Comprehensive Implementation Guide: PRJ-GCP-DATA-080 - Data Catalog and Governance

This document provides a comprehensive, step-by-step guide for implementing a robust data governance and security framework on Google Cloud Platform (GCP). The solution, identified as **PRJ-GCP-DATA-080**, establishes a unified, automated, and scalable governance model leveraging GCP-native services to protect sensitive data, ensure regulatory compliance, and mitigate data exfiltration risks.

---

1. Project Overview

The **PRJ-GCP-DATA-080** project is designed to enforce a **zero-trust data security posture** across critical GCP data services, including BigQuery, Cloud Storage, and Cloud SQL. The core architecture integrates four key GCP services to create a layered defense mechanism:

- **Dataplex**: Serves as the **unified data management and governance layer**, providing a central catalog, data quality checks, and metadata management across data lakes and warehouses. It is the foundation for data classification and discovery.
- **VPC Service Controls (VPC SC)**: Creates a **security perimeter** around the specified GCP services, effectively preventing data exfiltration to unauthorized external networks or projects. This is a critical control for protecting data at the network level.
- **Cloud Data Loss Prevention (Cloud DLP)**: Automates the **discovery, classification, and redaction** of sensitive data (e.g., PII, financial data) at scale. It integrates with Dataplex to enrich the data catalog with sensitivity labels.
- **Cloud Key Management Service (Cloud KMS)**: Manages **Customer-Managed Encryption Keys (CMEK)**, ensuring that data at rest in BigQuery and Cloud Storage is encrypted with keys controlled by the organization, meeting stringent compliance requirements.

The primary objective is to move from a reactive, manual governance model to a proactive, automated, and policy-driven framework that scales with the organization’s data growth.

2. Business Context

Organizations operating in the cloud face a dual challenge: maximizing the value of their data through analytics while simultaneously managing the escalating risks associated with data breaches and regulatory non-compliance. The manual processes traditionally used for data governance are not sustainable given the exponential growth of data volumes and the complexity of modern data architectures.

The Problem and Solution Alignment

Challenge	Impact of Challenge	PRJ-GCP-DATA-080 Solution
Data Breaches & Exfiltration	Significant financial penalties, reputational damage, loss of customer trust. Average cost of a data breach is in the millions.	**VPC Service Controls** and **CMEK** establish a network perimeter and strong encryption, making data inaccessible even if a breach occurs.
Regulatory Non-Compliance	Risk of massive fines (e.g., up to 4% of global annual revenue for GDPR), operational halts, and increased audit costs.	**Cloud DLP** and **Dataplex** automate data classification and discovery, providing auditable evidence of compliance with frameworks like GDPR and HIPAA.
Inefficient Data Access	Delays in data-driven decision-making due to manual access requests and inconsistent security policies.	**Dataplex** provides a unified catalog, and **BigQuery Column-Level Security** grants fine-grained access, enabling secure and faster data consumption.

Quantified Business Value and ROI

The implementation of this framework delivers significant, quantifiable returns on investment:

1. **Risk Mitigation (Avoided Costs):** By implementing VPC Service Controls and CMEK, the project drastically reduces the probability and impact of a data breach. A conservative estimate suggests a **20-30% reduction in data security risk exposure** related to cloud data services. The cost of a single major compliance fine (e.g., a multi-million dollar GDPR penalty) is entirely avoided.
2. **Operational Efficiency (Time Savings):** Automated data classification via Cloud DLP and Dataplex eliminates the need for manual data tagging and inventory. This can result in a **50-70% reduction in time spent on data discovery and classification** for compliance reporting and audit preparation. The unified governance plane of Dataplex also simplifies data management, reducing administrative overhead.

3. **Audit and Compliance Cost Reduction:** The centralized audit logs, classification reports, and access controls provide immediate, verifiable evidence for auditors. This streamlined process can reduce the duration and cost of external audits by **up to 40%**.
4. **Enabling Secure Innovation:** By providing a secure, governed environment, data scientists and analysts can access the data they need faster and with confidence, accelerating time-to-insight and fostering data-driven innovation without compromising security.

