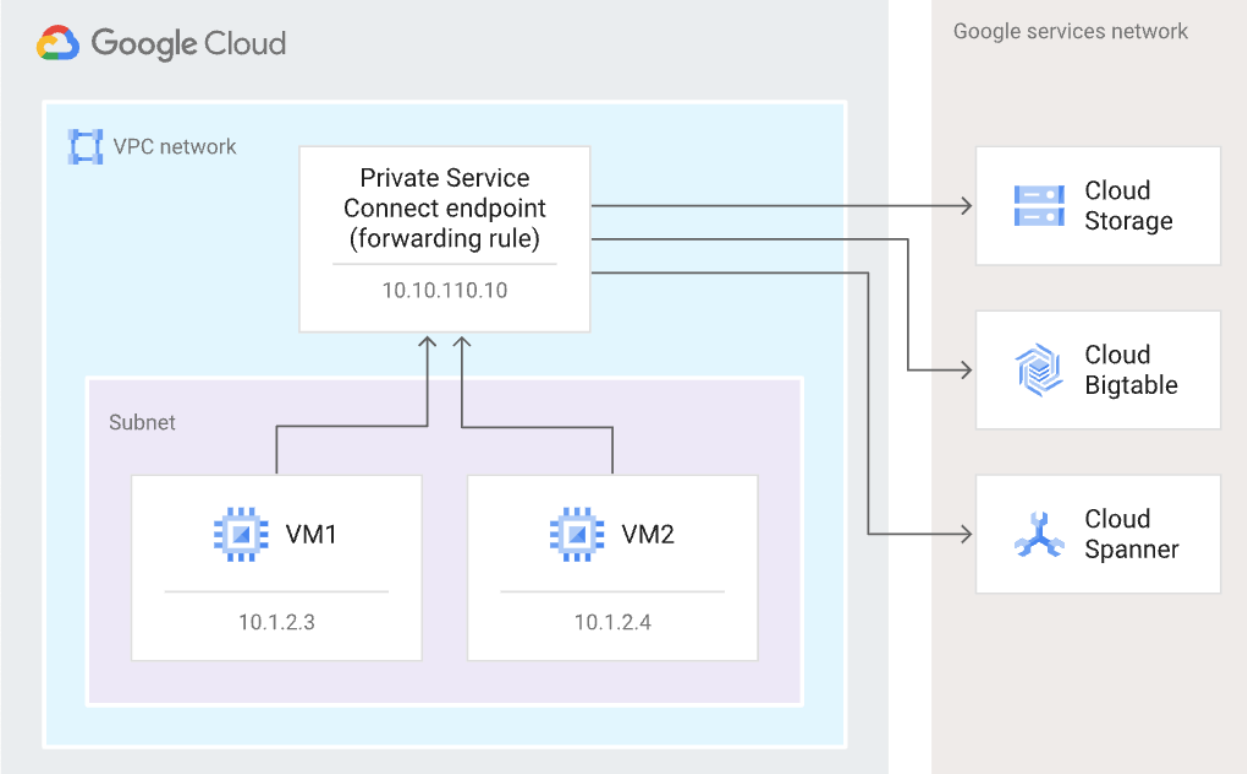
In Quest's On-premises network, DNS Servers must be configured to resolve "*.googleapis.com" similar as presented in the previous table.

Private Service Connect

You can use Private Service Connect to access all supported Google APIs and services from VMs in your VPCs and from on-premises. Private Service Connect is a capability of Google Cloud networking that allows consumers to access managed services privately from inside their VPC network. With Private Service Connect, consumers can use their own internal IP addresses to access services without leaving their VPC networks. Traffic remains entirely within Google Cloud. Private Service Connect provides service-oriented access between consumers and producers with granular control over how services are accessed.

Using Private Service Connect lets you do the following:

- Create one or more internal IP addresses to access Google APIs for different use cases.
- Direct on-premises traffic to specific IP addresses and regions when accessing Google APIs.



With Private Service Connect, you can create private endpoints using global internal IP addresses within your VPC network. You can assign DNS names to these internal IP addresses with meaningful names like storage-vialink1.p.googleapis.com and bigtable-adsteam.p.googleapis.com. These names and IP addresses are internal to your VPC network and any on-premises networks that are connected to it using Cloud VPN tunnels or VLAN attachments/Interconnect.

5.5.2 Private Services Access

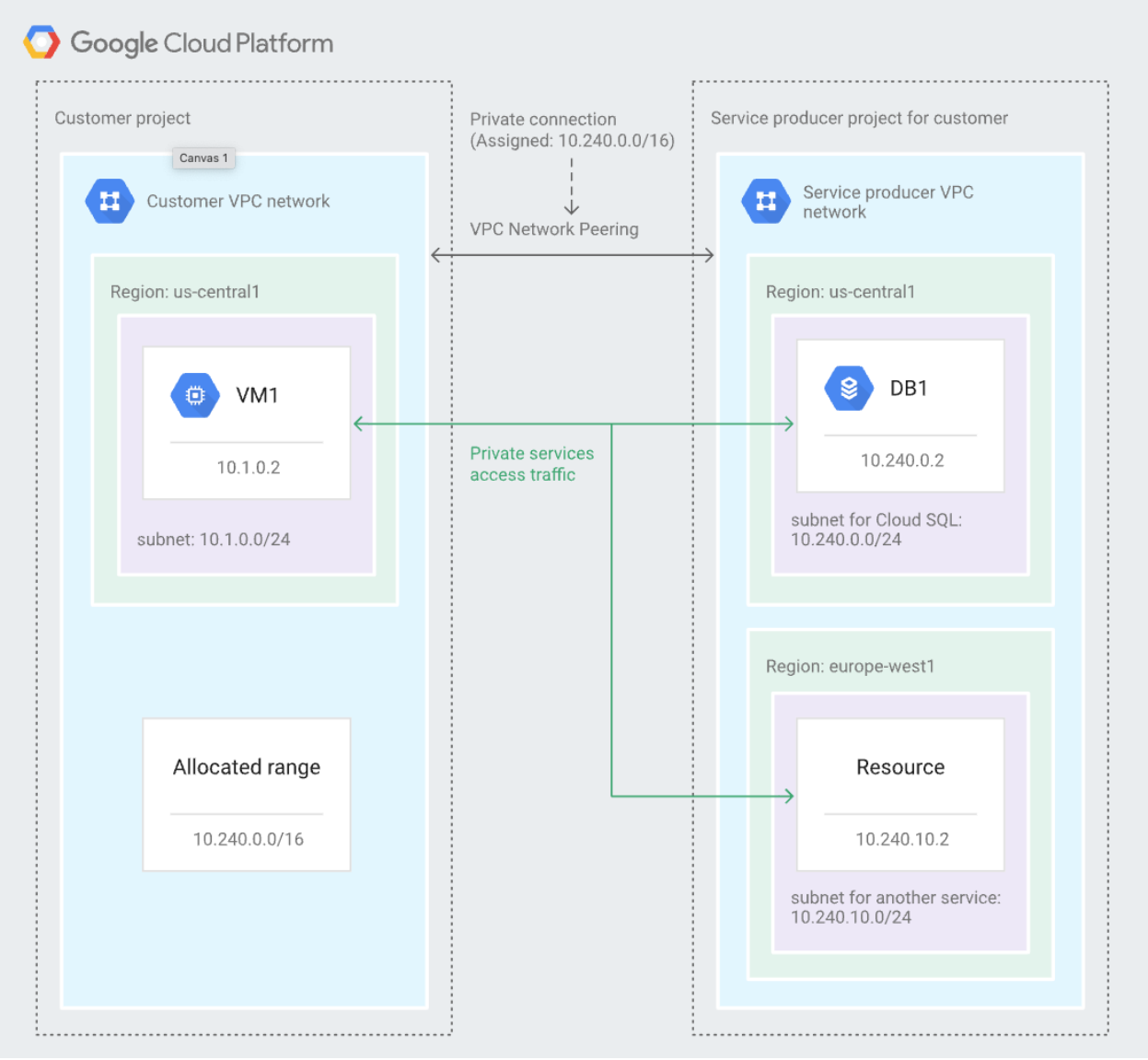
Google can offer services with internal IP addresses that are hosted in a VPC network. These VPC network resources sit in a Google or third manager project (sometimes called a tenant project).

These services include but are not limited to:

- Cloud SQL
- Cloud Memorystore
- Cloud Filestore
- Google Kubernetes Engine (master connectivity)

The full list of services accessed through private services access can be found here.

Private Services Access (PSA) enables reaching those internal IP addresses using private connectivity. This is useful if you want your VM instances in your VPC network to use internal IP addresses instead of external IP addresses. Additionally, the producer services can be reached from on-premises networks over VPN or Partner/Dedicated Interconnect.



PSA requires you to first allocate an internal IP address range and then create a private connection (peering). An allocated range is a reserved CIDR block that can't be used in your VPC network for other purposes (e.g. as subnet CIDR). It's set aside for service producers only and prevents overlap between your VPC network and the service producer's VPC network.

The following ranges will be initially allocated for PSA for Production and Non Production:

VPC	IP Range
Prod US East	10.141.127.0/24
Prod US West	10.142.127.0/24
Non Prod US East	10.141.255.0/24
Non Prod US West	10.142.255.0/24

5.6 Traffic management

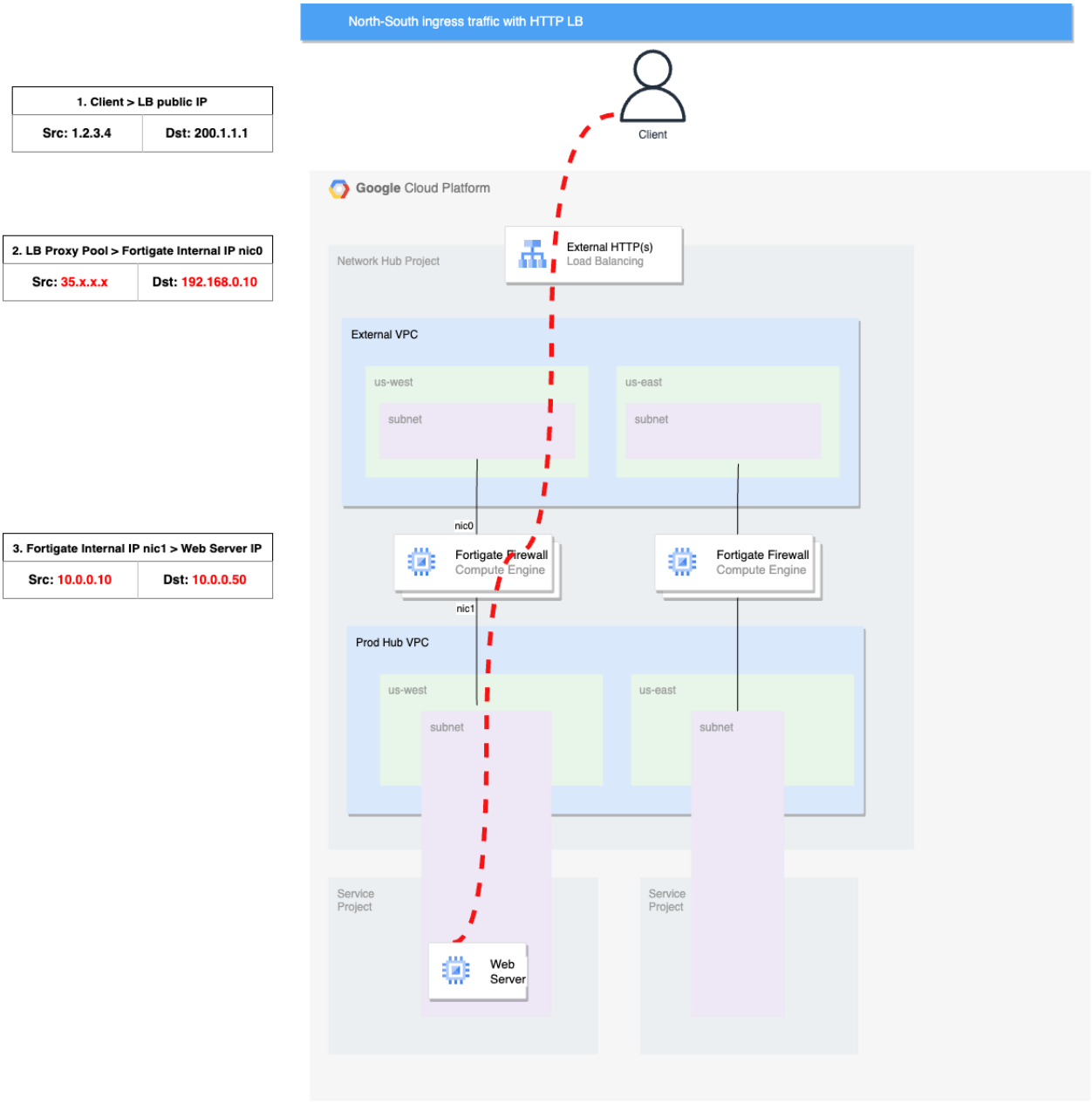
5.6.1 Ingress Traffic

Quest wants to take advantage of Google Cloud Armor and its WAF features to protect public applications. For this purpose, applications will be exposed to the Internet with Regional External Application Load Balancers. Google external Applications load balancer terminates incoming Layer 7 connections to its frontend and initiates corresponding new connections to the Fortigate VMs in the backend.

The traffic will flow as follows:

1. User initiates connection to an public anycast VIP assigned to a External ALB
2. HTTPs requests from the Internet are terminated by ALB and passed to the active FortiGate instance, sourcing these connections from the IP range 35.191.0.0/16.
3. FortiGate DNATs connection to the workload. SNAT is also applied to ensure the return packet will be sent back to FortiGate instance and not directly to ALB address
4. Return traffic is sent to FortiGate VM which originally processed the packet. FortiGate then forwards the return packet back to ALB.

The backend of the Load balancer are the Instance Groups of Fortigate VMs. The Fortigate cannot forward traffic directly to the workloads in the Internal VPCs. So the Fortigate needs to apply DNAT to route traffic to the desired workload.

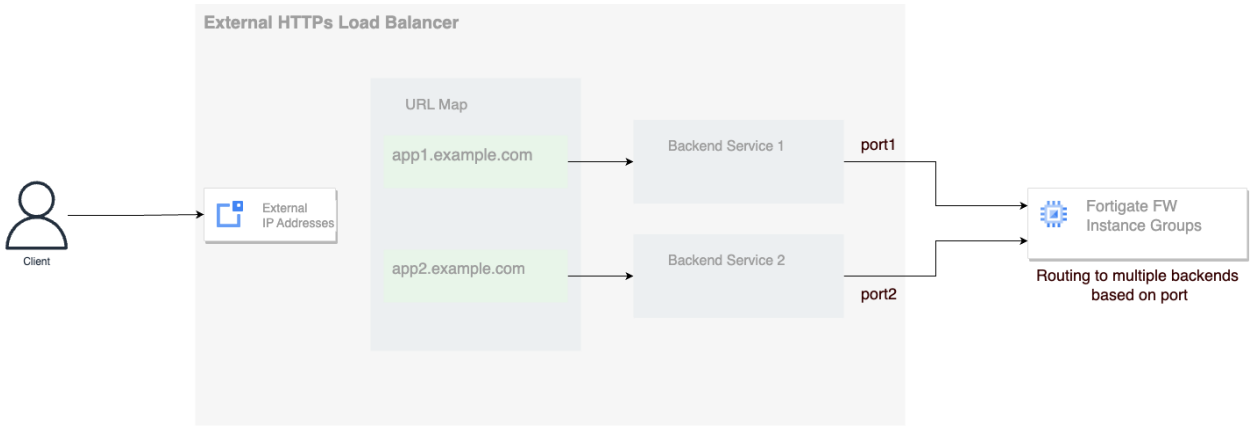


Expose multiple services in the External Load Balancer

Google Regional External Application load balancer terminates the connection from the client and starts a new connection sourced from the LB Proxy pool, which means the original client IP address will not be visible in the IP layer. The destination will be set to FortiGate VM private IP address.

Due to this, to expose multiple services in the same Load balancer, different ports must be used to differentiate between workloads. This can be achieved by creating a Backend Service with a specific Named Port for each application that needs to be exposed. The named port defines the destination port used for the TCP connection between the LB and the backend instance. The LB forwards traffic to different backend services based on host/url.

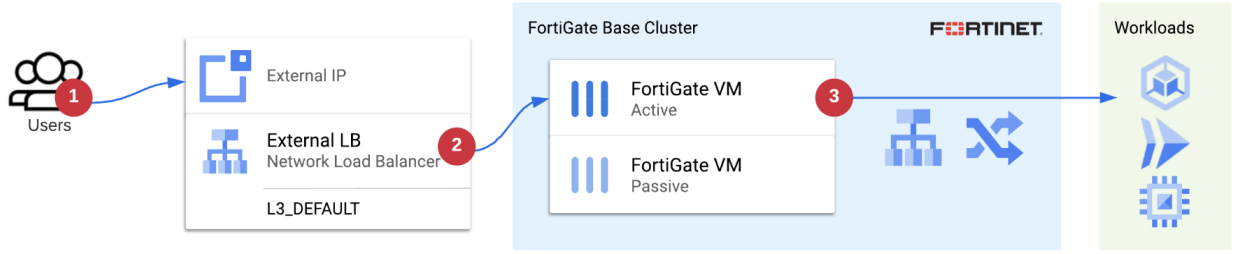
And then the Backend Services sends traffic to the Fortigate VMs through the named port. Finally the Fortinet FW routes traffic to the destination based on the port.



For each Applications/Service exposed in the Load Balancer, Cloud Armor security policies will be applied with required WAF rules. Cloud Armor policies are applied in the Backend Services of the Load Balancer. Cloud Armor security policies should also be used to protect applications based on client IP. Additionally, if required, Backend Services can be configured as Origins for Google Cloud CDN.

Non HTTP applications

Quest will want to deploy non-HTTP based applications. For example, applications relying on UDP traffic. For this purpose, these applications should be exposed via an External Passthrough Network Load Balancer. The passthrough NLB is a regional non proxied load balancer. To expose multiple services in the same Load balancer, multiple public IPs/ports can be used. Similar to HTTP load balancers, each NLB will have two Unmanaged Instance Groups of Fortigate VMs as backend.



