**PRACTICAL: 8**

**AIM:**

Security Best Practices in Google Cloud:

1. Using Customer-Supplied Encryption Keys with Cloud Storage.
2. Using Customer-Managed Encryption Keys with Cloud Storage and Cloud KMS.
3. Using Cloud Security Scanner to find vulnerabilities in an App Engine application.
4. Configuring Identity Aware Proxy to Protect a Project.
5. Configuring and Using Credentials with Secret Manager.

**THEORY:**

**8.1 Using Customer-Supplied Encryption Keys with Cloud Storage**

Cloud Storage always encrypts your data on the server side with a Google-managed encryption key, before it is written to disk, at no additional charge. As an alternative to a Google-managed server-side encryption key, you can choose to provide your own AES-256 key, encoded in standard Base64. This key is known as a customer-supplied encryption key.

Here we will configure customer-supplied encryption keys (CSEK) for Google Cloud storage. Files will then be uploaded into a storage bucket. Here we will generate a new encryption key and rotate CSEK keys.

Cloud storage does not permanently store your key on Google’s servers or otherwise manage your key. Instead, you provide your key for each Cloud Storage operation and your key is purged from Google’s servers after the operation is complete. Cloud Storage stores only a cryptographic hash of the key so that future requests can be validated against the hash. Your key cannot be recovered from this hash and the hash cannot be used to decrypt your data. If you lose your CSEK, you will permanently lose access to all of your data encrypted with the key.

**8.2 Using Customer-Managed Encryption Keys with Cloud Storage and Cloud KMS**

Cloud KMS is cloud-hosted key management service. Encryption keys are created by Cloud KMS and managed by you in the same manner you would manage them on-premises. Using Cloud KMS you can generate, use, rotate and destroy AES256 symmetric encryption keys for direct use by all of your cloud services.

Here we will use Cloud KMS to create KeyRings and CryptoKeys and then use those keys with Cloud Storage to set default keys on buckets, and encrypt individual objects with a Cloud KMS key. Additionally, you will manually perform server-side encryption with your KMS keys, and upload encrypted data to Google Cloud storage.

KMS permissions will be managed with IAM, and Cloud Audit Logging will be used to view all activity for CryptoKeys and KeyRings.

**8.3 Using Cloud Security Scanner to find vulnerabilities in an App Engine Application**

Cloud Security Scanner is used to scan an App Engine application for vulnerabilities. Cloud Security Scanner is a web security scanner for common vulnerabilities in Google App Engine applications. It can automatically scan and detect four common vulnerabilities, including cross-site-scripting (XSS), Flash Injection, mixed content (HTTP in HTTPS) and outdated/insecure libraries. It enables early identification and delivers very low false positive rates. Here by setting up the environment we can easily run, schedule, and mange security scans and it is free for Google Cloud Platform users.

**8.4 Configuring Identity Aware Proxy to Protect a Project**

Here you can configure and enable Cloud Identity Aware Proxy (Cloud IAP) to project an application running in App Engine. Cloud IAP controls access to your cloud applications running on Google Cloud IAP works by verifying a user’s identity and determining if that user should be allowed to access the application. Cloud IAP is building block toward BeyondCorp an enterprise security model that enables every employee to work from untrusted networks without the use of VPN.

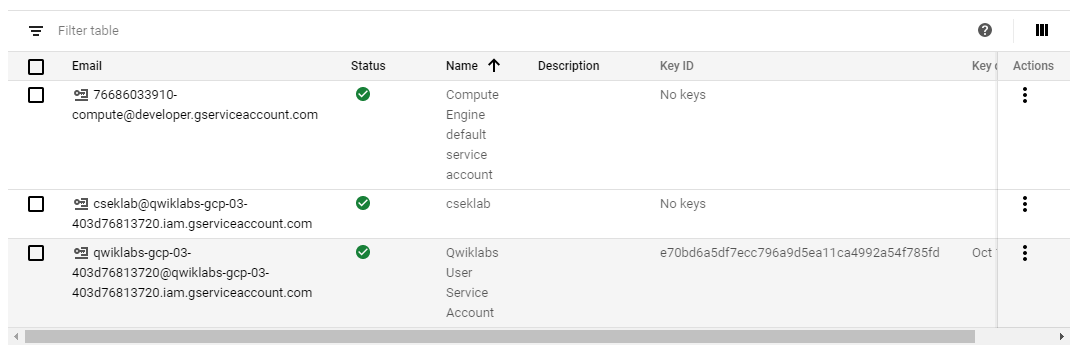
**8.5 Configuring and Using Credentials with Secret Manager**

Here you will use Secret Manager from Google Cloud Console and the Command Line Interface (CLI) to create and use a secret, replace a secret and finally reinstate an older version of a secret.