3. GRC Mapping

The PRJ-GCP-DATA-080 framework is specifically designed to address key controls across major global Governance, Risk, and Compliance (GRC) standards. The layered security approach ensures that controls for data protection, access management, and auditability are met.

Framework	Control/Requirement	GCP Service Alignment	Alignment Description
NIST 800-53	**PR.DS-1** (Data Protection), **PR.DS-5** (Data Leakage Protection)	CMEK, Cloud DLP, VPC Service Controls	Strong encryption (CMEK) and data loss prevention (Cloud DLP) are core to data protection. VPC SC directly addresses data leakage by enforcing a network perimeter.
ISO 27001:2022	**A.8.2** (Information Classification), **A.18.1** (Data Protection)	Dataplex, Cloud DLP, CMEK	Dataplex and Cloud DLP automate classification. CMEK ensures data is protected at rest, aligning with A.18.1.
SOC 2	**CC6.1** (Logical Access), **CC6.7** (Data Classification)	BigQuery Column-Level Security, Dataplex	Granular access control (CC6.1) is achieved via Policy Tags. Automated classification (CC6.7) is provided by Dataplex and Cloud DLP.
GDPR	**Article 32** (Security of Processing), **Article 25** (Privacy by Design)	Cloud DLP, CMEK, BigQuery Column-Level Security	DLP helps identify and redact PII, supporting the right to pseudonymization. CMEK and access controls ensure a high level of security for personal data.
HIPAA	**§ 164.312(a)(2)(iv)** (Encryption), **§ 164.312(e)** (Transmission Security)	CMEK, VPC Service Controls	CMEK is essential for encrypting ePHI at rest. VPC SC ensures that ePHI is not transmitted outside the secure perimeter, addressing transmission security.
PCI DSS	**Requirement 3** (Protect Stored Data), **Requirement 4** (Encrypt Transmission)	CMEK, VPC Service Controls	CMEK meets the requirement for protecting Cardholder Data (CHD) at rest. VPC SC prevents unauthorized network access, supporting secure transmission and storage.

Audit Evidence Generation

The implementation provides several artifacts that serve as direct evidence for GRC audits:

- **Data Classification Reports**: Generated by Dataplex and Cloud DLP, detailing the location and sensitivity of all data assets.

- **Access Logs and Audit Trails**: Cloud Audit Logs capture every data access attempt, including those blocked by VPC Service Controls or BigQuery Policy Tags, providing a complete chain of custody.
- **Encryption Key Usage Records**: Cloud KMS logs track all key usage, demonstrating adherence to key management policies.

4. Prerequisites

Before beginning the deployment, ensure the following prerequisites are met.

Required Accounts and Permissions

1. **GCP Project**: A dedicated GCP project (e.g., `prj-gcp-data-080`) must be created with **billing enabled**.
2. **IAM Permissions**: The deploying user must have the **`roles/owner`** or a custom role with equivalent permissions to enable services, create resources, and manage IAM policies. Specific roles required for the deployment steps include:
   - `roles/cloudkms.admin`
   - `roles/accesscontextmanager.policyAdmin`
   - `roles/dataplex.admin`
   - `roles/bigquery.admin`
   - `roles/storage.admin`
   - `roles/datacatalog.admin` (for Policy Tag Taxonomy creation)

Tools and Setup

1. **Google Cloud SDK (gcloud CLI)**: Must be installed and configured on the local machine.
2. **gcloud Authentication**: Authenticate and set the project context.

```
# Set your project ID
export PROJECT_ID="PRJ-GCP-DATA-080"

# Authenticate the gcloud CLI
gcloud auth login

# Set the active project
gcloud config set project $PROJECT_ID
```

Enable Required APIs

The following APIs must be enabled for the project to support the Dataplex, VPC SC, KMS, and DLP components.

```
# Enable necessary APIs
gcloud services enable \
    dataplex.googleapis.com \
    bigquery.googleapis.com \
    cloudkms.googleapis.com \
    dlp.googleapis.com \
    accesscontextmanager.googleapis.com \
    serviceusage.googleapis.com
```

5. Architecture Overview

The architecture is a **Hub-and-Spoke model** secured by a VPC Service Controls perimeter, creating a robust and centralized data governance system.