The traffic will flow as detailed below:

1. User initiates connection to an external IP assigned as ELB frontend
2. External Load Balancer forwards connection to active FortiGate instance
3. FortiGate uses public IP and Firewall Policy to DNAT connection and send it to the desired target workload - the destination IP:port will be mapped to the workload internal IP:port
4.

Workload receives connection and replies. Return traffic is routed (VPC Static routes) to regional Internal Load Balancer. Internal Load Balancer forwards to active Fortigate and Fortigate NATs traffic back to client on the Internet.

With the External Network Load Balancer, Cloud Armor Advanced network DDoS protection will be used to protect its public Endpoints.

5.6.2 Egress Traffic

For connections initiated from internal workloads, traffic will be routed to the respective regional Internal Load Balancer that forwards traffic to an active Fortinet VM. Fortinet will be handling NATing to a public IP address:

1. Workload initiates connection to Internet
2. VPC static route routes traffic to ILB ip
3. ILB selects currently active Fortinet VM and forwards traffic to it
4. Fortinet inspects the traffic according to its policy.

After inspection, FortiGate performs source NAT to one of the external IPs of the External Load Balancer and passes the connection. Fortigate IP Pools should be used for this. It's also possible to use Google native Cloud NAT to handle the translation to an external IP address. But using IP address pools in FortiGate for SNAT offers better control over the external IP addresses assigned to individual groups of outbound connections when compared to using directly attached external IP addresses or first-party NAT solutions.

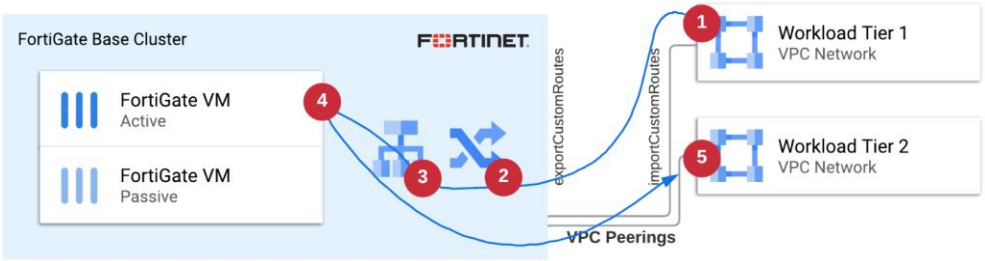
5.6.3 East-West Traffic

As stated before, traffic within a Shared VPC won't be inspected by the Fortinet Firewall. It is always routed internally and it doesn't leave the VPC.

Whenever traffic inspection is required within two environments, for example, applications following the three tier architecture, a different setup is required so that traffic is inspected by Fortinet. In this case, both segments (tiers) need to be placed in a dedicated VPC peered to the Hub VPC (standalone/spoke VPCs). Spoke VPCs will require a static route (applied to VMs with a tag for the region) and with next hop to the Internal Load Balancer IP.

In the case the traffic is routed as detailed below:

1. Workload in a Spoke VPC initiates connection towards workload in another spoke VPC
2. Static route forwards traffic to region specific ILB
3. ILB forwards the traffic to active Fortinet VM for inspection
4. Fortinet inspects traffic and forwards back to Hub VPC
5. Routes in Hub VPC deliver traffic to Spoke VPC



5.6.4 Inter Region Traffic

Inter-region traffic will be treated as normal traffic within the VPC. That's because Google Cloud VPCs are global resources, and can host subnets from multiple regions.

5.6.5 Production environment to Non Prod environment Traffic

Quest requires that traffic should be able to flow from nonproduction environments to production environments.
This traffic will flow as described below:

1. Workload initiates connection from production VPC to Non Production VPC
2. VPC static route routes traffic to ILB ip
3. ILB selects currently active Fortinet VM and forwards traffic to it
4. Fortinet inspects the traffic according to its policy and forwards to External VPC
5. Static route in External VPC forwards traffic to Non production ILB
6. Non Production ILB selects currently active Fortinet VM and forwards traffic to it
7. Traffic is forwarded to Non Production VPC.

In this scenario, traffic will traverse Fortigate Firewalls twice.

5.6.6 On-Premises, AWS and Azure Traffic

Quest also requires inspection of all traffic from other Quest environments: On-Premises, AWS and Azure. Traffic from these environments will be reaching the External VPC over an Interconnect or VPN Cloud Router.

1. Cloud Routers advertise the Google Cloud IP Ranges to On-Premises, AWS and Azure
2. Traffic traverses Interconnect/VPN links arrives to Google Cloud in the External VPC
3. Custom Static routes route traffic to region/environment specific ILB
4. ILB forwards traffic to active Firewall appliance
5. Firewall inspects traffic and forwards to Hub VPC
6. Routes in Hub VPC redirect traffic to destination workload (in the VPC or peered VPC)
7. Reply traffic goes back over Firewall
8. In external VPC, BGP routes advertise link back to On-premises/AWS/Azure

5.7 Fortinet Firewall Configuration

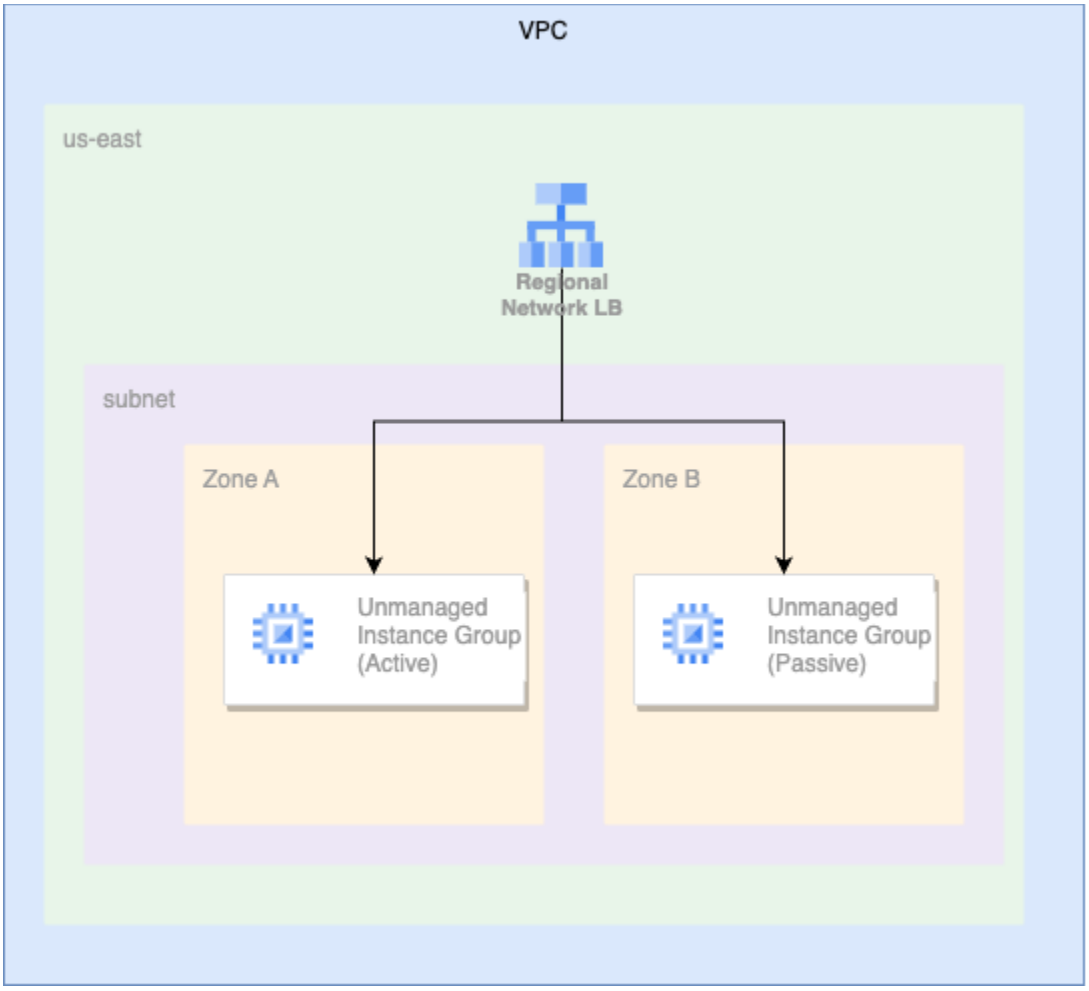
Each Fortinet Firewall will be composed of two VMs in an Active-Passive configuration for High Availability. Each VM will be in a different zone of the same region. Internal Network Load Balancers will be used to forward internal traffic to the Active VM. The backends of the NLBs will be two unmanaged instance groups (one for each VM). Unmanaged IGs must be created in the External VPC. But are used by Load Balancers in all VPCs.

Internal and Passthrough network load balancers will be used with health checks configured on tcp:8008.

During a failover, Load Balancers identify the active VM and redirect traffic through it. During a failover, the connection tracking feature of Google Cloud Load Balancing helps sustain existing TCP connections. All internal Load Balancers will have frontends listening on all protocols (L3) and all ports. And will have a reserved static internal IP. The following configuration will be used for all load balancers:

Internal LB Configurations	Value
Frontend protocol	TCP for Internal LBs
Ports	All
Global Access	Disabled
Internal IP	Non-shared, static reserved IP
Health Checks	TCP:8008

External VPC will have 2 External Applications Load Balancers per region to expose applications to the Internet (production and Non Production LBs) . Similarly, it will have 4 Internal NLBs to expose applications internally to On-premises/AWS/Azure. These load balancers will have the Fortigate UIGs as backends. Similarly, Hub VPCs will have 2 internal passthrough NLBs each - 1 per region.



In each VM, interface 1 and 2 will be used for External and Internal VPCs. Interface 3 will be used for HA sync and interface 4 for management. These interfaces should use a static internal IP address. Management Interface might require a Public IP for external access.

More details on Terraform configuration of Fortigate VMs in Google Cloud can be found in official documentation and Fortinet docs.

5.8 IP addressing spaces

Quest will assign the following CIDR ranges for their Google Cloud Landing Zone, according to regional needs.

Region	CIDR Range	Environment
GCP US East	10.141.0.0/16	10.141.0.0/17 (Prod) 10.141.128.0/17 (Non Prod)
GCP US West	10.142.0.0/16	10.142.0.0/17 (Prod) 10.142.128.0/17 (Non Prod)
GCP Shared Infra	10.143.0.0/16	N/A

5.9 Routing

The following routing should be applied to the internal Production and Non Production Hub VPCs:

Route Type	Destination	Instance Tags	Next Hop	Observations
static	0.0.0.0/0	Region Tag	Regional ILB	These routes must point to the internal Static IP of the ILB
Subnet	N/A	N/A	N/A	Automatically created
static	199.36.153.8/30	N/A	Default Internet	Private Google access

Note that to route traffic destined outside of the VPC via the Fortigate FW, a static route is added with next hop as the regional ILB. This route is applied to regional instances based on a network tag. So all VMs in the VPCs must have a network tag to identify the region.

Standalone Spoke VPCs

Route Type	Destination	Instance Tags	Next Hop	Observations

static	0.0.0.0/0	Region Tag	Regional ILB From Peered Hub	These routes must point to the internal Static IP of the ILB
Subnet	N/A	N/A	N/A`	Automatically created
static	199.36.153.8/30	N/A	Default Internet	Private Google access

External VPC

Route Type	Destination	Instance Tags	Next Hop	Observations
Cloud Router advertised routes to OnPrem and other Cloud providers				
static	us-west NP CIDR us-east NP CIDR us-west Prd CIDR us-east Prd CIDR	N/A	Specific ILB	
static	0.0.0.0/0	N/A	Default Internet	
Subnet	N/A	N/A	N/A	Automatically created

Note: Cloud Routers will advertise Google Cloud regional CIDR ranges back to OnPremises, AWS and Azure.

Fortigate Management VPC

Route Type	Destination	Instance Tags	Next Hop	Observations
static	0.0.0.0/0	N/A	Default Internet	
Subnet	N/A	N/A	N/A	Automatically created

Fortigate HA Sync VPC

Route Type	Destination	Instance Tags	Next Hop	Observations
static	0.0.0.0/0	N/A	Default Internet	
Subnet	N/A	N/A	N/A	Automatically created

5.10 Google Cloud Stateful Firewall Rules

GCP firewall rules are L3/L4 stateful rules that let you allow or deny traffic to and from your VMs based on a configuration you specify. Enabled GCP firewall rules are always enforced, protecting your instances regardless of their configuration and operating system, even if they have not started up.

Every VPC network functions as a distributed cloud firewall. While GCP firewall rules are defined at the network level, connections are allowed or denied on a per-instance basis. You can think of the GCP firewall rules as existing not only between your instances and other networks, but between individual instances within the same network. GCP firewall rules are matched via IP address, arbitrary tags, or via service accounts.

Firewall rules, along with Tags in the VMs, will be used to ensure isolation between applications or teams in the same Shared VPC. The following rules should be created for an example Application 1 in a subnet:

Direction	Target	Source	Destination	Action
ingress	tag: app1	tag: app1	n/a	allow
egress	tag: app1	n/a	Quests Private IP ranges	allow

Firewall rules will also need to be configured to allow communication to and from the interfaces of the Fortinet Firewall appliance.

VPC	Direction	Target	Source	Destination	Protocol/Port
External	ingress	tag: fortigate-external	0.0.0.0/0	tag: fortigate-external	all
External	egress	tag: fortigate-external	n/a	0.0.0.0/0	all
Hub Prod	ingress		<Internal Google Cloud Ranges>	tag: fortigate-hub-prod	all
Hub Non Prod	ingress		<Internal Google Cloud Ranges>	tag: fortigate-hub-nprod	all
Management	ingress		<Quest source>	tag: fortigate-mgmt	all
HA Sync	ingress		tag: fortigate-ha	tag: fortigate-ha	all

5.11 DNS

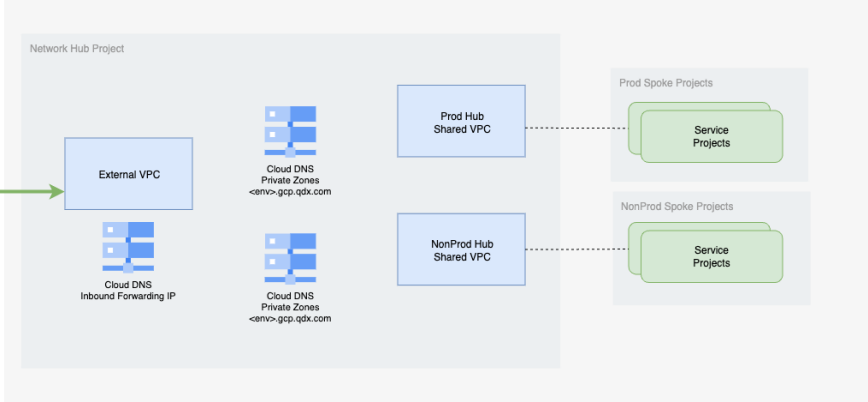
In terms of internal DNS resolution, Quest is looking for the following:

- Being able to look up GCP DNS VMs in a VPC
- Use gcp.qdx.com private domains from within Google Cloud
- Resolve gcp.qdx.com subdomains from other Quest environments
- Resolve qdx.com domains from Google Cloud

Authoritative DNS resolution for the private Google Cloud environment is done by Cloud DNS. This private Cloud DNS domain will be created as gcp.qdx.com. On-premises DNS servers will remain authoritative for other Quest DNS domains.

5.10.1 Resolution of Google Cloud names from On-Prem

To resolve gcp.qdx.com domains from On-Premises networks, DNS forwarding will be configured on Quests On-Premises DNS Server to target the Cloud DNS inbound forwarding IP address, which will be created via the Inbound Server Policy configuration in the External VPC.



Environment specific Cloud DNS private Zones (dev.gcp.qdx.com) will be created, and associated with all the VPCs. These DNS private zones will be made visible to any VPC, because Quest requires communication between all environments.

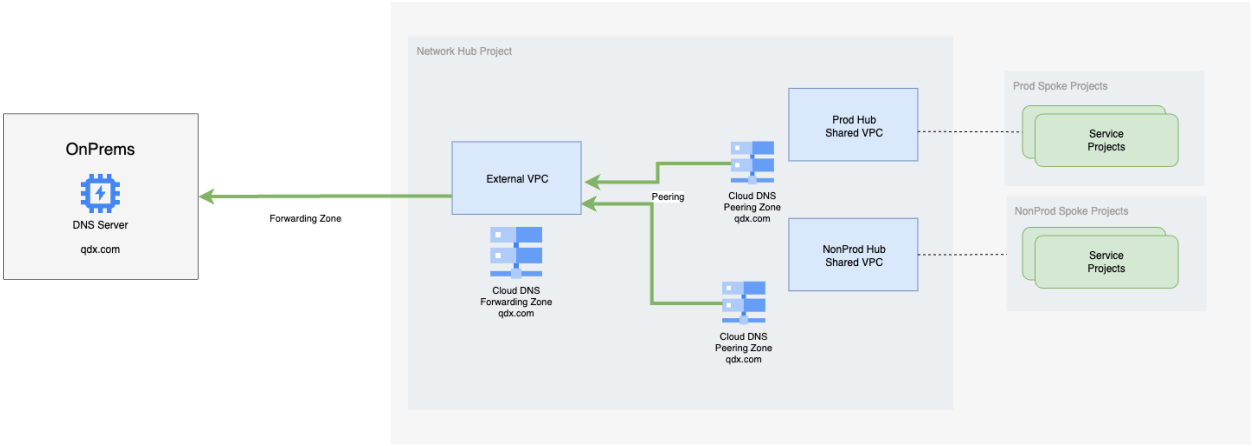
Additional application specific DNS Private zones can be created in the shared VPC spoke projects to delegate zone management to the app teams. These zones can be consumed by selected VPC's.

To ensure that traffic can flow from on-premises to the forwarding IP addresses, Cloud Routers must advertise the IP address range “35.199.192.0/19” into Quest On-Premises.

5.10.2 Resolution of OnPrem domain from Google Cloud

To resolve on-premises DNS names from Google Landing Zone, a Cloud DNS Forwarding Zone will be created in the External VPC targeting on-premises DNS servers. DNS Forwarding will go over the private dedicated connection (VPN or Cloud Interconnect). If other private domains are required to be resolved from onprem, additional Forwarding Zones for each domain must be created in the External VPC.