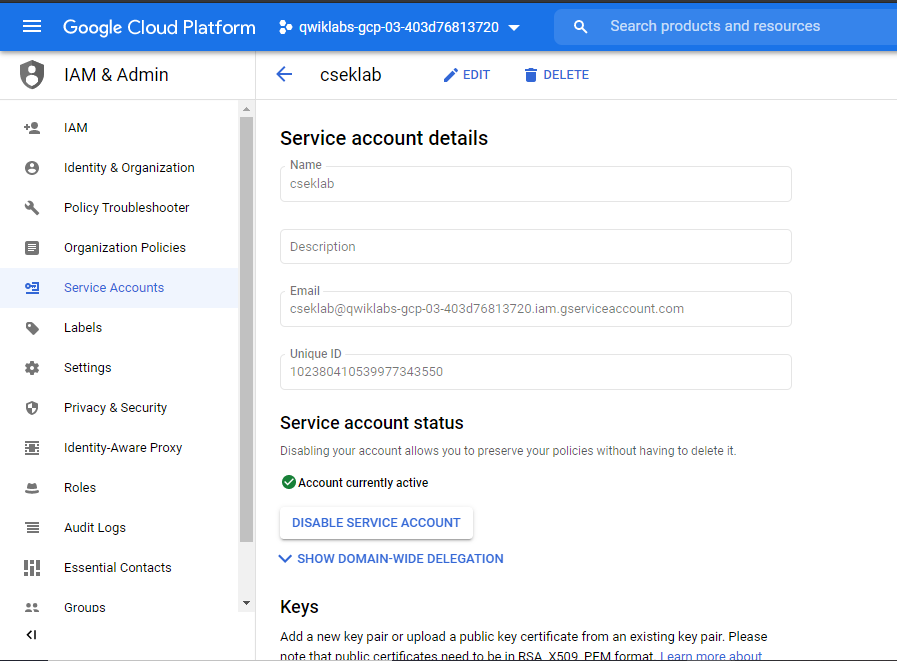
Secret Manager is available in Google Cloud Console. It is also available from the command line using the CLI or from a program, using the REST API or one of the supported Software Development Kits (SDKs) Supported SDKs include a complete list of programs.

**OUTPUT:**

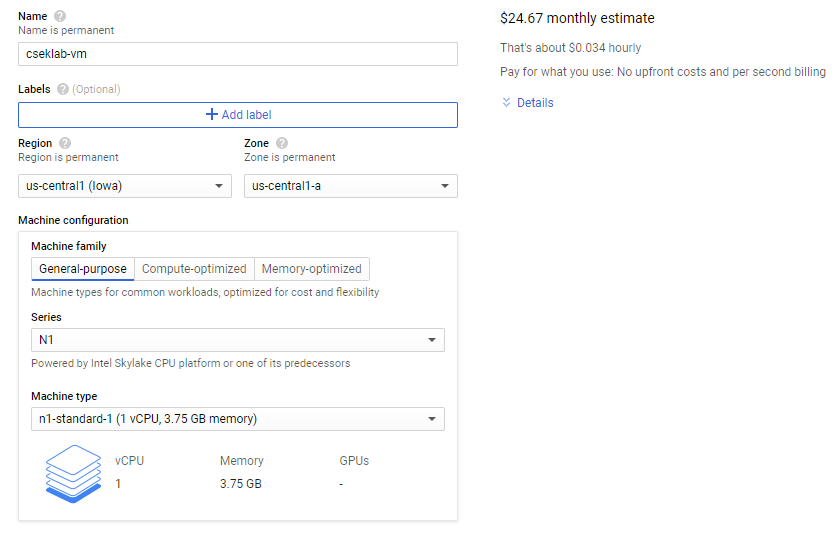
**8.1 Using Customer-Supplied Encryption Keys with Cloud Storage**