The Hub: Dataplex Lake

The **Dataplex Lake** acts as the central **Hub** for metadata, governance, and security policies. It does not store the data itself but provides a unified view and management layer over distributed data assets.

- **Dataplex Zones**: The Lake is subdivided into Zones (e.g., `raw-zone`, `curated-zone`) which define data quality, security, and governance rules for the data assets they contain.
- **Dataplex Assets**: These are the **Spokes**, which are pointers to the actual data stored in BigQuery datasets and Cloud Storage buckets. Dataplex automatically discovers and catalogs the metadata for these assets.

The Security Layers

1. **VPC Service Controls Perimeter (Network Layer)**: This is the outermost security layer. It establishes a **network perimeter** around the BigQuery, Cloud Storage, and Cloud DLP services. Any attempt to access these services from outside the perimeter (e.g., from an external network or a non-trusted project) is blocked, effectively preventing data exfiltration.

2. **Cloud KMS (Encryption Layer)**: CMEK ensures that all data within the perimeter is encrypted with keys managed by the organization, providing control over the encryption lifecycle.
3. **Cloud DLP (Classification Layer)**: This service scans data assets and applies sensitivity labels, which are then used by Dataplex and BigQuery to enforce access policies.
4. **BigQuery Column-Level Security (Access Layer)**: This is the most granular layer, using **Data Catalog Policy Tags** to restrict access to specific sensitive columns (e.g., `user_email`, `credit_card_number`) within a dataset, even if a user has general access to the table.

The architecture ensures that data is protected at rest (CMEK), in transit (VPC SC), and at the point of access (Column-Level Security), providing defense-in-depth.

6. Step-by-Step Implementation

The deployment is broken down into four main steps, executed via the `gcloud` and `bq` command-line tools.

Step 6.1: Configure Cloud KMS for CMEK

This step creates the encryption key and grants the necessary service accounts permission to use it for BigQuery and Cloud Storage.

```
export KMS_REGION="us-central1"
export KEY_RING_NAME="data-governance-keyring"
export KEY_NAME="data-encryption-key"

# 1. Create Key Ring
echo "Creating KMS Key Ring: $KEY_RING_NAME in $KMS_REGION"
gcloud kms keyrings create $KEY_RING_NAME \
    --location $KMS_REGION

# 2. Create Crypto Key
echo "Creating Crypto Key: $KEY_NAME"
gcloud kms keys create $KEY_NAME \
    --keyring $KEY_RING_NAME \
    --location $KMS_REGION \
    --purpose "encryption"

# Get the project number for service account construction
export PROJECT_NUMBER=$(gcloud projects describe $PROJECT_ID --
format="value(projectNumber)")

# 3. Grant BigQuery and Cloud Storage Service Accounts KMS Encrypter/Decrypter
role
# BigQuery Service Account (region-specific)
export BQ_SA="service-$PROJECT_NUMBER@gcp-sa-bigquery.iam.gserviceaccount.com"
# Cloud Storage Service Account (multi-region)
export GCS_SA="service-$PROJECT_NUMBER@gs-project-
accounts.iam.gserviceaccount.com"

echo "Granting KMS role to BigQuery Service Account: $BQ_SA"
gcloud kms keys add-iam-policy-binding $KEY_NAME \
    --location $KMS_REGION \
    --keyring $KEY_RING_NAME \
    --member "serviceAccount:$BQ_SA" \
    --role "roles/cloudkms.cryptoKeyEncrypterDecrypter"

echo "Granting KMS role to Cloud Storage Service Account: $GCS_SA"
gcloud kms keys add-iam-policy-binding $KEY_NAME \
    --location $KMS_REGION \
    --keyring $KEY_RING_NAME \
    --member "serviceAccount:$GCS_SA" \
    --role "roles/cloudkms.cryptoKeyEncrypterDecrypter"
```