Additionally, a DNS Peering Zone (for on-premises DNS names) will be set up and associated with all workload VPCs, setting the External VPC as the peer network. DNS resolution for on-premises will go via External VPC. This way, all workloads in Google will be able to resolve qdx.com names.



Firewall and Routing configurations

DNS traffic should be allowed inside the VPCs and On-premises:

- Quests on-premises firewalls should allow DNS traffic coming from the IP range "35.199.192.0/19" used by Google Cloud DNS (UDP and TCP traffic on port 53)
- On-premises should have a route to "35.199.192.0/19" via the VPN/Interconnect link to Google Cloud (this route can be automatically advertised into Quest via Cloud Router)

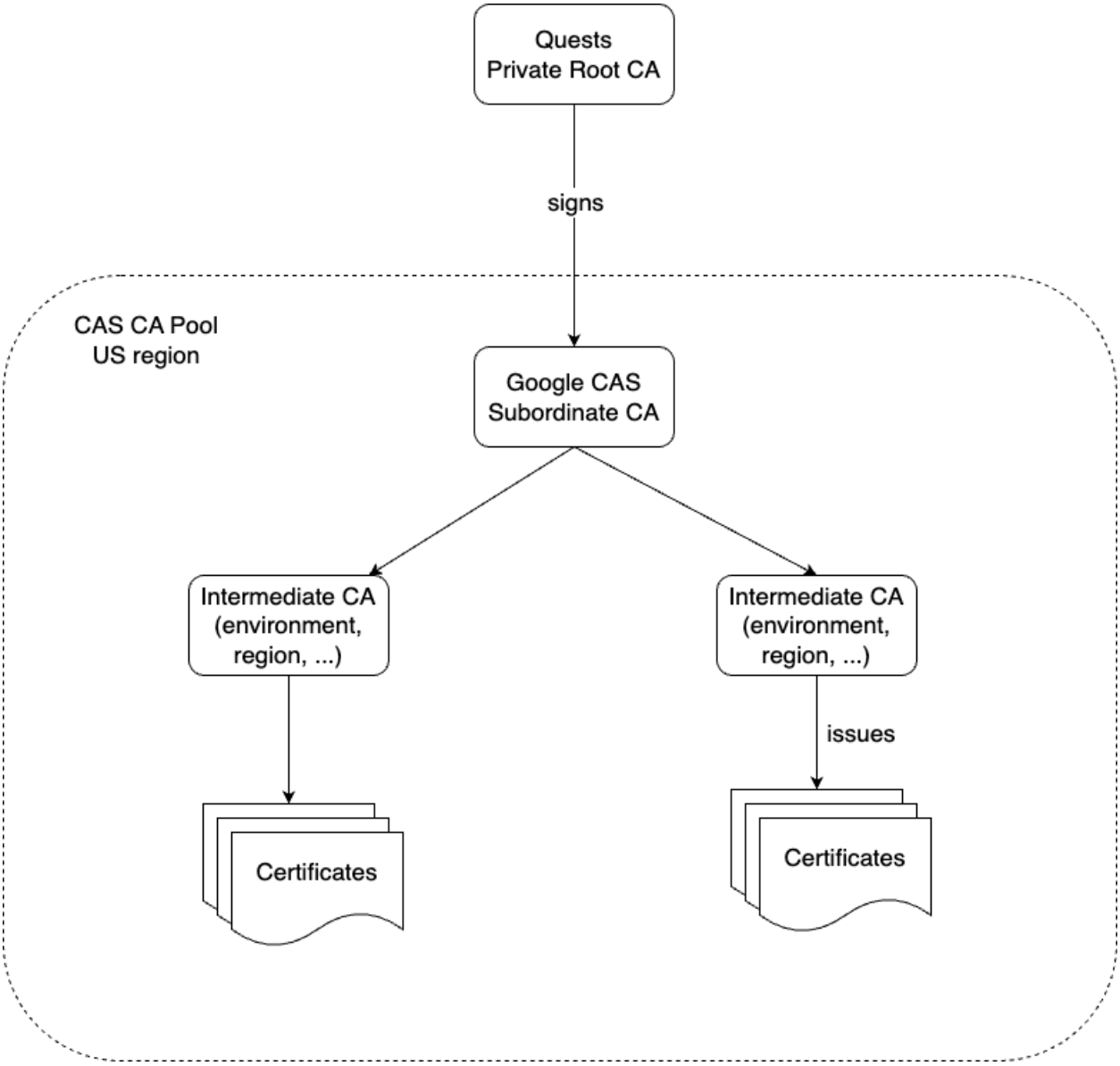
5.12 Private SSL Certificates

Quest requires a native solution to manage private certificates for Google Cloud workloads, that can have CAs serving as subordinate to Quests onprem internal CA. For this purpose, Quest will use Google Cloud Certificate Authority Service (CAS). CAS is a highly available and scalable service offered by Google Cloud that simplifies, automates, and customizes the deployment and management of private Certificate Authorities (CAs).

Google CA Service is intended to be set up as a subordinate CA, signed by Quests private Root CA. From there, intermediate CAs can be configured based on environment, region, or other use cases aligned to Quests operations. Intermediate CAs are then able to sign certificates for internal applications.

Intermediate CAs can be set and separated likely by a few key factors, including:

- Environment (dev, prod, uat)
- Geographical regions
- Other use cases



5.11.1 CA Pool

In CAS, a CA pool is needed to host all the CA resources. A CA pool is the container for CAs and certificates. IAM conditions, issuance policies, and other configurations are set on the CA pool. A CA Pool is bound to a Google Cloud region, which means that resources created by a CA will be stored in that region.
For compliance requirements, Quest will require a CA Pool in the North America region. Additional CA Pools can be created later on in other regions where Quest is operating.

Additionally, a CA Pool can operate in one of two modes, optimized to the type of workloads that are going to use the certificates. CA Pool tier is defined when the CA Pool is created. There are two options:

- DevOps: Focused on high volume, short-lived certificate issuance which is found in microservice-based applications.
- Enterprise: Focused on lower volume, long-lived certificate issuance which is normally found in devices and user identity, where lifecycle management is important.

In case Quests wants to use customer-managed KMS keys (instead of the default Google-managed), Enterprise tier should be used.

5.11.2 Certificate Authority (CA)

At the top of the hierarchy of the CA Pool will be a subordinate CA signed by Quests root CA. The subordinate CA will be trusted by any workload that trusts Quests root CA. You can then use the subordinate CA to issue certificates without needing to reach back to Quests root CA. Or, create other intermediate CAs according to Quests specific needs.

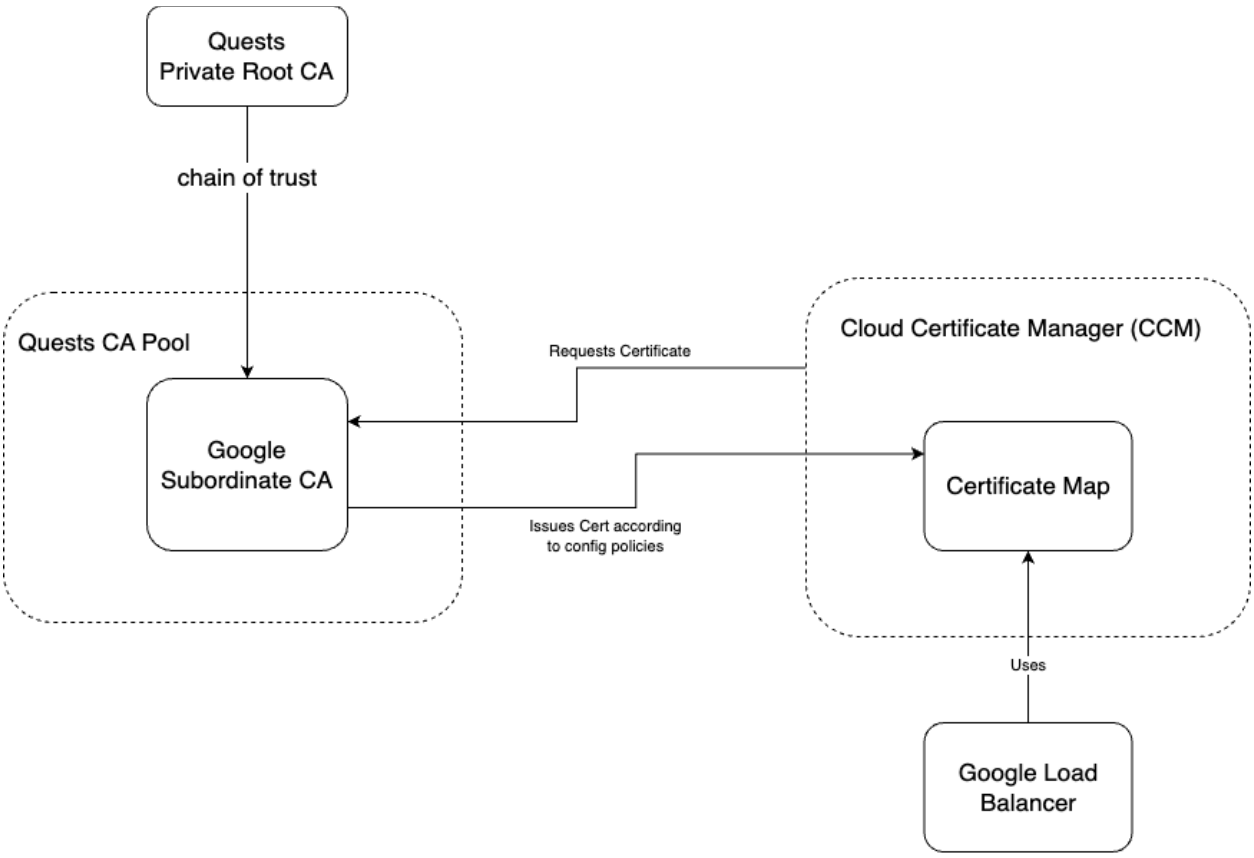
Configuring the subordinate CA in Google CAS is a three-step manual process. The first step is to create a CSR for this CA and download the CSR. Once you have the CSR, it needs to be signed by Quests root CA. Finally, the signed PEM certificate chain is uploaded to activate your subordinate CA.

By default, the subordinated CA will use a Google-managed Cloud KMS signing key. Based on Quests requirements, a custom KSM key might be required. This will allow Quest to choose any of the supported asymmetric signing algorithms.

A GCS bucket must be created for this CA to store CA artifacts, certificates and including CRLs. This bucket should be in the same location as your CA Service resources. You cannot create the Cloud Storage bucket outside the continent where you have created the CA Service resources.

5.11.3 Certificate Manager

Additionally, CAS can be integrated with Google Cloud Certificate Manager (CCM) to simplify the process of managing the lifecycle of private certificates and provisioning private certificates to load balancers before the certificates expire. CA Service contains the CA pool that issues the private certificates while Certificate Manager lets you configure the issuance and provisioning of certificates to your load balancers.



5.11.4 Additional Certificate Manager Considerations

1. Certificate Authority Service (CAS)

Deployment Details: Google best practice for deploying CAS is to start with a single CA pool in a centralized project within your organization. Consider using the "Enterprise" tier for CA pools that issue certificates to other CAs and end-entities, offering higher throughput and availability.

IAC Example (Terraform):

Terraform

```
resource "google_privateca_ca_pool" "default" {
  name     = "my-ca-pool"
  location = "us-central1"
  publishing_options {
    publish_ca_cert = true
    publish_crl     = true
  }
  tier = "ENTERPRISE" # Use Enterprise tier for higher throughput and availability
```

```
}
```

2. Automated Certificate Lifecycle Management

Deployment Details: To leverage automated certificate lifecycle management, ensure that the GCP services you're using (Load Balancers, Kubernetes Engine Ingresses) are configured to integrate with CAS. This is typically done through the respective service's configuration settings. For example, when creating a new HTTPS Load Balancer, you can select a CAS-issued certificate for SSL termination.

3. Internal Load Balancer (ILB) Integration

Deployment Details: When deploying ILBs with CAS, Google recommends using a dedicated subnetwork for your ILBs. This allows you to apply firewall rules specifically to your internal load balancing traffic, further enhancing security. Configure your ILB's backend services to use CAS-issued certificates for secure communication between your instances.

- **IAC Example (Terraform):**

Terraform

```
resource "google_compute_subnetwork" "ilb_subnetwork" {
  name          = "ilb-subnet"
  ip_cidr_range = "10.128.0.0/20"
  region        = "us-central1"
  project       = "your-project-id"
}

resource "google_compute_forwarding_rule" "ilb_forwarding_rule" {
  name                  = "ilb-forwarding-rule"
  ip_protocol           = "TCP"
  load_balancing_scheme = "INTERNAL"
  all_ports              = true
  allow_global_access   = false
  subnetwork            = google_compute_subnetwork.ilb_subnetwork.id
  backend_service        = google_compute_backend_service.default.id
}

resource "google_compute_backend_service" "default" {
  name                  = "ilb-backend-service"
  port_name             = "https"
  protocol              = "TCP"
  health_checks         = [google_compute_health_check.default.id]
  load_balancing_scheme = "INTERNAL"
```

```
  # ... (rest of the backend service configuration)
}
```

4. Programmatic Access and Guardrails

- **Deployment Details:** Google recommends following the principle of least privilege when granting programmatic access to CAS. Create custom IAM roles with only the necessary permissions for each team. This ensures that teams can only perform the actions required for their specific tasks.

- **IAC Explanation:**

    - **Service Account Creation:** The provided Terraform code defines a service account (`team_a_sa`) specifically for "Team A". This dedicated account ensures that their actions are traceable and their permissions are limited.
    - **IAM Binding:** The `google_project_iam_member` resource grants the "Team A" service account the `roles/privateca.certificateManager` role at the project level. This allows the service account to manage certificates within the project.
    - **CA Pool IAM Binding:** The `google_privateca_ca_pool_iam_member` resource grants the same role, but specifically for the `default` CA pool. This ensures the service account can only interact with that specific pool.
    - **Certificate Template with Constraints:** The `google_privateca_certificate_template` resource defines a template named `team-a-template`. The crucial part is the `identity_constraints` block. This block uses a CEL expression to enforce that any certificate issued using this template must have a Subject Alternative Name (SAN) that starts with `*.applicationA.gcp.qdx.com`. This effectively restricts "Team A" to only issuing certificates for their designated domain.
- **IAC Example (Terraform):**

Terraform

```
resource "google_service_account" "team_a_sa" {
  account_id   = "team-a-cas-sa"
  display_name = "Team A CAS Service Account"
}

resource "google_project_iam_member" "team_a_cas_iam" {
  project = "your-project-id"
  role    = "roles/privateca.certificateManager"
  member  = "serviceAccount:${google_service_account.team_a_sa.email}"
}
```

```
resource "google_privateca_ca_pool_iam_member" "team_a_ca_pool_iam" {
  ca_pool = google_privateca_ca_pool.default.name
  role    = "roles/privateca.certificateManager"
  member  = "serviceAccount:${google_service_account.team_a_sa.email}"
}

resource "google_privateca_certificate_template" "team_a_template" {
  name = "team-a-template"
  location = "us-central1"

  identity_constraints {
    cel_expression {
      expression = "subjectAltNames.all(san, san.type == DNS_NAME &&
san.value.startsWith('*.applicationA.gcp.qdx.com'))"
      title = "Restrict FQDN to team A domain"
    }
  }
}
```

5. Monitoring, Alerting, and Reporting

- **Deployment Details:** Google recommends creating a centralized logging sink to collect all CAS-related logs. This allows you to store logs for a longer duration and perform advanced analysis across your organization. Additionally, integrate CAS with your preferred alerting and incident management tools for timely notifications and response.
- **IAC Explanation:**
  - **Alert Policy:** The `google_monitoring_alert_policy` resource defines an alert policy named `cert_expiry_alert`.
  - **Condition:** The `conditions` block specifies the criteria for triggering the alert. In this case, it uses the `privateca.googleapis.com/certificate/lifetime_percentage` metric to check if any certificate's remaining lifetime falls below 20%.
  - **Notification Channel:** The `notification_channels` attribute links the alert policy to a notification channel (defined elsewhere in your Terraform code, likely an email channel in this example). This ensures that you receive notifications when the alert is triggered.
- **IAC Example (Terraform):**

Terraform

```
resource "google_monitoring_alert_policy" "cert_expiry_alert" {
  display_name = "Certificate Expiry Alert"
  combiner = "OR"
  conditions {
    display_name = "Certificates Expiring Soon"
    condition_threshold {
      filter =
"metric.type=\"privateca.googleapis.com/certificate/lifetime_percentage\" AND
resource.type=\"privateca_certificate\""
      threshold_value = 20 # Alert if certificate lifetime is below 20%
      comparison = "COMPARISON_LT"
      duration = "3600s" # 1 hour
      trigger {
        count = 1
      }
    }
  }
  notification_channels =
[google_monitoring_notification_channel.email_channel.name]
}
```

PHASE 2 CERTIFICATE MANAGER OPTIONS

While Google Cloud's Certificate Manager provides a robust foundation for managing private certificates, certain scenarios require custom solutions to extend its capabilities and address specific needs. These solutions, though not "out-of-the-box," enhance Certificate Manager operations by providing automation, flexibility, and deeper integration with your environment.

In the event of a “phase 2” landing zone, if Quest decides to enhance their certificate management posture beyond native capabilities, below are some avenues to explore:

1. Automated Certificate Management for VMs and Instance Groups

- **Challenge:** Certificate Manager doesn't natively automate certificate provisioning for VMs and Instance Groups.
- **Solution:**
  - **Startup Scripts:** Embed scripts within instance templates to automatically generate CSRs, request certificates from Certificate Manager via its API, and install the certificates upon instance startup.
  - **Renewal Scripts:** Utilize cron jobs or other scheduling mechanisms to periodically trigger scripts that check for upcoming expirations, request renewals from Certificate Manager, and install the updated certificates.

- **Google Connection:** These scripts interact directly with the Certificate Manager API using authentication mechanisms like service accounts.

2. Kubernetes Certificate Management

- **Challenge:** While Certificate Manager can integrate with Kubernetes Ingress, managing certificates for individual pods or services within the cluster often requires additional tooling.
- **Solution:** Deploy cert-manager, a popular open-source Kubernetes certificate management controller. cert-manager automates certificate issuance and renewal for various Kubernetes resources by integrating with Certificate Manager as its certificate issuer.
- **Google Connection:** cert-manager uses a service account to authenticate and interact with the Certificate Manager API.

