Obtaining and Preparing SSL Certificates for PostgreSQL in GCP

Overview

When connecting to a PostgreSQL instance hosted in Google Cloud Platform (GCP), secure communication requires SSL certificates. This guide outlines the process for obtaining these certificates from GCP and preparing them for use in your applications.

Prerequisites

- Access to Google Cloud Console
- Permissions to manage your PostgreSQL instance
- OpenSSL installed on your local machine

Obtaining Certificates

Step 1: Access Your PostgreSQL Instance in GCP

1. Log in to the Google Cloud Console
2. Navigate to SQL in the left navigation menu
3. Select your PostgreSQL instance from the list

Step 2: Create and Download Certificates

1. Go to the Connections tab
2. Under the SSL section, click Create a client certificate
3. Enter a descriptive name for the certificate (e.g., "collibra-connection")
4. Click Create
5. Download all three generated files:
    - client-cert.pem (client certificate)
    - client-key.pem (client private key)
    - server-ca.pem (server CA certificate)

Step 3: Prepare Certificates for Use

After downloading the certificates, combine the client certificate and key into a PKCS12 (.p12) format:

```
openssl pkcs12 -export -in client-cert.pem -inkey client-key.pem -out
client.p12 -name "client"
```

> Note: When running this command, you'll be prompted to create and confirm a password. Store this password securely as it will be required when configuring your database connection.

Configuring PostgreSQL Connection in Collibra Edge

After obtaining and preparing your SSL certificates from GCP, follow these steps to configure the connection in Collibra Edge:

1. Navigate to the connection configuration screen in Collibra Edge.
2. Enter the following connection details:
    - Driver class name: cdata.jdbc.postgresql.PostgreSQLDriver
    - Driver jar: postgresql-8963.jar (or the appropriate PostgreSQL driver JAR file)
    - Connection string: jdbc:postgresql://[HOST]:[PORT]/[DATABASE_NAME] Example: jdbc:postgresql://[HOST_IP]:5432/[DATABASE_NAME]
3. Configure the SSL properties by adding the following properties:

Name	Type	Value Type	Value
SSLClientCert	File	Plaintext	client.p12
SSLClientCertType	Text	Plaintext	PFXFILE
SSLClientCertPassword	Text	Plaintext	[Password created during p12 generation]
UseSSL	Text	Plaintext	true
SSLServerCert	File	Plaintext	server-ca.pem

4. Explanation of SSL properties:
    - SSLClientCert: The p12 file created by combining client-cert.pem and client-key.pem
    - SSLClientCertType: Set to PFXFILE to indicate a PKCS12/PFX format certificate
    - SSLClientCertPassword: The password you created when generating the p12 file
    - UseSSL: Set to true to enable SSL for the connection
    - SSLServerCert: The server CA certificate downloaded from GCP
5. After entering all details, click the "Test Connection" button to verify the configuration.
6. If the connection test is successful, click "Edit" to save the connection.

Certificate Management

Certificate Validity

GCP-generated certificates typically remain valid for 10 years. You can check the validity period using:

```
openssl x509 -in client-cert.pem -noout -dates
```

Certificate Rotation

For security best practices, periodically rotate your certificates. In GCP, you can create new certificates without immediately invalidating existing ones, allowing for a smooth transition in production environments.

Troubleshooting Tips

- PKIX path building failed error: Ensure you're using the correct server CA certificate
- Connection issues: Verify the certificate password is correctly specified in your configuration
- SSL handshake failures: Confirm that all required certificate files are properly referenced
- Connection error: Validate that within the connection string there is database defined and not the instance
- If you encounter a "PKIX path building failed" error, verify that:
    - The server-ca.pem file contains the correct certificate
    - The client.p12 file was created correctly
    - The SSLClientCertPassword matches the password used when creating the p12 file