Step 6.2: Implement VPC Service Controls Perimeter

This step creates a security perimeter to restrict access to the data services.

```
export PERIMETER_NAME="data_governance_perimeter"
# Get the policy ID for the organization
export POLICY_ID=$(gcloud access-context-manager policies list --
format="value(name)")

if [ -z "$POLICY_ID" ]; then
    echo "ERROR: Could not find an Access Context Manager Policy. Ensure you have
the necessary permissions and a policy is configured for your organization."
    exit 1
fi

# 1. Create the perimeter configuration file (perimeter.yaml)
cat << EOF > perimeter.yaml
kind: ServicePerimeter
spec:
  perimeterType: PERIMETER_TYPE_REGULAR
  description: "VPC Service Controls Perimeter for Data Governance"
  resources:
  - projects/$PROJECT_ID
  restrictedServices:
  - bigquery.googleapis.com
  - storage.googleapis.com
  - dlp.googleapis.com
EOF

# 2. Create the perimeter
echo "Creating VPC Service Controls Perimeter: $PERIMETER_NAME"
gcloud access-context-manager perimeters create $PERIMETER_NAME \
    --policy $POLICY_ID \
    --config-file perimeter.yaml
```

Step 6.3: Deploy Dataplex Lake and Assets

This step deploys the central governance hub (Dataplex Lake) and links the BigQuery and Cloud Storage data assets.

```
export LAKE_ID="governance-lake"
export ZONE_ID="raw-zone"
export BQ_DATASET_ID="governed_data"
export GCS_BUCKET_NAME="$PROJECT_ID-governed-data"
export LAKE_REGION="us-central1"

# 1. Create a Cloud Storage bucket and BigQuery dataset, enforcing CMEK
echo "Creating CMEK-protected Cloud Storage bucket: gs://$GCS_BUCKET_NAME"
gsutil mb -l $LAKE_REGION gs://$GCS_BUCKET_NAME
gsutil defkms set
projects/$PROJECT_ID/locations/$KMS_REGION/keyRings/$KEY_RING_NAME/cryptoKeys/$KEY_N
gs://$GCS_BUCKET_NAME

echo "Creating CMEK-protected BigQuery dataset: $BQ_DATASET_ID"
bq mk --location $LAKE_REGION \
    --default_kms_key
projects/$PROJECT_ID/locations/$KMS_REGION/keyRings/$KEY_RING_NAME/cryptoKeys/$KEY_N
\
    $BQ_DATASET_ID

# 2. Create Dataplex Lake
echo "Creating Dataplex Lake: $LAKE_ID"
gcloud dataplex lakes create $LAKE_ID \
    --location $LAKE_REGION \
    --display-name "Data Governance Lake" \
    --project $PROJECT_ID

# 3. Create Dataplex Zone (Raw Zone)
echo "Creating Dataplex Zone: $ZONE_ID"
gcloud dataplex zones create $ZONE_ID \
    --location $LAKE_REGION \
    --lake $LAKE_ID \
    --type RAW \
    --resource-location-type SINGLE_REGION \
    --display-name "Raw Data Zone" \
    --project $PROJECT_ID

# 4. Add BigQuery Dataset as an Asset
echo "Adding BigQuery Dataset as Dataplex Asset"
gcloud dataplex assets create bq-asset \
    --location $LAKE_REGION \
    --lake $LAKE_ID \
    --zone $ZONE_ID \
    --display-name "BigQuery Governed Data" \
    --resource-type BIGQUERY_DATASET \
    --resource-name "projects/$PROJECT_ID/datasets/$BQ_DATASET_ID" \
    --project $PROJECT_ID

# 5. Add Cloud Storage Bucket as an Asset
```

```
echo "Adding Cloud Storage Bucket as Dataplex Asset"
gcloud dataplex assets create gcs-asset \
    --location $LAKE_REGION \
    --lake $LAKE_ID \
    --zone $ZONE_ID \
    --display-name "Cloud Storage Governed Data" \
    --resource-type STORAGE_BUCKET \
    --resource-name "projects/$PROJECT_ID/buckets/$GCS_BUCKET_NAME" \
    --project $PROJECT_ID
```