3. Enhanced Monitoring and Alerting

- **Challenge:** Certificate Manager's built-in monitoring might not cover all your needs for comprehensive visibility and alerting.
- **Solution:** Develop custom scripts or tools that:
  - Gather more granular certificate information from various sources.
  - Implement custom logic for analyzing certificate data and identifying potential issues.
  - Trigger alerts through preferred channels (e.g., email, Slack, PagerDuty) based on specific conditions or thresholds.
- **Google Connection:** These scripts or tools can leverage Google Cloud's monitoring and logging APIs (e.g., Cloud Monitoring API, Cloud Logging API) to collect data and integrate with existing monitoring infrastructure.

4. Certificate Revocation

- **Challenge:** While Certificate Manager supports certificate revocation, enforcing CRL checks and providing comprehensive monitoring for revocation events often require custom implementations.
- **Solution:**
  - Develop mechanisms to ensure applications and services perform CRL checks before establishing secure connections.
  - Create custom monitoring and alerting systems to track revocation events and identify any potential security issues.
- **Google Connection:** Integrate with Certificate Manager's CRL distribution point and potentially leverage Cloud Logging or Security Command Center for enhanced monitoring and analysis.

6. Instrumentation

Google Cloud Operations Suite will be used for centralized logging, monitoring, and auditing at Quest, and logs and metrics will be exported to Splunk and Dynatrace for enhanced analytics, visualization, and alerting capabilities.

Key Considerations:

1. Log Destination and Retention:
    a. Splunk, Wiz, and Dynatrace: Logs will be sent to Splunk and Dynatrace to support real-time monitoring and analysis.
    b. Long-Term Storage: For extended retention, logs will also be archived in a Google Cloud Storage (GCS) bucket.
2. SIEM Logging and GDPR, HIPAA, and PCI Compliance:
    a. Define a data retention policy for cloud audit logs (Audit logs help you answer "who did what, where, and when?) to be stored for six years to be HIPPA compliant.
    b. Allow on the web page of the client the right to erasure to proceed with the data deletion on google cloud by the Quest request to enforce the right to erasure to be GDPR compliant.
    c. Implement an effective SIEM daily log monitoring to be PCI-DSS compliant, for example if the SIEM is splunk you can create automate it daily reports of the following:
        i. All security events
        ii. Logs of all system components that store, process, or transmits Card Holder Data
        iii. Logs of all critical system components
        iv. Logs of all servers and system components that perform security functions (for example, firewalls, intrusion-detection systems/intrusion-prevention systems (IDS/IPS), authentication servers, e-commerce redirection servers, etc.)
        v. Review logs from all other system components periodically based on the organization's policies and risk management strategy, as determined by the organization's annual risk assessment
3. Access Control:
    a. Security and Operations Teams: Require access to all logs across the entire Google Cloud Platform (GCP) organization.
    b. Project-Based Access: Other teams will have log access based on the projects they own, ensuring adherence to the principle of least privilege.
4. Budget Monitoring:

a. Budget Alerts by Tags: Budget alerts will be set up in GCP using tags to monitor and control costs effectively.
b. Metric Pricing: Review and align with Google Cloud metric pricing to optimize costs for monitoring and alerts.

5. Security and Compliance Requirements:
   a. Data Sensitivity and Project Segregation: Quest has stringent security requirements that necessitate separating logs containing Protected Health Information (PHI) and other sensitive data into dedicated, controlled projects. This approach ensures that sensitive data is isolated and managed according to security and compliance standards.
   b. Regional Segregation for Compliance: To meet compliance requirements, Quest requires that logs be segregated between the EU and US regions. This ensures that log data remains within the respective geographic regions, aligning with data residency and regulatory standards specific to each location.

Additional Details

The "security events" mentioned above encompass a broad range of security-related events, including but not limited to:

```
* **Access control events:**  Successful and failed login attempts, privilege escalations, access to sensitive resources.
* **Network events:**  Suspicious network traffic, connection attempts from blocked IPs, firewall denials.
* **System events:**  Unexpected system shutdowns or restarts, service failures, unusual process activity.
* **Application events:**  Application crashes, errors, and security-related events like input validation failures.
* **Data events:**  Data exfiltration attempts, unauthorized data access, data modification.
```

"Periodically" in point 2 refers to a flexible timeframe determined by the organization's policies, risk management strategy, and annual risk assessment. This could mean:

```
* **Weekly:** For systems with moderate sensitivity.
* **Monthly:** For less critical systems.
* **Quarterly or annually:** For systems with low risk profiles.
```

The specific timeframe for each system component will be documented in the organization's security policies and reviewed during the risk assessment.

Data sensitivity is determined through a data classification process, which typically involves:

```
* **Data discovery and inventory:** Identifying all data stored and
processed by the organization.
* **Data analysis and labeling:**  Analyzing the data to understand its
content and purpose.
* **Risk assessment:** Evaluating the potential impact of unauthorized
access or disclosure.
* **Assigning sensitivity levels:**  Categorizing data into levels such as
public, confidential, restricted, or highly sensitive.
```

This process will be conducted in collaboration with data owners and stakeholders across the organization.

File Integrity Monitoring

- **GCP-Native FIM:** We'll leverage Google Cloud's Security Command Center, specifically its File Integrity Monitoring feature for Google Kubernetes Engine (GKE). This will help us detect unauthorized modifications to container images and ensure the integrity of your containerized applications.
- **Cloud Audit Logs:** We'll utilize Cloud Audit Logs to track file system access and modifications across various GCP services. This provides a comprehensive audit trail for investigating potential security incidents and demonstrating compliance.

Example FIM Events:

```
# Example FIM Events


* **File hashes changes:**  Detecting alterations in critical file hashes, indicating unauthorized modifications.

* **Permission changes:**  Identifying modifications to file permissions and ownership, which could signal privilege escalation attempts.

* **Unexpected file creation or deletion:** Monitoring for the creation or deletion of files in sensitive directories.

* **Configuration file changes:** Tracking modifications to system and application configuration files.
```

6.1 High-level Logging & Monitoring Decisions

This section details the procedures for Logging & Monitoring logs within the Google Cloud environment. It provides an overview of the log type to enable, where to route and the retention period based on Quest Diagnostics' requirements.

Destination to send logs	Purpose	Retention Period
Google Cloud Storage (GCS) Location: Multi-Region (US)*	Provides log data storage in Cloud Storage for long-term retention. Log entries are stored as JSON files For regulatory purposes, you can apply Cloud Storage Bucket Lock	6 years
BigQuery	BigQuery provides storage for log entries in datasets. BigQuery's big data analysis capabilities can be used to analyze the stored logs	6 years
Splunk	Used to send logs to Splunk Instance	6 years
Dynatrace	Used to send logs to Dynatrace tenants	90 days

*A multi-region is a large geographic area, such as the United States, that contains two or more geographic places providing data redundancy across regions (asynchronous).
***To set project log retention period, go to the Logging page in your GCP console, navigate to "Logging" -> "Logs Storage."**

- **Select or create a log bucket: Log buckets are containers for your logs. Choose the bucket you want to configure or create a new one.**
- **Set the retention period: When creating or editing a bucket, you'll find a "Retention period" field. Specify the number of days you want to keep the logs (from 1 to 3650 days).**
- **Click "Create" or "Update": Save your changes, and GCP will automatically enforce the retention policy on that bucket.**

2. Using the gcloud command-line tool:

- **List your log buckets: Use the command `gcloud logging buckets list` to see your existing buckets.**
- **Update the bucket: Use the following command to set the retention period:**

```
gcloud logging buckets update BUCKET_NAME --location=LOCATION --
retention-days=NUMBER_OF_DAYS
```

- **Replace `BUCKET_NAME` with the name of your bucket, `LOCATION` with the bucket's location (e.g., `global`), and `NUMBER_OF_DAYS` with your desired retention period.**

Cloud Storage logs are stored in JSON format to leverage its specific advantages within the Google Cloud Platform ecosystem:

- **Integration with BigQuery: JSON facilitates integration with BigQuery allowing for analysis of Cloud Storage access patterns, identifying trends, and generating security insights.**
- **Cloud Monitoring and Logging: The structured nature of JSON allows Cloud Monitoring and Logging to ingest and process these logs efficiently. This enables real-time monitoring, alerting on anomalies, and visualizing access trends within Cloud Storage.**
- **Dataflow Processing: JSON's structured format is ideal for processing within Dataflow, enabling the creation of pipelines to aggregate, transform, and analyze Cloud Storage log data for security and operational purposes.**
- **Programmatic Access: Google Cloud client libraries can parse JSON-formatted logs, allowing developers to programmatically access and analyze this data for custom applications and security tools.**

6.1.1 Logging & Monitoring Description

No.	Name	Log type	Description
1	Project Level Logs	• Admin Activity audit logs • Data Access audit logs • System Event audit logs • Policy Denied audit logs • Access Transparency audit logs • Cloud DNS Logs • Cloud NAT Logs • Firewall Rules Logs • Cloud IDS Logs • HTTP(S) LB Logs • VM Syslog • VM Windows Event Logs	Logs generated at the project-level
2	Project Sinks (Default)	Any log entry that isn't stored in the _Required bucket is routed by the _Default sink	Default project-level sink. Logging routes logs with the Log Router by using sinks
3	Project Log Bucket (Default)	Any log entry that isn't stored in the _Required bucket is routed by the _Default sink to the _Default bucket, unless you disable or otherwise edit the _Default sink.	Default project-level log storage/bucket. Log sinks send default logs to log storage/bucket at each resource hierarchy
4	Folder Level Logs	• Admin Activity audit logs • Data Access audit logs • System Event audit logs • Policy Denied audit logs • Access Transparency audit logs	Logs generated at the folder-level
5	Organization Level Logs	• Admin Activity audit logs • Data Access audit logs • System Event audit logs • Policy Denied audit logs • Access Transparency audit logs	Logs generated at the organization-level
6	Cloud Identity Audit Logs	• Admin Audit logs • Enterprise Groups Audit logs • Login Audit logs • OAuth Token Audit logs • SAML Audit logs	Logs generated from Cloud Identity
7	Aggregated Log Sink Centralized Storage Bucket	Aggregated log sink to route the following logs: • Cloud Audit Logs - Admin Activity • Data Access audit logs • Policy Denied audit logs • Access Transparency Logs • Cloud DNS Logs • Cloud NAT Logs • Firewall Rules Logs • HTTP(S) LB Logs	Aggregated sinks combine and route log entries from the Google Cloud resources contained by an organization or folder

No.	Name	Log type	Description
8	Centralized GCS Storage Bucket - Long Term Retention Location: Multi-Region (US)	• Cloud Audit Logs - Admin Activity • Data Access audit logs • Policy Denied audit logs • Access Transparency Logs • Cloud DNS Logs • Cloud NAT Logs • Firewall Rules Logs • HTTP(S) LB Logs	Centralized Storage Bucket (Google Cloud Storage). Used to store logs for long-term retention.
9	Bucket Lock (Optional)	N/A	Allows you to configure a data retention policy for a Cloud Storage bucket that governs how long objects in the bucket must be retained. The feature also allows you to lock the data retention policy, permanently preventing the policy from being reduced or removed.
10	Aggregated Log Sink - BigQuery	Aggregated log sink to route the following logs: • Cloud Audit Logs - Admin Activity • Data Access audit logs • Policy Denied audit logs • Access Transparency Logs • Cloud DNS Logs • Cloud NAT Logs • Firewall Rules Logs HTTP(S) LB Logs	Aggregated sinks combine and route log entries from the Google Cloud resources contained by an organization or folder
11	BigQuery	Aggregated log sink to route the following logs: • Cloud Audit Logs - Admin Activity • Data Access audit logs • Policy Denied audit logs • Access Transparency Logs • Cloud DNS Logs • Cloud NAT Logs • Firewall Rules Logs • HTTP(S) LB Logs	BigQuery provides storage for log entries in datasets. BigQuery's big data analysis capabilities can be used to analyze the stored logs
12	Aggregated Log Sink - SIEM Export	Aggregated log sink to route the following logs: • Cloud Audit Logs - Admin Activity • Policy Denied audit logs • Access Transparency Logs • Cloud DNS Logs • Cloud NAT Logs • Firewall Rules Logs • HTTP(S) LB Logs	Aggregated sinks combine and route log entries from the Google Cloud resources contained by an organization or folder
13	Pub/Sub - Pull-based (Splunk)	N/A	Pub/Sub - to allow data to be fetched from Google Cloud APIs through the Splunk Add-on for Google Cloud Platform.

No.	Name	Log type	Description
14	Security Command Center - Premium (SCCP)	N/A	SCCP is a managed service for Event Threat Detection and Security Health Analytics, to detect security threats in the Google Cloud environment.
15	SCCP IAM	N/A	SCC uses IAM to control access to resources at different levels of your resource hierarchy. You use IAM roles to control who can do what with assets, findings, and security sources in your SCC environment.
16	Pub/Sub - SCCP Notifications API	SCCP Notification Pub/Sub Filter: state="ACTIVE" AND NOT mute="MUTED"	SCC API notifications feature sends information to a Pub/Sub topic to provide findings updates and new findings
17	Splunk (SIEM)	N/A	SIEM tool
18	OneAgent Log Forwarder (Dynatrace)	Application logs Operations logs (non-security and non-audit related logs)	Allow Dynatrace OneAgent running on VMs and GKE clusters to forward logs to Dynatrace SaaS tenant to further enhance the context for monitoring and analysis
19	Pub/Sub - Pull-based (Dynatrace)	N/A	Pub/Sub - to allow data to be fetched from Google Cloud APIs through the Dynatrace ActiveGate running on Google Cloud Platform.
20	Aggregated Log Sink - Dynatrace Export	Application logs Operations logs (non-security and non-audit related logs)	Aggregated sinks combine and route log entries from the Google Cloud resources contained by an organization or folder
21	GCP Service Logs Supported by Dynatrace	GCP Supported Service Logs	Dynatrace log and metric integration running on GKE Autopilot Cluster will collect GCP Service Logs supported by Dynatrace from Pub/Sub

Note: The above table is based on the logs that will be captured and monitored. Additional logs may exist at each level, depending on the Google Cloud services that will be used.

6.1.2 Recommended Logs to enable

Domain	Log type	Enabled by default	Recommendation	Quest	Quest Details	Retention Period

				Comment		
	Cloud Audit Logs - Admin Activity	Enabled by default	Enable at the Organization Level	Send to Splunk for ReliaQuest and Cloud Operations monitoring, and to Wiz for CNAPP enrichment	Admin Activity logs are critical for security monitoring. These logs in Splunk will enable ReliaQuest (RQ) to create custom alerts and facilitate security investigations by the SOC and Quest Security Teams. Additionally, Cloud Operations and DevOps teams can utilize them to troubleshoot access issues and deployment errors. Wiz also requires these activity logs to enrich CNAPP findings and provide recommendations on excessive IAM access. The logs must be separated for the US and EU regions due to separate Splunk instances and SOC teams, ensuring GDPR compliance.	8 years
Audit Logs	**Cloud Audit Logs - Data Access**	Not enabled by default (except for BigQuery Data Access audit logs)	Enable at the Organization Level*	Enable only for production projects or projects containing sensitive data (PHI, PII, PCI, etc.). Send to Splunk for security monitoring.	For production projects and cases where sensitive data is stored in non-production environments, data access logs should be enabled. Forwarding these logs to Splunk allows ReliaQuest to monitor for unauthorized data access or potential exfiltration attempts. We need guidance from Google on how to enforce data access logging for projects falling into these categories.	8 years
	Cloud Audit Logs - System Events	Enabled by default	Enable at the Organization Level	Send to Splunk for Security and	For Cloud Operations the logs might also need to be in BQ for analysis. These logs also need to be segregated between US and EU regions.	8 years

				Operations access.		
	Cloud Audit Logs - Policy Denied	Enabled by default	Enable at the Organization Level	Send to Splunk for Security and Operations access.	Policy Denied logs capture attempts to access resources blocked by policy constraints. Sending these to Splunk allows ReliaQuest (RQ) to identify and investigate potentially malicious activities where users or services face repeated access denials. This information can also assist Cloud Operations in resolving policy-generated access issues. As with other logs, these need to be segregated between US and EU regions.	8 years
	Access Transparency Logs	Not enabled by default	Enable at the Organization Level	Send to Splunk for Security and Operations access.	As with other logs, these need to be segregated between US and EU regions.	8 years
Network Logs	**Cloud DNS Logs**	Not enabled by default	Enable at the Organization Level	Send to Splunk for RQ and Cloud Operations	ReliaQuest can use these logs in Splunk to identify suspicious DNS queries that may indicate compromised resources or domain-based threats such as malware communications or DNS tunneling. They can help Cloud Operations troubleshoot network connectivity issues. As with other logs, these must be segregated between US and EU regions.	8 years
	Cloud NAT Logs	Not enabled by default	Enable at the Organization Level	Send to BQ for analysis	While useful, NAT logs can generate a high volume of data, CDOE and Cloud Operations will need to decide if they can stay in BQ or if we need them in Splunk	8 years

	Firewall Rules Logs	Not enabled by default	Enable at the Organization Level	Send to Splunk for RQ and Cloud Operations	Firewall logs are crucial for security monitoring. For firewall rules protecting sensitive or production environments, forwarding logs to Splunk enables RQ to detect and respond to threats.	8 years
	VPC Flow Logs	Not enabled by default	Enable at the Organization Level	Send to BQ for analysis for production or any environment with access to sensitive data.	VPC Flow Logs generate large volumes of data. Analyzing them in BQ might be more cost-effective. If we deploy an NDR tool this might not be needed.	8 years
	HTTP(S) LB Logs	Not enabled by default	Enable at the Organization Level	Send to Splunk for RQ and Cloud Operations	These logs are essential for detecting web-based attacks and monitoring traffic patterns. ReliaQuest can leverage them in Splunk to identify anomalies such as DDoS attacks, suspicious request patterns, or attempts to exploit web vulnerabilities.	8 years
Compute Engine Logs**	**VM Syslog**	Not enabled by default	Enable at the Organization Level	Send to Splunk using the Splunk agent		
	VM Windows Event Logs	Not enabled by default	Enable at the Organization Level	Send to Splunk using the Splunk agent		

	Admin audit log	Not enabled by default	By default, Cloud Identity (Google Workspace) is not a part of Cloud Logging. Google recommends enabling sharing Google workspace data with Google Cloud and aggregate to BQ and Storage for further investigation and compliance purposes.	Send to Splunk for RQ access	These logs track administrative actions in Cloud Identity and Google Workspace. ReliaQuest can monitor for unauthorized changes, privilege escalations, or policy violations, which are vital for security.	8 years
Cloud Identity Logs (Workspace)	**Login audit log**	Not enabled by default	By default, Cloud Identity (Google Workspace) is not a part of Cloud Logging. Google recommends enabling sharing Google workspace data with Google Cloud and aggregate to BQ and Storage for further investigation and compliance purposes.	Send to Splunk for RQ access	Monitoring login activities helps detect compromised accounts and unauthorized access attempts. RQ can analyze these logs in Splunk to identify suspicious login patterns or brute-force attempts.	8 years
	Groups audit log	Not enabled by default	By default, Cloud Identity (Google Workspace) is not a part of Cloud Logging. Google recommends enabling sharing Google workspace data with Google Cloud and aggregate to BQ and Storage for	Send to Splunk for RQ access	TBD	8 years

		further investigation and compliance purposes.			
OAuth Token audit log	Not enabled by default	By default, Cloud Identity (Google Workspace) is not a part of Cloud Logging. Google recommends enabling sharing Google workspace data with Google Cloud and aggregate to BQ and Storage for further investigation and compliance purposes.	Send to Splunk for RQ access	TBD	8 years
SAML audit log	Not enabled by default	By default, Cloud Identity (Google Workspace) is not a part of Cloud Logging. Google recommends enabling sharing Google workspace data with Google Cloud and aggregate to BQ and Storage for further investigation and compliance purposes.	Send to Splunk for RQ access	TBD	8 years

GCP Service Operations Logs	**Google Cloud AlloyDB**	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: alloydb_database, alloydb_instance	90 days
	Google App Engine	Enabled by default In Dynatrace	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: gae_app	90 days
	Google Cloud Functions	Enabled by default In Dynatrace	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: cloud_function	90 days
	Google Cloud Router	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: nat_gateway	90 days
	Google Cloud Run	Enabled by default In Dynatrace	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: cloud_run_revision	90 days

Google Cloud Storage	Enabled by default In Dynatrace	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: gcs_bucket	90 days
Google Cloud Tasks	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: cloud_tasks_queue	90 days
Google Cloud Composer	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: cloud_composer_environment	90 days
Google Compute Engine	Enabled by default In Dynatrace	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: autoscaler, gce_autoscaler, gce_instance_group, gce_instance, tpu_worker	90 days
Google Cloud Storage Transfer Service	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: storage_transfer_job	90 days

Google Cloud Dataproc	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: cloud_dataproc_cluster	90 days
Google Cloud Hybrid Connectivity	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: gce_router, vpn_gateway	90 days
Google Kubernetes Engine	Enabled by default In Dynatrace	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: k8s_cluster, k8s_container, k8s_node, k8s_pod	90 days
Google Cloud Load Balancing	Enabled by default In Dynatrace	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: http_load_balancer, internal_http_lb_rule, internal_network_lb_rule, network_lb_rule, tcp_ssl_proxy_rule	90 days
Google Cloud Memorystore	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: redis_instance	90 days

Google Cloud Network Security	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: network_security_policy	90 days
Operations: Cloud Monitoring and Logging	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: uptime_url	90 days
Google Cloud Pub/Sub	Enabled by default In Dynatrace	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: pubsub_snapshot, pubsub_subscription, pubsub_topic	90 days
Google Cloud Spanner	Not enabled by default	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: spanner_instance	90 days
Google Cloud SQL	Enabled by default In Dynatrace	Create Sink at Organization Level to Pub/Sub destination for Dynatrace	Dynatrace to capture via log integration	Entities: cloudsql_database	90 days

*For Data Access logs, in order to limit cost and management overhead, it's recommended to:
- Prioritize relevant services, projects, subset of data access operations, relevant resources, and exclude specific principals

**Enable Compute engine logs if Virtual Machines are to be deployed

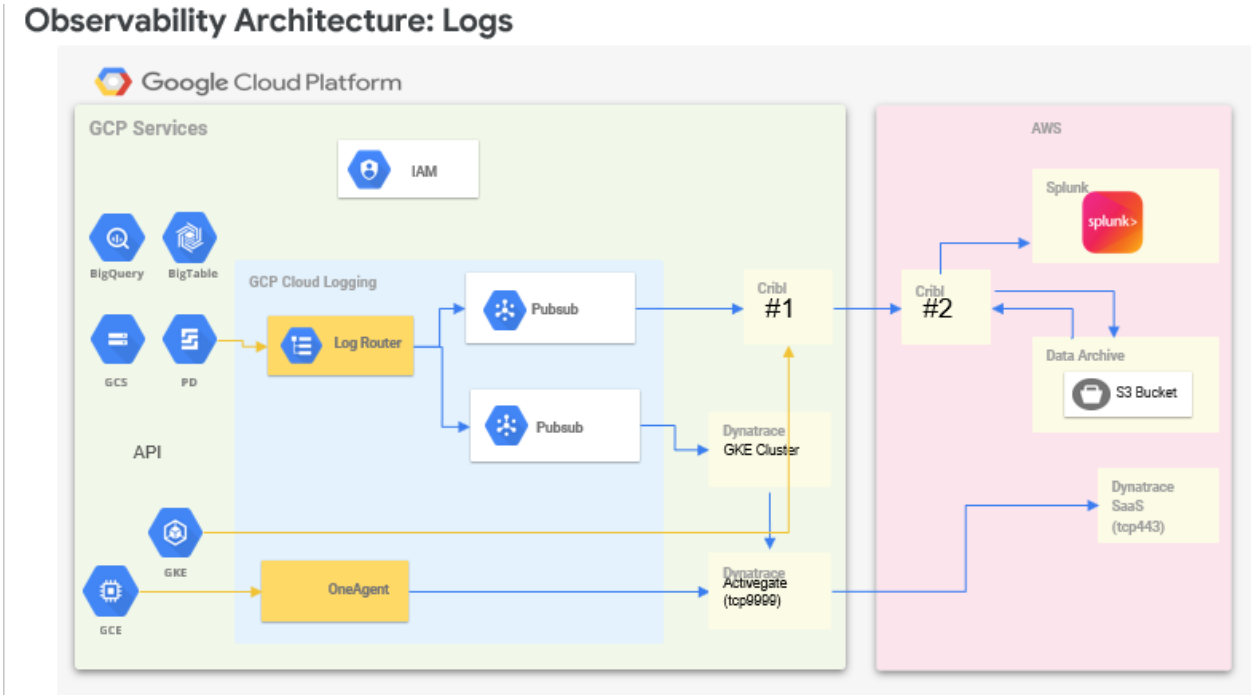
Note: Logs designated as exports at the Organization level are assumed to have the "included-all-children" feature enabled, which accounts for all logs of that type in the Organization.

Reference Architecture here

Configure Cribl Stream to Receive Data from Pub/Sub

https://docs.cribl.io/stream/sources-google_pubsub/

[Design Diagram for Logs]



6.1.3 Monitoring Framework (Dynatrace)

Introduction

Dynatrace is an Application Performance Monitoring (APM) tool that monitors application performance end to end. This includes:

- **Frontend Monitoring**
  - **Real User Monitoring (RUM)** - with the implementation of Real User Monitoring (RUM) into the frontend application web pages, we can gather end user performance metrics for page loads, XHR actions, and errors (HTTP, Custom, and JavaScript).
  - **User Session Query Language (USQL)** - using a SQL like language, we can create queries to further analyze and create dashboards to identify things like those pages with response times greater than 5 seconds, etc.
  - **Synthetic Monitoring** - we can setup browser clickpaths and HTTP monitors that allow us to continuously test endpoints and run through transactional tests against an application and then alert the teams based on defined criteria.
  - **Session Replay** - by enabling session replay, we can replay the actions performed by a user while analyzing the user session captured by Dynatrace
  - **Privacy Note:** It is extremely important that we follow the privacy and masking guidelines as well as any requirements for PHI, HIPPA, and PCI described by Quest Privacy and compliance policies. Global masking is enabled in each Dynatrace environment for IP addresses, query strings, etc. Session replay is set to Mask All for capture and playback
- **Middle Tier**
  - **Services and API Layer** - with Dynatrace in full stack mode, we gain insight and details around the performance with request made at the service or API layer. This includes viewing the Service Flow and the Distributed Traces for the requests. Custom metrics, Key requests, and multidimensional views can be established and created. These additional configuration settings will allow us to further enhance the anomaly detection thresholds and custom alerts
  - **Infrastructure** - The Dynatrace OneAgent will capture infrastructure metrics and alert on thresholds being reached, some of these metrics include:
    - Availability
    - Monitoring unavailable
    - CPU
    - Memory
    - Disk
    - Network Connectivity
- **Backend Tier**

  - **Database Calls** - by monitoring the calling processes from the service layer, we can view and analyze the SQL Queries, Transactions, and Modifications. Response time, throughput, and errors can be further reviewed.
  - **Oracle Extension** - for Oracle databases, to gather most time-consuming queries and Oracle DB health
- **Cloud**
  - **Kubernetes** – Ability to monitor clusters, nodes, pods, processes, services, etc
  - **Google Cloud Services**
    - Google Cloud AI Platform monitoring (deprecated)
    - Google Cloud AlloyDB monitoring
    - Google Cloud APIs monitoring
    - Google Cloud Apigee monitoring
    - Google App Engine with Operations suite metrics monitoring
    - Google Cloud Assistant Smart Home monitoring
    - Google BigQuery monitoring
    - Google Cloud Bigtable monitoring
    - Google Cloud DNS monitoring
    - Google Cloud Functions monitoring
    - Google Cloud IoT Core monitoring (deprecated)
    - Google Cloud Router monitoring
    - Google Cloud Run monitoring
    - Google Cloud Storage monitoring
    - Google Cloud Tasks monitoring
    - Google Cloud Composer monitoring
    - Google Compute Engine with Operations suite metrics monitoring
    - Google Cloud Data Loss Prevention monitoring
    - Google Cloud Storage Transfer Service monitoring
    - Google Cloud Dataflow monitoring
    - Google Cloud Dataproc monitoring
    - Google Cloud Firestore in Datastore mode monitoring
    - Google Cloud Filestore monitoring
    - Google Cloud Firebase monitoring
    - Google Cloud Firestore monitoring
    - Google Cloud Hybrid Connectivity monitoring
    - Google Kubernetes Engine monitoring
    - Google Cloud Load Balancing monitoring
    - Google Managed Microsoft AD monitoring
    - Google Cloud Memorystore monitoring
    - NetApp on Google Cloud monitoring
    - Google Cloud Network Security monitoring

- Operations: Cloud Monitoring & Logging
- Google Cloud Pub/Sub monitoring
- Google Cloud Pub/Sub Lite monitoring
- Google Cloud Spanner monitoring
- Google Cloud SQL monitoring

Environments
Quest Diagnostics is currently using the Dynatrace SaaS environment which is hosted in AWS.

US Tenants
- Prod: https://ucg59307.live.dynatrace.com/
- Non-Prod: https://vbk56183.live.dynatrace.com/

EU Tenants (Ireland)
- Prod: https://xwd88192.live.dynatrace.com/
- Non-Prod: https://whm98497.live.dynatrace.com/

Google Cloud Integration

The Dynatrace infrastructure and IAM roles and policies should be automated using Terraform with the appropriate CI/CD pipelines built

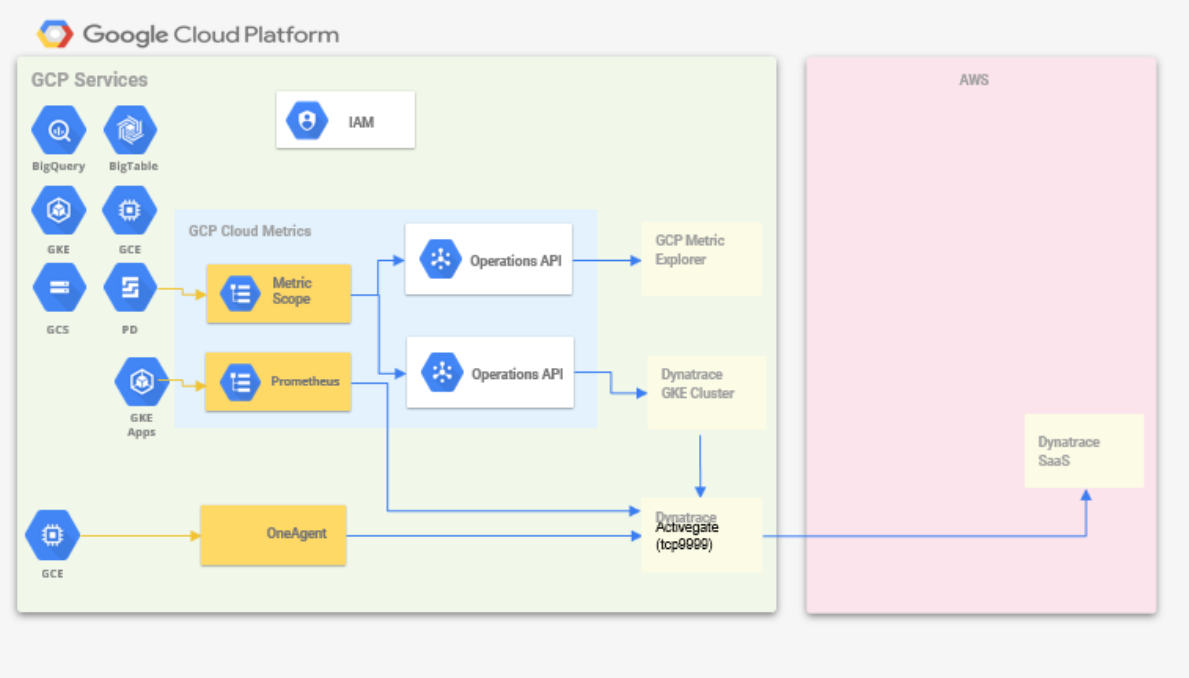
For each non-prod and prod environment in each region:

- Project details:
  - Will be under Global Infra Shared Services
  - US
    - Non-Prod - prj-dynatrace-d-us-v1
    - Prod - prj-dynatrace-p-us-v1
  - EU
    - Non-Prod - prj-dynatrace-d-eu-v1
    - Prod - prj-dynatrace-p-eu-v1
- GKE Autopilot cluster for Operation Suite metrics and logs integration (https://docs.dynatrace.com/docs/ingest-from/google-cloud-platform/gcp-integrations/gcp-guide/deploy-k8)
  - GitHub repo for Helm deployment package
  - dynatrace-gcp-monitor-helm-deployment-role.yaml for permissions
  - Values.yaml for custom settings and configuration
  - Auto-scaling in place for log containers
  - Metrics and logs to flow through ActiveGate rather than direct to SaaS tenant

- Dynatrace ActiveGate for routing and monitoring of OneAgent metrics as well as extensions. Need to have auto-scaling 1min:5max (recommended VM Linux 8 Core / 64GB RAM)
- 2 Dynatrace Private Synthetic ActiveGates for internal synthetic monitors and certificate health checks. No auto-scaling (recommended VM Linux 8 core / 32 GB)
- Dynatrace OneAgent installed on
    - VMs
    - ActiveGates
- Dynatrace Operator for GKE clusters running in Cloud Native mode
    - GitHub repo for:
        - Operator image
        - Containerized ActiveGate image
        - OneAgent image
        - Code modules for CSI Driver
        - Helm chart and Values.yaml
        - Images tagged with “latest”
    - Container registry – artifact
    - WIZ.io scanning through GitHub Action pipeline

[Design Diagram for Metrics]

Observability Architecture: Metrics



7. Data management

The purpose of data management design is to define the shared responsibility of both Google and the Quest in the appropriate and compliant collection, storage, and processing of information, particularly sensitive data such as personally identifiable information (PII).

7.1 Cloud KMS

Encryption

Encryption at Rest

Data in Google Cloud Platform is broken into subfile chunks for storage, and each chunk is encrypted at the storage level with an individual encryption key. The key used to encrypt the data in a chunk is called a data encryption key (DEK). Because of the high volume of keys at Google, and the need for low latency and high availability, these keys are stored near the data that they encrypt. The DEKs are encrypted with (or "wrapped" by) a key encryption key (KEK). Quest can choose which key management solution they prefer for managing the KEKs that protect the DEKs that protect their data.