Step 6.4: Implement BigQuery Column-Level Security

This step creates the Data Catalog Policy Tag Taxonomy and demonstrates how to apply a tag to a sensitive column.

```
export TAXONOMY_ID="PII_Taxonomy"
export POLICY_TAG_ID="Sensitive_PII"
export FULL_TAXONOMY_PATH="projects/$PROJECT_ID/locations/$LAKE_REGION/taxonomies"

# 1. Create a Policy Tag Taxonomy
echo "Creating Policy Tag Taxonomy: $TAXONOMY_ID"
bq mk --policy_tag_taxonomy \
    --location $LAKE_REGION \
    --display_name $TAXONOMY_ID \
    --description "Taxonomy for Personally Identifiable Information" \
    --policy_tags $POLICY_TAG_ID

# 2. Retrieve the full resource name of the Policy Tag
# This is required for the next step.
# Note: The actual Policy Tag ID is a system-generated number. We will use a
placeholder for demonstration.
# In a real deployment, you would query the Data Catalog API to get the full
resource name.
# For this guide, we assume the full path is:
export POLICY_TAG_RESOURCE_NAME="$FULL_TAXONOMY_PATH/$(bq show --format=json --
policy_tag_taxonomy $FULL_TAXONOMY_PATH/$TAXONOMY_ID | jq -r
'.taxonomyId')/policyTags/$(bq show --format=json --policy_tag_taxonomy
$FULL_TAXONOMY_PATH/$TAXONOMY_ID | jq -r '.policyTags[] | select(.displayName ==
"Sensitive_PII") | .policyTagId')"

# 3. Create a sample table and apply the Policy Tag to a column
# First, create a sample table without the tag
echo "Creating sample table 'users' in $BQ_DATASET_ID"
bq mk --table $BQ_DATASET_ID.users \
    user_id:INTEGER,user_name:STRING,user_email:STRING,join_date:DATE

# 4. Apply the Policy Tag to the 'user_email' column
echo "Applying Policy Tag to 'user_email' column"
# The schema file for the patch operation
cat << EOF > users_schema_patch.json
[
  {
    "name": "user_id",
    "type": "INTEGER"
  },
  {
    "name": "user_name",
    "type": "STRING"
  },
  {
    "name": "user_email",
    "type": "STRING",
    "policyTags": [
      "$POLICY_TAG_RESOURCE_NAME"
```

```
    ]
  },
  {
    "name": "join_date",
    "type": "DATE"
  }
]
EOF

# Patch the table schema
bq update --schema users_schema_patch.json $BQ_DATASET_ID.users
```

7. Validation & Testing

Validation is crucial to ensure that the security and governance controls are correctly enforced.

7.1. Dataplex and CMEK Validation

Verify that the Dataplex components are active and that the data assets are correctly linked and protected by CMEK.

```
# 1. Verify Lake and Assets status
echo "Verifying Dataplex Lake status..."
gcloud dataplex lakes describe $LAKE_ID --location $LAKE_REGION --
format="value(state)"

echo "Listing Dataplex Assets..."
gcloud dataplex assets list --lake $LAKE_ID --zone $ZONE_ID --location
$LAKE_REGION

# Expected Output: State should be 'ACTIVE' for the Lake and Assets.

# 2. Verify CMEK on BigQuery Dataset
echo "Verifying CMEK on BigQuery Dataset..."
bq show --format=json $BQ_DATASET_ID | grep defaultKmsKeyName

# Expected Output: The defaultKmsKeyName should match the key created in Step 6.1.
```

7.2. VPC Service Controls Validation

The most effective way to validate VPC SC is to attempt a prohibited action.

1. **Simulate Exfiltration Attempt**: From a machine or a project *outside* the defined perimeter, attempt to copy data from the governed Cloud Storage bucket (`gs://$GCS_BUCKET_NAME`) to an external, public bucket.

```
# Run this from an unauthorized environment
gsutil cp gs://$GCS_BUCKET_NAME/test_file.txt gs://external-untrusted-bucket/
```

   Expected Result: The command should fail with a `403 Request is prohibited by organization's policy` error, confirming the perimeter is active.

7.3. Column-Level Security Validation

This test confirms that the Policy Tag is correctly restricting access to the sensitive column.

1. **Setup Test User**: Create a test user (e.g., `user-analyst@example.com`) and grant them the basic BigQuery Viewer role.

```
gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member="user:user-analyst@example.com" \
    --role="roles/bigquery.dataViewer"
```

2. **Test 1 (Failure)**: The test user attempts to query the `user_email` column.

```
-- Run as user-analyst@example.com
SELECT user_id, user_email FROM governed_data.users;
```

   Expected Result: The query should fail with a `Permission denied` error on the `user_email` column, as the user lacks the Policy Tag Viewer role.