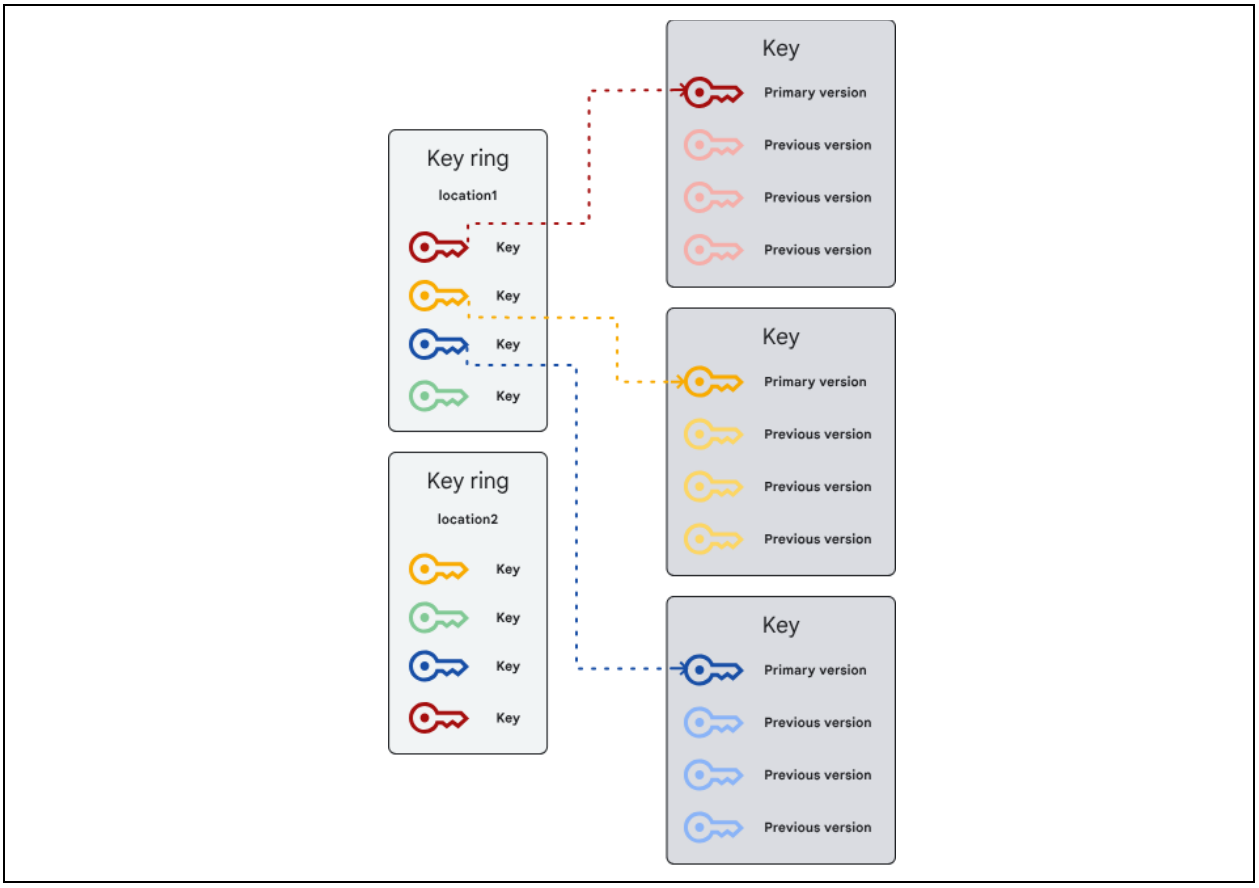
There are three main encryption-at-rest options available to Quest Diagnostics which are detailed in the table below:

Encryption-at-Rest Options

Encryption at Rest Option	Description	Supported Services
Default Encryption	• Data is automatically encrypted prior to being written to disk • Each encryption key is itself encrypted with a set of master keys • Keys and encryption policies are managed the same way, in the same keystore, as for Google's production services	All Services. Granularity varies by Services
Customer-managed encryption keys (CMEK) using Cloud KMS	• Create, rotate, automatically rotate and destroy symmetric encryption keys	Multiple Services. See Supported CMEK Services
Customer-supplied encryption keys (CSEK)	• Use Quest Diagnostics' generated encryption keys as part of services on Google Cloud Platform • Google uses the key in memory and does not write it to storage • You provide the keys as part of API service calls	Compute Engine Cloud Storage

Cloud KMS Components

Cloud KMS, and Default Encryption, uses envelope encryption to provide multiple layers of keys for the encrypted data. Using multiple layers of keys gives Google the ability to store and encrypt data at scale.



Cloud KMS - Key Hierarchy

Cloud KMS stores keys in a key hierarchy designed for ease, with access to resources in the key hierarchy governed by Cloud IAM. The following shows the main levels of a Cloud KMS key hierarchy:

Cloud KMS - Components Descriptions

Component	Description
Key Ring	A key ring is a grouping of keys for organizational purposes. A key ring belongs to a GCP Project and resides in a specific location. Keys inherit permissions from the key ring that contains them. Grouping keys with related permissions together in a key ring allows you to grant, revoke, or modify permissions to those keys at the key ring level, without needing to act on each one individually.
Key	A key is a named object representing a cryptographic key used for a specific purpose. The key material, the actual bits used for encryption, can change over time as new key versions are created. The key is considered the Key Encryption Key (KEK) and also the resource to which IAM access is bound.

Key Version	A key version represents the key material associated with a KEK at some point. Each key can have arbitrarily many versions, but must have at least one. Versions are numbered sequentially, starting with 1.

7.2 Cloud KMS Separation of Duties

Cloud Key Management Service (Cloud KMS) organizes resources in a hierarchical structure, with keys nested within key rings, and key rings residing within projects. This structure facilitates granular management and access control over resources.

For production environments, it's advisable to use Cloud KMS predefined roles or create custom roles tailored to your specific requirements. Basic roles like owner, editor, and viewer don't distinguish between key management and cryptographic operation permissions, so they're not recommended.

To further enhance security, align with the principle of least privilege by storing your keys in a separate Google Cloud project from the data they protect. This prevents users with basic or highly privileged roles in one project from gaining unauthorized access to keys in another project.

Resource hierarchy
The scope of an IAM role is contingent on the level of the resource hierarchy at which it is granted. For instance, the Cloud KMS CryptoKey Encrypter role (roles/cloudkms.cryptoKeyEncrypter) would have the following effective capabilities if granted at the different levels of the hierarchy, as shown in the table below:

Cloud KMS - Effective Capabilities

Resource Hierarchy	Capability
Organization	Encrypt using all keys in all projects in the organization
Folder	Encrypt using all keys in all projects in the folder
Project	Encrypt using all keys in the project
Key ring	Encrypt using all keys on the key ring
Key	Encrypt using only that key

Note: You can manage access to keys or key rings, but not to individual key versions.

8. Security

The purpose of security design is to define the Google Platform configuration required to prevent unauthorized access to Quest resources on Google Cloud, the resiliency of deployments to attacks, and the prevention/notification of platform misconfigurations. Part of the design decisions impacting security is covered in other sections. For further details, refer to the “Identity Management” and “Networking” sections.

8.1 GCP Platform security

Google provides many protections to its GCP customers, however security of workloads running in GCP is a shared responsibility. All of the decisions reflected in this document are recommended to advise Quest Diagnostics on how to implement Google's cloud technology in a secure manner, as there are also insecure options and approaches that could be implemented.

Quest Diagnostics is responsible for the following aspects of their applications' security. It should be noted that the following list is not intended to be exhaustive and that care must be taken in each case when deploying and configuring resources in GCP.

More information on best practices for implementing detective controls that use platform telemetry to detect misconfigurations, vulnerabilities, and potentially malicious activity in the cloud environment, can be found in Google Cloud Security Foundations Guide - Section 10.

8.1.1 Authentication

If all user authentication factors require the manual entry of information (e.g. passwords and tokens), their accounts can be phished. Quest can implement cryptographic hardware factors such as Yubikeys and machine certificates to prevent successful phishing and user impersonation.

Service accounts require cryptographic keys, which are files that contain unique strings to be provided to receive authorization tokens. These tokens grant access to GCP resources via APIs. Quest Diagnostics is responsible for ensuring that the cryptographic keys and authorization tokens are stored in locations which can only be accessed by authorized people and applications and rotated at regular intervals.

8.1.2 Authorization

All access to resources configured within a project are denied by default. Access permissions are expressly granted to a user by virtue of being assigned an IAM role. Quest Diagnostics are responsible for ensuring that roles are assigned accurately at all times.

When Cloud Identity groups are used for IAM roles assignments, Quest Diagnostics is responsible for ensuring that the membership of these groups is accurate at all times.

There are managed services within GCP that permit granting access to data without establishing IAM roles. Quest Diagnostics must ensure that the granted accesses are accurate at all times. These services are:

- Google Compute Engine operating system user accounts.
- Google Cloud Storage bucket and object ACLs
- BigQuery shared datasets

8.1.3 Network

Google Compute Engine resources can be provisioned with external IP addresses. Quest Diagnostics is responsible for ensuring that only the systems which need to be addressable from outside of the GCP network receive external IP addresses.

Network tags are used by firewall rules to allow or deny access to Google Compute Engine resources from other systems inside or outside of the GCP network. Quest Diagnostics must ensure that all GCE instances receive only the network tags that they require.

Firewall rules allow or deny access between any system configured in Google Compute Engine or external addresses. Quest Diagnostics is responsible for ensuring that firewall rules are configured to grant network access only to systems, protocols, and ports that are required.

Google provides protections against Denial of Service attacks by leveraging a Load Balancer for externally-facing ports. Quest Diagnostics is responsible for deploying load balancers in front of all externally facing ports to protect their applications from DoS attacks.

8.1.4 Operating system

Quest Diagnostics is responsible for ensuring that the operating systems of their Google Compute Engine resources receive timely patches.

Quest Diagnostics is responsible for ensuring that external startup scripts for Google Compute Engine resources (either by metadata or by reference to a file in a Google Cloud Storage bucket) are only writable by the authorized people.

Operating systems can be configured to grant access to users with accounts outside of Cloud Identity. Quest Diagnostics is responsible for all aspects of ensuring authentication and authorization controls for these users. Quest Diagnostics may also perform intrusion testing and run intrusion detection software in their environments.

8.1.5 Application

All application-level security concerns such as cross-site scripting, flash injection, the use of mixed content, and the use of insecure JavaScript libraries is Quest Diagnostics' responsibility. Quest Diagnostics should be aware of each of the common concerns presented in the Open Web Application Security Project (OWASP).

Quest Diagnostics is permitted to perform their own penetration testing of applications running on their GCP infrastructure so long as this testing has no impact on other Google customers.

8.1.6 Data

Google encrypts all data communication channels that it uses to transmit data between services. Quest Diagnostics is responsible for ensuring that the communication of its data within its own applications is communicated over an encrypted channel.

Google encrypts all data on storage devices to prevent anyone with physical access to physical devices from being able to inspect the data contained on those devices. Quest Diagnostics can provide their own encryption keys for the encryption of Google Compute Engine Persistent Disks and Google Cloud Storage buckets.

Data stored within databases are all encrypted at the storage level, however additional encryption is advisable at the application level to prevent Quest Diagnostics users from accessing content and limiting spillage in the event of intrusion.

Quest Diagnostics may load data which may include PII and PCI into BigQuery for analysis. Quest Diagnostics is responsible for being aware of and abiding by any regulations regarding the use and storage of this data and are responsible for developing their own aggregation capabilities.

8.1.7 Additional Security Considerations:

The following are clarifications on design decisions made regarding service account management best practices, rotation, and DLP details for credentials:

Service Accounts

Each service created and managed within GCP will be granted only the minimum necessary permissions, adhering to the principle of least privilege outlined in the "Service Account" section above.

Service account creation and management for GCP resources will be handled through the Google Cloud IAM console for centralized control and auditability.

Key Rotation

To mitigate the risks associated with compromised keys, a robust key rotation strategy will be implemented. Quest will enforce key rotation by leveraging the organization policy `constraints/iam.serviceAccountKeyRotationPeriod`. This policy will be configured to mandate a maximum key rotation period (e.g., 30 days, 90 days) for all service accounts within the organization.

Quest has expressed interest in setting an organizational wide limit of 90 days as a catch all forced rotation TTL to ensure the rotation of service account keys.

Once a key is rotated within GCP, the responsibility for updating the key lies with the users and applications utilizing that service account. This decentralized process ensures each workload maintains uninterrupted access. While GCP provides centralized key management features, individual workloads are responsible for fetching and utilizing the latest key versions. This approach promotes agility and reduces dependencies.

Cloud Asset Inventory will be used to gain visibility into service account key usage across the organization. This service provides a centralized view of key metadata, including creation time, last rotation time, and associated resources.

Protecting GCS Buckets

GCS buckets containing sensitive data will be secured using a combination of encryption, access control, data loss prevention (DLP), and network security measures.

- **Encryption:** All GCS buckets will have Google Cloud's default server-side encryption enabled, ensuring data is encrypted at rest. For enhanced security, customer-managed encryption keys (CMEK) with Cloud KMS may be employed for granular control over encryption.

- **Access Control:** Fine-grained access control will be implemented using Google Cloud IAM, restricting access to authorized users and service accounts based on the principle of least privilege. Access policies will be regularly reviewed and audited to ensure they remain appropriate.
- **Data Loss Prevention (DLP):** Google Cloud DLP will be integrated to proactively scan GCS buckets for sensitive data. Specifically, DLP will be configured with detectors from the "Credentials and Secrets" section to identify potential exposures of secret types (e.g., API keys, passwords, certificates). DLP rules will be established to automatically remediate identified exposures, such as quarantining files or redacting sensitive information.

8.1.8 VPC Service Controls

In this project, the decision has been made to establish basic Virtual Private Cloud (VPC) perimeters initially and defer the implementation of VPC Service Controls. This phased approach is chosen for the following reasons:

- **Careful Planning and Thorough Understanding:** VPC SC requires a deep understanding of the organization's data flow and service dependencies. Implementing VPC SC without a comprehensive view of how services interact can lead to unintended consequences, such as service disruptions or unintentional data exposure. By initially setting up basic VPC perimeters, the customer gains time to thoroughly map their application landscape and identify precise service communication requirements.

- **Methodical Execution:** VPC SC policies are powerful security controls that can restrict data access and movement. Incorrect implementation can severely impact business operations. A phased approach allows for methodical policy creation, testing, and refinement, minimizing the risk of operational disruptions.

- **Flexibility and Adaptability:** Starting with basic VPC perimeters allows the customer to adapt to evolving requirements. As the customer gains more experience with their cloud environment and refines their security needs, they can iteratively implement VPC SC policies that align with their specific use cases.

Security Benefits

VPC Service Controls provide a robust security layer by establishing a security perimeter around Google Cloud resources. Key benefits include:

- **Data Exfiltration Prevention:** VPC SC helps prevent data exfiltration by restricting data movement from within the perimeter to unauthorized external destinations.
- **Containment of Compromised Accounts:** Even if an attacker gains access to an account within the perimeter, VPC SC limits their ability to move sensitive data outside.
- **Granular Control:** VPC SC offers granular control over data access and service interactions. This allows organizations to define precise security policies tailored to their specific needs.

Setup Tips

- **Start with a Well-Defined Perimeter:** Clearly define the resources that need to be protected within the VPC SC perimeter.
- **Inventory Services and Dependencies:** Thoroughly inventory all services operating within the perimeter and map their interdependencies.
- **Implement Policies Incrementally:** Start with broader policies and gradually refine them to achieve a granular security posture.
- **Test Thoroughly:** Rigorously test all VPC SC policies in a non-production environment before applying them to production.
- **Monitor and Adapt:** Continuously monitor the effectiveness of VPC SC policies and adapt them to evolving security threats and business requirements.

VPC Service Control Implementation in Terraform

Purpose

This Terraform configuration implements a VPC Service Control (VPC SC) perimeter within Quest to enhance data security and mitigate exfiltration risks. VPC Service Controls enable the creation of security boundaries around sensitive Google Cloud resources, restricting how they can communicate with other services and the internet.

Modules

vpc-sc

Source: https://github.com/terraform-google-modules/terraform-google-vpc-service-controls

This module from Google Cloud Foundation Fabric facilitates the provisioning and configuration of VPC SC perimeters.

Perimeter: quest_vpcsc_perimeter

Restricted Services: This perimeter restricts access to the API's supplied via the vpc_sc_restricted_services variable.

Important Variables

Variable Name	Type	Description
vpc_sc_restricted_services	List of strings	Services restricted within the VPC SC perimeter
vpc_sc_vpc_accessible_services	List of strings	VPC Accessible Services configuration (restricted or allowed)
vpc_sc_access_levels	Map	Definitions of access levels within the perimeter

Ingress / Egress

Ingress

VPC Service Control ingress policies define what traffic is allowed inbound into a VPC SC perimeter. They consist of the following key elements:

- **from:** Specifies the source of the traffic:
  - **identity_type:** Can be:
    - "ANY_IDENTITY" (Any caller)
    - "ANY_USER_ACCOUNT" (Any authenticated user)
    - "ANY_SERVICE_ACCOUNT" (Any service account)
  - **identities:** A list of specific service account emails, used when identity_type is "ANY_SERVICE_ACCOUNT".
  - **access_levels:** A list of Access Levels previously defined in your VPC SC configuration

- **to:** Defines the destination resources and allowed operations:
  - **operations:** A list of permitted actions. Within each item:
    - **method_selectors:** Can be used to specify specific API methods (more granular).
    - **service_name:** The name of the Google Cloud service the rule applies to (e.g., "storage.googleapis.com").

- **resources:** Specifies the target resources for the operations (e.g., specific bucket names if using storage.googleapis.com)

Example:

Allow Any Authenticated User:

```
vpc_sc_ingress_policies = {
  anyone_authenticated = {
    from = {
      identity_type = "ANY_USER_ACCOUNT"
      access_levels = ["*"] # Or restrict to specific Access Levels
    }
    to = {
      operations = [{
        service_name = "storage.googleapis.com"
      }]
      resources = ["*"]
    }
  }
}
```

Traffic from Specific Service Accounts:

```
vpc_sc_ingress_policies = {
  from_cloud_build = {
    from = {
      identities = ["serviceAccount:cloud-build@my-
project.iam.gserviceaccount.com"]
      access_levels = ["*"]
    }
    to = {
      operations = [{
        service_name = "compute.googleapis.com"
      }]
      resources = ["*"]
    }
  }
}
```

Egress

VPC Service Control egress policies control which resources within your perimeter can communicate with resources outside the perimeter (egress traffic). Here's a breakdown of their components:

- **from:** Specifies the source of the traffic within your perimeter:
  - **identity_type:** Can be:
    - "ANY_IDENTITY" (Any caller)
    - "ANY_USER_ACCOUNT" (Any authenticated user)
    - "ANY_SERVICE_ACCOUNT" (Any service account)
  - **identities:** A list of specific service account emails, used when identity_type is "ANY_SERVICE_ACCOUNT".
- **to:** Defines the allowed destination resources and actions outside the perimeter:
  - **operations:** A list of permitted operations. Within each item:
    - **method_selectors:** Can be used to specify specific API methods (more granular).
    - **service_name:** The name of the Google Cloud service the rule applies to (e.g., "storage.googleapis.com").
  - **resources:** Specifies the target resources outside the perimeter (e.g., specific bucket names within storage.googleapis.com)

Examples:

Allow Any Traffic to Google Cloud Storage:

```
vpc_sc_egress_policies = {
  allow_storage = {
    from = {
      identity_type = "ANY_IDENTITY"
    }
    to = {
      operations = [{
        service_name = "storage.googleapis.com"
        method_selectors = ["*"]
      }]
      resources = ["*"]
    }
  }
}
```

Traffic from a Service Account to Specific Resources:

```
vpc_sc_egress_policies = {
  from_logging_agent_to_pubsub = {
    from = {
      identities = ["serviceAccount:logging-agent@my-
project.iam.gserviceaccount.com"]
    }
    to = {
      operations = [{
        service_name = "pubsub.googleapis.com"
        method_selectors = ["google.pubsub.v1.Publisher.Publish"] #
Allow only Publish
      }]
      resources = ["projects/my-project/topics/log-data"]
    }
  }
}
```

Access Policy / Access Level

VPC SC Access Policies

A VPC Service Control Access Policy is a top-level policy that works alongside individual service perimeter configurations. It provides these features:

- **Parent:** Defines the scope of the policy. Can be at the organization, folder, or project level.
- **Title:** A descriptive name for the policy.