3. **Test 2 (Success)**: Grant the test user the **Policy Tag Viewer** role on the specific taxonomy.

```
# Grant the role on the Policy Tag Taxonomy
gcloud data-catalog taxonomies add-iam-policy-binding
$FULL_TAXONOMY_PATH/$TAXONOMY_ID \
    --member="user:user-analyst@example.com" \
    --role="roles/datacatalog.policyTagViewer"
```

4. **Test 3 (Success)**: The test user attempts the same query again. **Expected Result**: The query should now succeed, demonstrating that access to the sensitive column is explicitly granted only via the Policy Tag role.

8. Troubleshooting

This section outlines common issues encountered during deployment and operation, along with their resolutions.

Issue	Potential Cause	Resolution
403 Permission Denied on BigQuery/GCS	**VPC Service Controls** is blocking the request. The source IP/VPC is outside the perimeter.	Verify the source IP/VPC is within the perimeter's access level, or the service account is granted the necessary access level within the VPC SC configuration. Check the **Access Context Manager** audit logs for details on the blocked request.
CMEK Error on resource creation	BigQuery/GCS Service Account lacks `cloudkms.cryptoKeyEncrypterDecrypter` role.	Re-run Step 6.1, ensuring the correct service account (with project number) is granted the role. Note that BigQuery service accounts are region-specific.
Dataplex Asset Fails to Discover	Dataplex Service Account lacks permissions on the underlying data.	Grant the Dataplex service account (`service-PROJECT_NUMBER@gcp-sa-dataplex.iam.gserviceaccount.com`) the `roles/storage.objectViewer` and `roles/bigquery.dataViewer` roles on the respective data assets.
`gcloud access-context-manager perimeters create` fails	The `POLICY_ID` is not found, or the user lacks `accesscontextmanager.policyAdmin` permission.	Ensure the user has the correct IAM role at the Organization level. Verify the organization has an active Access Context Manager policy.
Policy Tag not restricting access	The Policy Tag Viewer role is too broadly applied (e.g., at the project level) or the column is not correctly tagged.	Ensure the Policy Tag Viewer role is only granted on the specific Policy Tag Taxonomy. Verify the table schema patch in Step 6.4 was successful and the column is correctly marked.

9. Cost Optimization

While security and governance are paramount, the following strategies can help manage the operational costs of the deployed services.

1. **Dataplex Tier Selection**: Utilize the **Standard tier** for Dataplex where possible. The Standard tier provides core governance features at a lower cost than the Premium tier, which is only necessary for advanced features like automated data quality enforcement.

2. **BigQuery Storage Management**: Implement **data lifecycle policies** to move older, less-frequently accessed data to cheaper storage tiers (e.g., Long-Term Storage). BigQuery automatically manages this, but proper partitioning and clustering can further reduce the amount of data scanned and stored.
3. **Cloud DLP Scanning Optimization**:
    - **Targeted Scans**: Configure DLP jobs to scan only high-risk data sources or specific tables/buckets known to contain sensitive data, rather than scanning the entire data lake.
    - **Sampling**: Utilize **data sampling** within DLP scan configurations. Instead of scanning every record, sample a statistically significant subset to identify sensitive data locations, significantly reducing scanning costs.
4. **Cloud KMS Key Rotation**: While key rotation is a security best practice, it does not directly reduce cost. However, consolidating key management under a single Key Ring and Crypto Key (as done in Step 6.1) simplifies billing and management overhead.

10. Security Best Practices

Beyond the core implementation, adhering to these best practices ensures the long-term security and compliance of the data governance framework.

1. **Principle of Least Privilege (PoLP)**:
    - **Service Accounts**: Strictly limit the IAM roles granted to service accounts (e.g., BigQuery, Dataplex) to the absolute minimum required for their function. Avoid granting broad roles like `Editor` or `Owner`.
    - **User Access**: Use **BigQuery Column-Level Security** as the primary mechanism for granting access to sensitive data, rather than relying on broad dataset-level roles.
2. **Key Rotation Policy**: Configure Cloud KMS to automatically rotate the `data-encryption-key` every 90 days (or a period mandated by your organization's policy). This limits the exposure window of any single key version.

```
# Example: Set key rotation to 90 days (7776000 seconds)
gcloud kms keys update $KEY_NAME \
    --location $KMS_REGION \
    --keyring $KEY_RING_NAME \
    --rotation-period 7776000s
```

3. **Regular DLP Scans and Remediation**: Schedule recurring Cloud DLP jobs to scan new or modified data assets. Integrate the DLP findings with Dataplex to ensure the data catalog's sensitivity labels are always up-to-date. Establish an automated or semi-automated process for remediating findings (e.g., tokenization or redaction).

4. **Centralized Audit Logging**: Ensure that **Data Access audit logs** are enabled for all governed services (BigQuery, Cloud Storage, Cloud KMS, Dataplex). Export these logs to a secure, centralized log sink (e.g., a separate, secured BigQuery dataset in a dedicated logging project) to ensure immutability and provide a complete audit trail for forensic analysis.

5. **VPC SC Access Levels**: Define granular **Access Levels** within VPC Service Controls to specify *who* (e.g., specific users, service accounts) and *from where* (e.g., specific IP ranges, devices) can access the services within the perimeter. This prevents unauthorized access even from within the trusted network.

Appendix: Cleanup

To completely remove all resources created by this deployment, execute the following commands. **Warning: This will permanently delete all data and resources.**

```
# Set environment variables again if the session was closed
export PROJECT_ID="PRJ-GCP-DATA-080"
export PROJECT_NUMBER=$(gcloud projects describe $PROJECT_ID --
format="value(projectNumber)")
export LAKE_ID="governance-lake"
export ZONE_ID="raw-zone"
export BQ_DATASET_ID="governed_data"
export GCS_BUCKET_NAME="$PROJECT_ID-governed-data"
export LAKE_REGION="us-central1"
export KMS_REGION="us-central1"
export KEY_RING_NAME="data-governance-keyring"
export KEY_NAME="data-encryption-key"
export PERIMETER_NAME="data_governance_perimeter"
export POLICY_ID=$(gcloud access-context-manager policies list --
format="value(name)")
export TAXONOMY_ID="PII_Taxonomy"

# 1. Delete Dataplex Assets, Zones, and Lake
echo "Deleting Dataplex Assets, Zones, and Lake..."
gcloud dataplex assets delete bq-asset --lake $LAKE_ID --zone $ZONE_ID --location
$LAKE_REGION --quiet
gcloud dataplex assets delete gcs-asset --lake $LAKE_ID --zone $ZONE_ID --location
$LAKE_REGION --quiet
gcloud dataplex zones delete $ZONE_ID --lake $LAKE_ID --location $LAKE_REGION --
quiet
gcloud dataplex lakes delete $LAKE_ID --location $LAKE_REGION --quiet

# 2. Delete BigQuery Dataset and Cloud Storage Bucket
echo "Deleting BigQuery Dataset and Cloud Storage Bucket..."
bq rm -r -f $BQ_DATASET_ID
gsutil rm -r gs://$GCS_BUCKET_NAME

# 3. Delete KMS Key and Key Ring
echo "Deleting KMS Key and Key Ring..."
gcloud kms keys destroy $KEY_NAME --keyring $KEY_RING_NAME --location $KMS_REGION
--quiet
gcloud kms keyrings delete $KEY_RING_NAME --location $KMS_REGION --quiet

# 4. Delete VPC Service Controls Perimeter
echo "Deleting VPC Service Controls Perimeter..."
gcloud access-context-manager perimeters delete $PERIMETER_NAME --policy
$POLICY_ID --quiet

# 5. Delete the Policy Tag Taxonomy
echo "Deleting Policy Tag Taxonomy..."
# Note: Taxonomy ID is required for this command.
# The following command is conceptual as the actual ID is dynamic.
# Replace <TAXONOMY_ID_NUMBER> with the actual ID retrieved after creation.
```

```
# bq rm --policy_tag_taxonomy
projects/$PROJECT_ID/locations/$LAKE_REGION/taxonomies/<TAXONOMY_ID_NUMBER>
```