VPC SC Access Levels

Access Levels are the core building blocks for granular control within VPC SC perimeters. They work as follows:

- **Definition:** An Access Level defines a set of conditions that requests must meet in order to be allowed.
- **Conditions:** Typical conditions include:
  - **IP Subnetworks:** Source IP addresses/ranges.
  - **Device Attributes:** Restrict based on device properties (under development, not widely available yet).
  - **Region Codes:** Geographic restrictions.
- **Usage:** Access Levels are referenced in your perimeter configuration and its ingress/egress policies.

- **vpcsc_access_policy_name**
  - This is a simple string variable. The value "SNI_Gateway_Quest" provides a descriptive name for your policy.
- **vpc_sc_access_levels**
  - You're defining an Access Level named from_sni.
  - **Conditions:** This Access Level only allows requests originating from the list of IP subnetworks you've provided in the ip_subnetworks list.

Examples: Restrict by Region

```
vpc_sc_access_levels = {
  allow_asia = {
    conditions = [
      { region_code = "asia-east1" }
    ]
  }
}
```

VPC SC Troubleshooting

VPC Service Controls log entries often contain data about denied requests to protected services, such as the resources being requested and the reason why access was denied. However, these details aren't always easily apparent and can require users to spend considerable time understanding the logs. The VPC Service Controls Troubleshooter is a tool that enables security administrators to better understand and troubleshoot a denial that is caused by VPC Service Controls.

8.1.9 Managed SSL Certificates in Google Cloud Platform

Quest Diagnostics has expressed interest in SSL Certificate management options in GCP. Google Cloud Platform offers managed SSL certificates as a convenient and secure way to encrypt communication between your applications and users. With managed SSL certificates, Google handles the entire certificate lifecycle, including procurement, deployment, and renewal. This eliminates the complexities of manual certificate management and ensures continuous protection for your applications.

Benefits of using managed SSL certificates:

- **Automatic renewal:** Google automatically renews your certificates before they expire, eliminating the risk of downtime due to expired certificates.
- **Simplified management:** Google handles all the complexities of certificate management, freeing you from manual tasks and reducing operational overhead.
- **Enhanced security:** Google ensures that your certificates are issued from trusted Certificate Authorities (CAs) and comply with industry best practices.
- **Cost-effective:** Managed SSL certificates are often included as part of Google Cloud Platform services, reducing the need to purchase and manage certificates separately.

How to use managed SSL certificates:

Managed SSL certificates can be used with various Google Cloud Platform services, including:

- **Load balancing:** Google Cloud Load Balancing can automatically provision and manage SSL certificates for your load balancers, ensuring secure communication between your applications and clients.
- **Kubernetes Engine:** Google Kubernetes Engine (GKE) integrates with managed SSL certificates, allowing you to secure your Kubernetes applications with minimal effort.
- **Cloud Run:** Cloud Run automatically provisions and manages SSL certificates for your deployed services, ensuring secure HTTPS connections.

To create and use managed SSL certificates, you typically need to:

1. **Create a managed certificate resource:** This involves specifying the domain names that you want to secure and any associated configurations.
2. **Associate the certificate with your service:** This links the managed certificate to your load balancer, Kubernetes Ingress, or Cloud Run service.
3. **Configure your application:** Ensure that your application is configured to use HTTPS and that the necessary ports are open.

8.1.10 Data Loss Prevention

Data Loss Prevention (DLP) is a set of tools and processes used to ensure that sensitive data is not lost, misused, or accessed by unauthorized users. DLP software classifies regulated, confidential and business critical data and identifies violations of policies defined by organizations or within a predefined policy pack, typically driven by regulatory compliance such as HIPAA, PCI-DSS, or GDPR. It can be implemented in many ways, some are:

- Data Encryption
- Data Masking
- Data loss prevention by controlling the data access and usage
- Defining data governance policies

Recommendation: Google Cloud team suggest the use of Cloud Data Loss Prevention (now part of Sensitive Data Protection services) to identify/mask/tokenize the confidential data. However, DLP implementation strategy is out of scope for the Data Foundation initiative.

8.1.11 Secret Manager

Secret Manager allows you to store, manage, and access secrets as binary blobs or text strings. With the appropriate permissions, you can view the contents of the secret.

Secret Manager works well for storing configuration information such as database passwords, API keys, or TLS certificates needed by an application at runtime. A key management system, such as Cloud KMS, allows you to manage cryptographic keys and to use them to encrypt or decrypt data. However, you cannot view, extract, or export the key material itself.

Similarly, you can use a key management system to encrypt sensitive data before transmitting it or storing it. You can then decrypt the sensitive data before using it. Using a key management system to protect a secret in this way is more complex and less efficient than using Secret Manager. Cloud KMS is designed to handle large encryption workloads, such as encrypting rows in a database or encrypting binary data such as images and files. You can also use Cloud KMS to perform other cryptographic operations such as signing and verification.

There are a few concepts that are worth understanding:

Secret: A secret is a project-global object that contains a collection of metadata and secret versions. The metadata can include replication locations, labels, and permissions. The secret versions store the actual secret data, such as an API key or credential.
Version: A secret version stores the actual secret data, such as API keys, passwords, or certificates. You can address individual versions of a secret. You cannot modify a version, but you can delete it.
Rotation: You rotate a secret by adding a new secret version to the secret. Any version of a given secret can be accessed, as long as that version is enabled. To prevent a secret version from being used, you can disable that version. It is impossible to schedule a secret for automatic rotation.

Recommendation

The recommendation is that passwords and access keys be secured in the Secret Manager, if the used service or tool supports this feature. However, the implementation of the Secret Manager is out of scope for the Data Foundation Set up.

8.1.12 Cloud Armor

Cloud Armor helps protect Google Cloud deployments from multiple types of threats, including distributed denial-of-service (DDoS) attacks and application attacks like cross-site scripting (XSS) and SQL injection (SQLi). Cloud Armor features some automatic protections and some that Quest needs to configure manually. This section provides a high-level overview of these features.

Cloud Armor provides always-on DDoS protection against network or protocol-based volumetric DDoS attacks. This protection is for applications or services behind load balancers. It is able to detect and mitigate network attacks in order to allow only well-formed requests through the load balancing proxies. Most of the Load Balancers included are Global external HTTP(S) load balancers.

Visibility and monitoring
Google Cloud Armor exports monitoring data from security policies to Cloud Monitoring. You can use monitoring metrics to check whether your policies are working as intended or to troubleshoot problems. For example, you can view the traffic that was blocked or allowed for each backend service. You can monitor the metrics of a single security policy (which can be applied to multiple backend services) or a single backend service.

Google Cloud Armor is integrated automatically with Security Command Center and exports two findings to the Security Command Center dashboard: Allowed Traffic Spike and Increasing Deny Ratio.

Security policies

Cloud Armor security policies protect your application by providing Layer 7 filtering and by scrubbing incoming requests for common web attacks or other Layer 7 attributes to potentially block traffic before it reaches LB's backend services. Each security policy is made up of a set of rules that filter traffic based on conditions such as an incoming request's IP address, IP range, region code, or request headers.

Cloud Armor security policies enable you to allow or deny access to deployments at the Google Cloud edge, as close as possible to the source of incoming traffic. This prevents unwelcome traffic from consuming resources or entering your VPC networks.

These are the requirements for using Cloud Armor security policies:

- The load balancer must be a global external HTTP(S) load balancer, global external HTTP(S) load balancer (classic), external TCP proxy load balancer, or external SSL proxy load balancer.
- The backend service's load balancing scheme must be *EXTERNA*L, or *EXTERNAL_MANAGED* if Quest is using a global external HTTP(S) load balancer.
- The backend service's protocol must be one of HTTP, HTTPS, HTTP/2, TCP, or SSL.

Threat Intelligence

Cloud Armor Threat Intelligence enables you to secure traffic by allowing or blocking traffic to external HTTP(S) load balancers based on several categories of threat intelligence data. Threat Intelligence data is divided into the following categories:

- Tor exit nodes: Tor is open-source software which enables anonymous communication. To exclude users who hide their identity, block the IP addresses of Tor exit nodes (points at which traffic exits the Tor network).
- Known malicious IP addresses: IP addresses that need to be blocked to improve your application's security posture because attacks on web applications are known to originate there.
- Search engines: IP addresses that you can allow to enable site indexing.
- Public cloud IP address ranges: This category can be either blocked to avoid malicious automated tools from browsing web applications, or allowed if your services use other public clouds.

To use Threat Intelligence, you will define security policy rules that allow or block traffic based on some or all of these categories by using the *evaluateThreatIntelligence* match expression along with a feed name that represents one of the preceding categories.

For details about how to Configure Threat Intelligence, please refer to the following link

Rate Limit

Cloud Armor provides capabilities to help protect Quest's GCP applications against a variety of Layer 3 and Layer 7 attacks. Rate-based rules help to protect applications from a large volume of requests that flood the instances and block access for legitimate users.

Rate limiting can do the following:

- Prevent any particular client from exhausting application resources.
- Protect Quest's application instances from erratic and unpredictable spikes in the rate of client requests.

In addition, when a resource is presented with a high volume of traffic from a small number of clients, Quest can prevent other clients from being affected by large spikes of traffic from that small number of clients, enabling the resources to handle as many requests as possible.

Cloud Armor has two types of rate-based rules:

1. Throttle: Quest can enforce a maximum request limit per client or across all clients by throttling individual clients to a user-configured threshold.
2. Rate-based ban: Quest can rate limit requests that match a rule on a per-client basis and then temporarily ban those clients for a configured time if they exceed a user-configured threshold.

Quest can preview the effects of rate limiting rules in a security policy by using preview mode and examining request logs.

Throttling traffic

The *throttle* action in a rule allows you to enforce a per-client request threshold to protect backend services. This rule enforces the threshold to limit traffic from each client that satisfies the match conditions in the rule. The threshold is configured as a specified number of requests in a specified time interval.

Example: you might set the request threshold to 2,000 requests within 1,200 seconds (20minutes). If a client sends 2,500 requests within any 1,200 second period, approximately 20% of the client's traffic is throttled until the permitted request volume is at or below the configured threshold.

You would need to set these parameters to control the action:

- ***rate_limit_threshold*:** The number of requests per client allowed within a specified time interval. The minimum value is 1 and the maximum value is 10,000.
    - *interval_sec*: The number of seconds in the time interval. The value must be 60, 120, 180, 240, 300, 600, 900, 1200, 1800, 2700, or 3600 seconds.
- ***exceed_action*:** When a request exceeds the rate_limit_threshold, Cloud Armor applies the configured exceed_action. Possible values for the *exceed_action* are as follows:

    - *deny(status)*: The request is denied and the specified error code is returned (valid values are 403, 404, 429 and 502). We recommend using the *429 (Too Many Requests)* response code.
    - *redirect*: The request is either redirected for reCAPTCHA Enterprise assessment or to a different URL, based on the exceed_redirect_options parameter.
- ***exceed_redirect_options*:** When the *exceed_action* is *redirect*, use this parameter to specify the redirect action:
    - *type*: Type for the redirect action, either GOOGLE_RECAPTCHA or EXTERNAL_302.
    - *target*: URL target for the redirect action. Only applicable when the type is EXTERNAL_302.
- ***conform_action*:** This is the action performed when the number of requests is under the *rate_limit_threshold*. This is always an *allow* action.

When a client's traffic rate is under or equal to the *rate_limit_threshold*, requests follow the *conform-action*, which is always an *allow* action. The request is allowed through the security policy and permitted to reach its destination. When a client's traffic rate exceeds the specified *rate_limit_threshold*, Cloud Armor applies the *exceed_action*, which can be either *deny* or *redirect*, for requests over the limit for the rest of the threshold interval.

Threshold enforcement

The configured thresholds for throttling and rate-based bans are enforced independently in each of the Google Cloud regions where Quest's HTTP(S) backend services are deployed. For example, if one of your services is deployed in two regions, each of the two regions rate limits each key to the configured threshold, so Quest's backend service might experience cross-region aggregated traffic volumes that are twice the configured threshold. If the configured threshold is set to 5,000 requests, the backend service might receive 5,000 requests from one region and 5,000 requests from the second region. However, for the key type IP address, it is reasonable to assume that traffic from the same client IP address is directed to the region that is closest to the region where Quest's backends are deployed. In this case, rate limiting can be considered to be enforced at a backend service level, regardless of the regions it is deployed in.

Best practices

Cloud Armor is deployed with either the global external HTTP(S) load balancer, the global external HTTP(S) load balancer (classic), the external TCP proxy load balancer, or the external SSL proxy load balancer. When Quest deploys Cloud Armor, a security policy can be attached to the load balancer backend service that Quest wants to protect. A security policy consists of a collection of pre-configured and custom rules that Quest determines.

Security policy and rule creation: These are some of the best practices and recommendations for new security policies and rules.

- Provide rule descriptions: Use rule descriptions to provide additional context about why each rule was created and the intended function of the rule. The

description field is limited to 64 characters, so references to configuration management databases or other repositories are the most efficient way to capture context.

- Consider priority spacing: When you initially configure rules, leave an interval of at least 10 between each rule priority value. For example, the first two rules in a security policy could have priorities 20 and 30. This lets Quest insert more rules when they need them. In addition, we recommend that Quest group similar rules into blocks, leaving larger intervals between groups.
- Use preview mode: Security policy rules, including Open Web Application Security Project (OWASP) signatures, can have unpredictable effects on Quest's application. Use preview mode, to evaluate whether the introduction of a rule will have a negative impact on production traffic.
- Enable Cloud Armor Adaptive Protection: this is for additional protection of your applications. Adaptive Protection monitors traffic and (as necessary) recommends new rules for Quest's security policies. In addition, we recommend that you put an alerting policy in place to ensure that the right people are alerted about potential attacks. Adaptive Protection is best suited for volumetric protection. Attacks that are not volumetric might not trigger Adaptive Protection.
- Enable JSON parsing: If Quest's application sends JSON content in the body of POST requests, ensure that JSON parsing it's enabled. If Quest does not enable JSON parsing, Cloud Armor does not parse the JSON content of POST bodies for preconfigured WAF rules, and the results can be noisy and generate false positives. For additional information, see JSON parsing.

Test the logic

A rule is triggered when its match condition evaluates to true; for example, the match condition origin.region_code == 'AU' evaluates to true if the region code of the request is AU. If a higher priority rule evaluates to true, then the action in a lower priority rule is ignored. In the following example, if Quest wants to create a security policy to block users from the AU region, except for traffic within the IP address range 10.10.10.0/24. Consider the following security policy with two rules:

```
Rule1
expr: inIPRange(origin.ip, '10.10.10.0/24')
action: allow
priority: 1
Rule2
expr: origin.region_code == 'AU'
action: deny(403)
priority: 2
```

In this example, Rule1 allows traffic that originates from the IP address range 10.10.10.0/24. Because Rule1 is the higher-priority rule, such traffic is allowed before it is evaluated against Rule2, meaning that Cloud Armor does not evaluate it against Rule2 (or any other remaining rules). When Quest creates Cloud Armor policies, test the logic of the rules to ensure it achieves the intended behavior. To do so, we recommend that Quest generates synthetic traffic to learn which rules are blocking traffic, and verify that their results are consistent with their rule design decisions. If you are unsure of how a request might flow through the system, use preview mode to see which rule matches the request.

Identify the source IP addresses of the scanners:
Quest's security scanners can be located inside or outside of Google. If Quest wants an outside and unfiltered assessment of their application, Quest can explicitly allow traffic based on IP address (or other token) prior to evaluating it against any other rules.

Group and sort rules in the security policy:
Quest's applications might serve different subsets of Quest's customers. In the following example, the need is to deny traffic from certain geographical areas or IP ranges, and therefore configure the first rule in the policy to deny such traffic. Additionally, the need is to explicitly allow some traffic into the application without the security policy processing it. For this example, we recommend the following structure of rule priority, from greatest-priority to least-priority:

1. Explicit deny rules (ASN, region, IP ranges)
2. Trusted explicit allow rules (scanners, trusted systems - use with extreme caution)
3. Security rules (OWASP, custom rules)
4. Explicit allow rules (ASN, presence of header value, IP range)
5. Default deny rules

Use bot management where appropriate
Cloud Armor integrates with Google's reCAPTCHA Enterprise. If Quest is using reCAPTCHA Enterprise, move the token assessment process to Cloud Armor. This reduces origin load and puts security controls closer to the end user than Quest's backends. For more information, see the bot management overview.

Set rate limiting thresholds
Rate limiting is a flexible and valuable capability to prevent abuse and mitigate high volume threats like credential stuffing or L7 DDoS attacks. When Quest deploys rate limiting for the first time, it is important to choose a threshold that makes sense for Quest's application. We recommend starting with enforcement in preview mode. As Quest analyzes and understands the traffic profile, Quest can adjust the rate limiting parameters. In addition, it is important to consider the priority that Quest assigns to the rate limiting rule. Traffic might be explicitly allowed or denied by a higher priority rule before it is evaluated against the rate limiting rule.

Rule tuning

Web applications might allow requests that appear to be attacks, and they might allow, or even require, that users send requests that match the signatures in preconfigured WAF rules. It is critical that Quest validates the Cloud Armor rules against their application and address any findings that might not be relevant for the application before promoting the rule by disabling preview mode on a production application. The following sections contain best practices and recommendations for tuning the preconfigured WAF rules.

Choose the preconfigured WAF rule sensitivity level:

When any of the preconfigured WAF rules are implemented, you can choose an appropriate sensitivity level based on their security requirements and timelines. We recommend that Quest begins with a sensitivity level of 1 for most applications that must meet their organization's security requirements. Rules configured for sensitivity 1 use high fidelity signatures and reduce potential noise from the rule. Signatures associated with higher sensitivities might detect and prevent a larger set of exploit attempts, at the expense of potential noise for some protected applications. However, workloads subject to more strict security requirements might prefer the highest sensitivity level. For these use-cases, there might be a great amount of noise or irrelevant findings, which Quest must address using tuning before the security policy goes into production.

Enable verbose logging:

If Quest requires additional information about which request attributes and payloads are triggering a particular WAF rule, enable verbose logging. Verbose logging provides details from requests that trigger particular rules, including a snippet of the offending portion of the request, which is helpful for troubleshooting and tuning Cloud Armor. Because verbose logging can cause end-user request content to be logged in Cloud Logging, there is a chance that Quest accumulates end-user PII in the logs. As a result, we do not recommend running production workloads with verbose logging enabled for long periods of time.

Use stable or canary rules:

There are two types of Cloud Armor preconfigured WAF rules: stable and canary. When new rules are added to the current ModSecurity Core Rule Set (CRS), we publish them to the canary rule builds before automatically publishing them into the stable rule builds. We recommend that Quest deploys the canary rules in a testing environment so that Quest can see the effects of any changes and additions in the environment. Quest can check rule names on the Tuning Cloud Armor WAF rules page to verify whether the canary build is in sync with the stable build.

Google Cloud Armor provides a comprehensive list of preconfigured WAF rules based on the OWASP ModSecurity Core Rule Set (CRS)

1. SQL injection attacks
2. Cross-site scripting attacks
3. Local file inclusion attacks
4. Remote file inclusion attacks
5. Remote code execution attacks
6. Method enforcement attacks
7. Scanner detection attacks
8. Protocol attacks
9. PHP injection attacks
10. Session fixation attacks
11. Java attacks
12. NodeJS attacks

Automation Using Cloud Armor Terraform Module

Terraform Module format

```
module security_policy {
  source = "GoogleCloudPlatform/cloud-armor/google"

  project_id                           = "my-project-id"
  name                                 = "my-test-ca-policy"
  description                          = "Test security policy with preconfigured
rules"
  default_rule_action                  = "deny(403)"
  pre_configured_rules                 = {} # These are based on OWASP Top 10 Pre-
Configured WAF rules
  security_rules                       = {} # Allow or deny traffic from set of IP
Addresses
  custom_rules                         = {} # Custom rules using Common Expression
Language
  threat_intelligence_rules            = {} # Rules based on Threat Intelligence
}
```

Example Preconfigured Rule

Google Cloud Armor preconfigured WAF rules are rules with dozens of signatures that are compiled from open source industry standards. Each signature corresponds to an attack detection rule in the ruleset. Google offers these rules as-is. The rules allow Google Cloud Armor to evaluate dozens of distinct traffic signatures by referring to conveniently named rules rather than requiring you to define each signature manually.

```
pre_configured_rules  = {
```

```
  "sqli_sensitivity_level_4" = {
    action            = "deny(502)"
    priority          = 1
    description       = "SQL Sensitivity Level 4"
    preview           = false
    redirect_type     = null
    target_rule_set   = "sqli-v33-stable"
    sensitivity_level = 4
    rate_limit_options= {}
    }

}
```

Google Cloud Armor preconfigured WAF rules can be tuned to best suit your needs. For more information about how to tune the rules, see Tune Google Cloud Armor preconfigured WAF rules.

8.2 HIPAA Compliance

It is important to note that there is no certification recognized by the US HHS for HIPAA compliance and that complying with HIPAA is a shared responsibility between Quest Diagnostics and Google. Specifically, HIPAA demands compliance with the Security Rule, the Privacy Rule, and the Breach Notification Rule. Google Cloud supports HIPAA compliance (within the scope of a Business Associate Agreement) but ultimately Quest Diagnostics is responsible for evaluating their own HIPAA compliance.

8.2.1 Decide how to meet compliance requirements for encryption at rest

Google Cloud automatically encrypts all your content stored at rest, using one or more encryption mechanisms. Depending on your compliance requirements, you might have an obligation to manage the encryption keys yourself.

The following sections describe the options for encryption at rest with the preferred, CMEK Solution voiced by Quest:

Manage encryption keys using Cloud KMS

In addition to default encryption at rest, you might require more control over the keys used to encrypt data at rest within a Google Cloud project. Cloud Key Management Service (Cloud KMS) offers the ability to protect your data using customer-managed encryption keys (CMEK).

For example, in the financial services industry, you might have a requirement to report to your external auditors how you manage your own encryption keys for sensitive data.

For additional layers of control, you can configure hardware security modules (HSM) or external key management (EKM) with CMEK. Customer-supplied encryption keys (CSEK) are not recommended; scenarios that historically were addressed by CSEK are now better addressed by Cloud External Key Manager (Cloud EKM) because Cloud EKM has support for more services and higher availability.

This option shifts some responsibility to application developers to follow the key management that your security team mandates. The security team can enforce the requirement by blocking the creation of non-compliant resources with CMEK organization policies.

Use this option when one or more of the following is true:

- You have a requirement to manage the lifecycle of your own keys.
- You have a requirement to generate cryptographic key material with a FIPS 140-2 Level 3 certified HSM.
- You have a requirement to store cryptographic key material outside of Google Cloud using Cloud EKM.

Avoid this option when the following is true:

- You don't have particular requirements for how to encrypt data or how encryption keys are managed.
- You prefer a managed service over the cost and operational overhead of managing your own encryption keys.

For more information, see the following:

- Manage encryption keys with Cloud Key Management Service in the enterprise foundations blueprint
- Customer-managed encryption keys (CMEK)
- Cloud HSM
- Cloud External Key Manager
- CMEK organization policies

8.2.2 Decide how to meet compliance requirements for encryption in transit

Google Cloud has several security measures to help ensure the authenticity, integrity, and privacy of data in transit. Depending on your security and compliance requirements, you might also configure application layer encryption.

The following sections describe the options for encryption in transit according to Quest's decision to ensure mandatory Layer 7 Encryption in transit:

In addition to default encryption in transit, you can configure Layer 7 encryption for application traffic. Google Cloud provides managed services to support applications that need application-layer encryption in transit, including managed certificates, Cloud Service Mesh, and Cloud Service Mesh.

For example, a developer is building a new application that accepts ingress traffic from the internet. They use an external Application Load Balancer with Google-managed SSL certificates to run and scale services behind a single IP address.

Application layer encryption is not a control that you can enforce centrally in the landing zone. Instead, this option shifts some responsibility to application developers to configure encryption in transit.

Use this option when applications require HTTPS or SSL traffic between components.

Consider allowing a limited exception to this option when you are migrating compute workloads to the cloud that did not previously require encryption in transit for internal traffic when the applications were on-premises. During a large-scale migration, forcing legacy applications to modernize before migration might cause an unacceptable delay to the program.

For more information, see the following:

- Using Google-managed SSL certificates
- Using self-managed SSL certificates
- Cloud Service Mesh
- Cloud Service Mesh service security

8.3 Data Management

8.3.1 Cloud KMS

If Quest Diagnostics needs to encrypt data at the application level or manage their own encryption keys for compliance or regulatory reasons they should look at Cloud Key Management Service (KMS). Cloud KMS is a global cloud-hosted key management service that lets you manage encryption for your cloud services the same way you do on-premises.

Best Practices

1. Key rotation - Regular rotation of the encryption key is encouraged. This limits the amount of data protected by a single key. Automatic rotation can be configured on a user defined schedule by using *gcloud* or the GCP Console.

2. Separation of duties - Cloud KMS should be run in its own project without an owner at the project-level and instead being managed by an Org Admin. The Org Admin is not able to manage or use keys, but they are able to set IAM policies to restrict who has permissions for key management and usage. Additionally, the ability to manage Cloud

3. KMS should have role separation from the ability to perform encryption and decryption operations. Any user with management access should not be able to decrypt data.

4. Additional authenticated data (AAD) - We recommend AAD as an additional integrity check as it can help protect your data from a confused deputy attack. Additional authenticated data is a string that you pass to Cloud KMS as part of an encrypt or decrypt API call. Cloud KMS cannot decrypt ciphertext unless the same AAD value is used for both encryption and decryption. By default an empty string is used for the AAD value.

9. Infrastructure Automation

When operating in a cloud computing setting where all IT assets are defined as code, it is best practice to also define infrastructure (among other assets) in code. There are a number of advantages to adopting an IaC strategy in your cloud operating model, some of which are:

- Governance of deployed infrastructure: who gets to deploy what and when.
- The enforcement of security controls, compliance and audit capabilities is facilitated.
- Repeatable, predictable, and rapid deployment process across multiple environments (e.g. Development, QA, Staging, Production, etc.)

- Declare and capture the state of your infrastructure in source files that anyone can read rather than depending on what's in a sysadmin's head.
- Roll back if needed. By using version control, the entire history of your infrastructure is now captured in the commit log.
- Validate each infrastructure change through code reviews and automated tests.

Recommendation:

The Cloud Foundation Fabric provides end-to-end blueprints and a suite of Terraform modules for Google Cloud, which support different use cases:

- Organization-wide landing zone blueprint used to bootstrap real-world cloud foundations
- Reference blueprints used to deep dive into network patterns or product features
- A comprehensive source of lean modules that lend themselves well to changes

9.1 Deployment Strategy

HCLTech will deploy a total of 13 GCP services using Infrastructure as Code (IaC). Below is the list of tasks to be performed to achieve the first milestone readiness for the Infrastructure Landing Zone.

Task	Sub-Task	Deployment Approach	Status	Dependency

Landing Zone Deployment	• Resource Hierarchy (Folder and Project Structure) • IAM Roles configuration • API services • Organization Policies • Shared VPCs (External, Prod, etc.) • Subnets • Cloud Router • FortiGate Firewall • Load Balancers (Internal/External) • Firewall Rules • Key Management • Security Command Center enablement	GitHub Actions	WIP	FortiGate license required by HCL team
VPN Tunnel Configuration	SSH-based VPN setup between US-West to Dallas and US-East to Philadelphia	Manual	WIP	Joint call with Quest Network Team and HCL
Creation of Organization	Establish centralized management	Manual	WIP	One-time activity, requires joint call with Quest IAM Team and HCL
Creation of Billing Account	Setting up billing for GCP	Manual	WIP	Requires coordination with Quest IAM Team and HCL

Cloud Identity Configuration	Azure AD Integration (SSO, SAML, user and group synchronization)	Manual	WIP	Requires a joint call with Quest IAM Team and HCL
IaC Deployment Prerequisites	● Create Bootstrap Folder ● Create Bootstrap Project ● Create Cloud Storage Bucket for State File ● Create Service Account for GitHub Authentication ● Workload Identity Federation configuration ● GitHub Repo & Runner setup	Manual	WIP	- Requires Quest IAM & DevOps Team collaboration - Self-hosted runner creation and connectivity to be allowed
Services to be Deployed via IaC	13 GCP services will be deployed through Infrastructure as Code (IaC).	IaC	Planned	None specified

9.2 State File Management

Proper state file management is crucial for ensuring security, reliability, and traceability in Test-Driven Development (TDD). Below are recommended best practices for managing state files effectively:

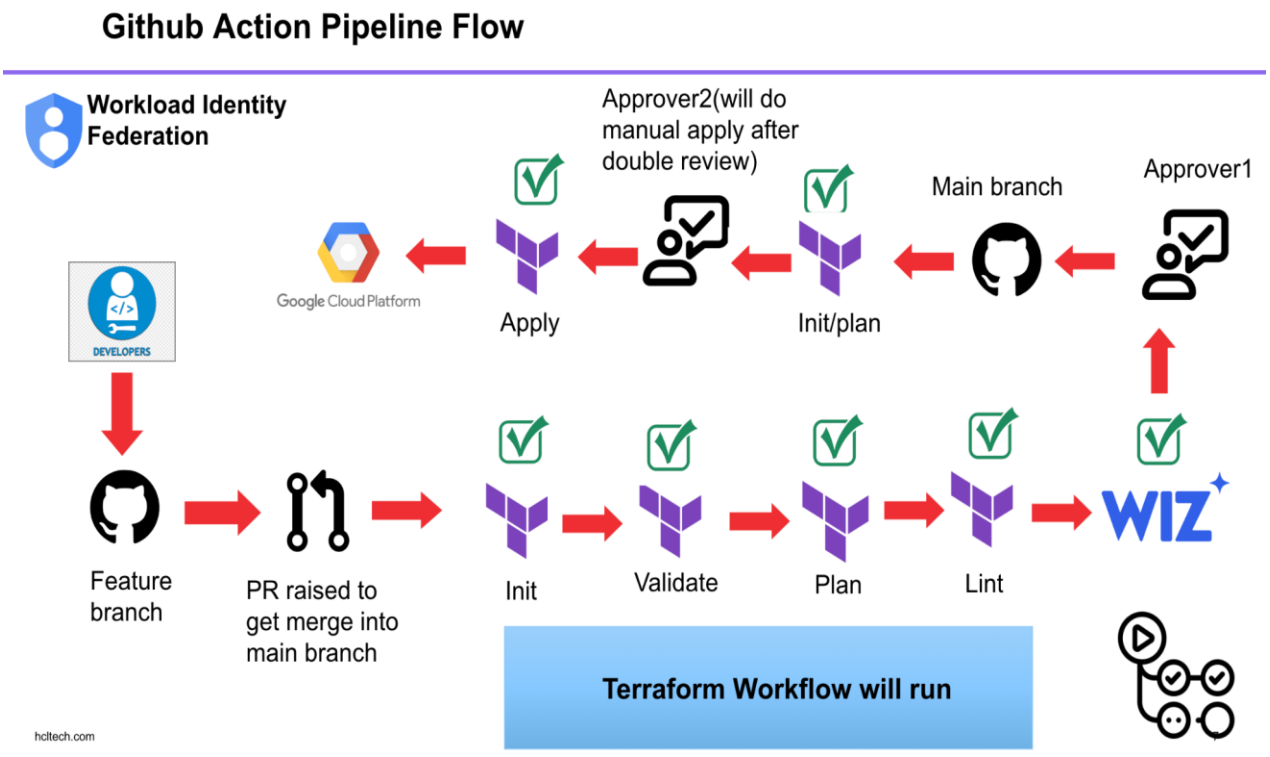
- No Public Access: Deny access to allUsers and allAuthenticatedUsers to prevent unauthorized access to state files.
- Fine-Grained Access Control: Implement object-level ACLs to control access for individual files. This ensures that only authorized users or systems involved in the TDD process can access specific state files.
- Object Versioning: Enable versioning for state files to keep a history of changes.

- Maintain a maximum of 5 versions per object to limit storage usage.
- Automatically expire noncurrent versions after 30 days to ensure that only relevant versions are retained.

- Data Encryption: Use Customer-Managed Encryption Keys (CMEK) to encrypt state files, ensuring compliance with security policies and protecting sensitive TDD-related data.
- Restrict Access: Limit access to state files only to necessary users or service accounts, reducing the risk of unauthorized changes or accidental deletions.
- Audit Logs: Enable audit logging to track all access and changes made to state files. This provides traceability and supports debugging during the TDD lifecycle.

9.3 Infrastructure CI/CD Pipeline

The process for setting up a GCP landing zone involves several steps. Initially, the setup should be performed within GCP using Compute Instances or Cloud Shell. Terraform code and configurations must be utilized to provision the landing zone resources. Once the initial setup is complete, the Terraform code and modules should be integrated into GitHub Actions pipelines, along with Wiz scanning, to streamline ongoing management. GitHub Actions runners, configured with private IPs, need internet connectivity (egress only) to communicate with github.com. After the runners are in place, all Terraform and Infrastructure-as-Code (IaC) pipelines must be executed exclusively through GitHub Actions going forward.



GitHub Actions pipeline flow for managing infrastructure as code (IaC) using Terraform in conjunction with Google Cloud Platform (GCP) and integrating security scans via Wiz. Here's a step-by-step breakdown:

1. **Feature Branch Development**: Developers make changes in a **feature branch** in GitHub.
2. **Pull Request (PR) Creation**: Developers create a pull request (PR) to merge their changes into the main branch.
3. **Terraform Workflow**: When the PR is raised, the GitHub Actions pipeline automatically runs a Terraform workflow:
    a. **Init**: Initializes Terraform working directory.
    b. **Validate**: Ensures that the Terraform configuration is syntactically and semantically valid.
    c. **Lint**: Checks for style or best practice issues in the Terraform code.
    d. **Plan**: Creates an execution plan showing what changes will be made
4. **Security Scan**:The plan is scanned by **Wiz** to identify potential security vulnerabilities.
5. **Approval Process** : After a successful scan, the plan goes for review:
    a. **Approver1** reviews the changes.
    b. If approved, **Approver2** does a final review and manually applies the changes (Terraform **Apply**).
    c.
6. **Deployment** : Upon final approval, Terraform applies the changes to the infrastructure in GCP.
7. **Merge to Main Branch** : Once the infrastructure changes are successfully applied, the feature branch is merged into the main branch.

9.4 Branch Protection Rules:

- **Require Pull Request Review Before Merging**: Changes to the branch must go through a pull request (PR) process and be reviewed by designated reviewers before they can be merged. This ensures code quality and catches potential issues early.
- **Require Status Checks to Pass Before Merging**: Automated checks (e.g., CI/CD pipelines, tests, security scans) must pass before the PR can be merged. This guarantees that the code is stable and meets predefined criteria.

- **Prevent Direct Pushes to Protected Branch (main):** Developers are prohibited from directly pushing changes to the main branch. All changes must go through a pull request to ensure proper review and approval.
- **Dismiss Stale Pull Request Approvals When New Commits Are Pushed**: If new commits are added to a pull request after it has been approved, the previous approvals are dismissed. This ensures reviewers re-evaluate the latest changes for quality and correctness.
- **Require Review from Code Owners**: Specific individuals or teams designated as Code Owners (those responsible for certain files or directories) must approve pull requests before they can be merged. This ensures domain experts review relevant code changes.

10. Billing

Google Compute Platform charges are handled through billing accounts. Billing accounts can be handled three ways for enterprise customers: credit card charges, bank account debits, or "offline" billing paid via purchase order. At least one billing account is necessary to get started in GCP. Each project can be associated with a single billing account; a single billing account can be associated with multiple projects.

Cloud Support API: https://cloud.google.com/support/docs/reference/support-api
Quota Request API: https://cloud.google.com/docs/quotas/api-overview
Quota default limit: https://cloud.google.com/docs/quotas/overview