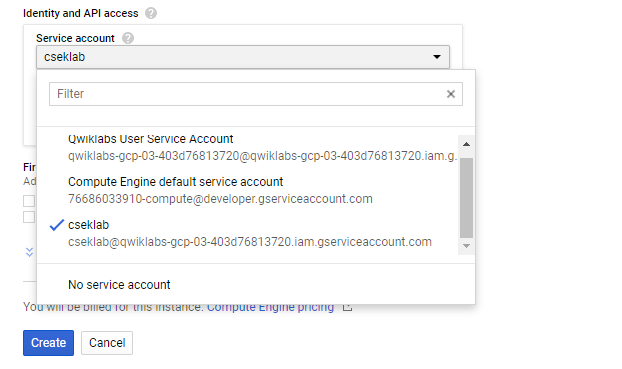
**Service account dashboard**



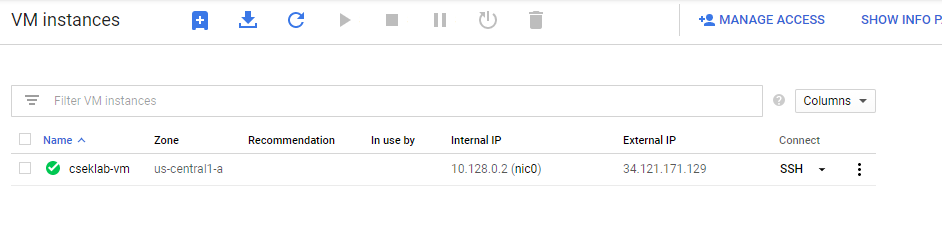
**Creating IAM service account**



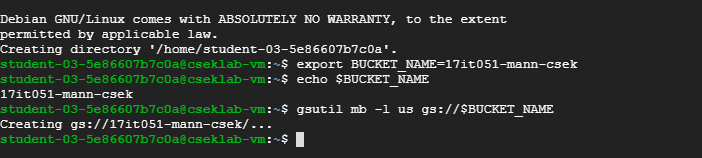
**Creating VM instance**



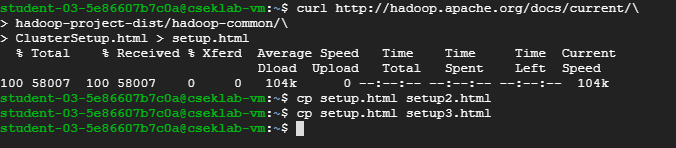
**Linking service account created in previous step**



**VM instance created**

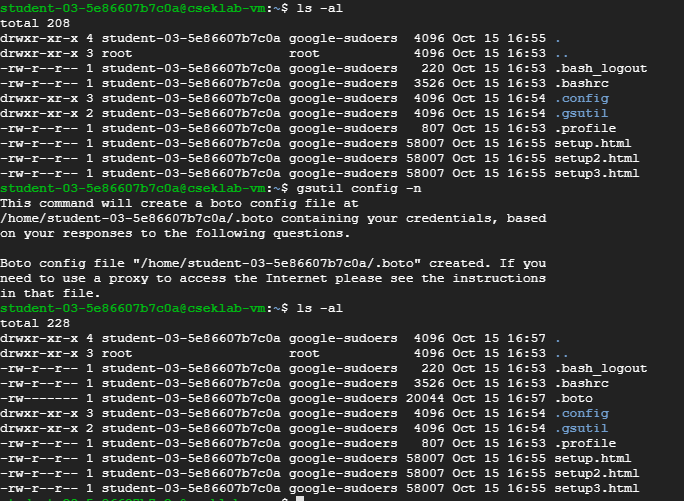


**Creating a Cloud Storage bucket**

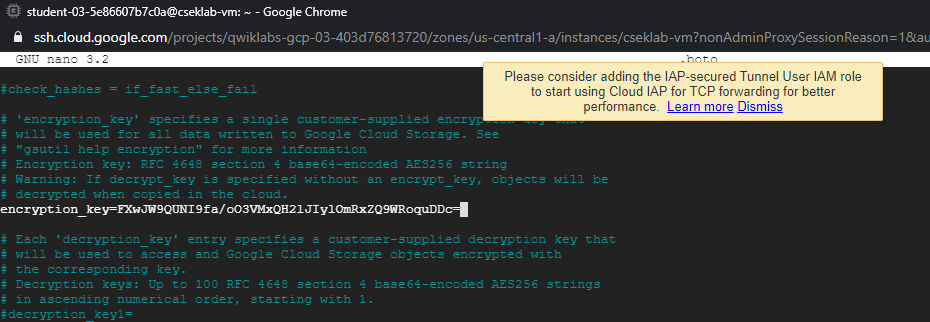


**Downloading sample file using CURL and making two copies of it**

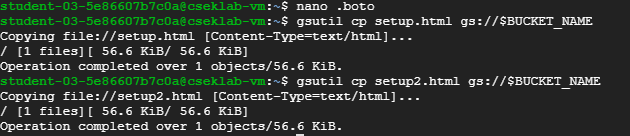




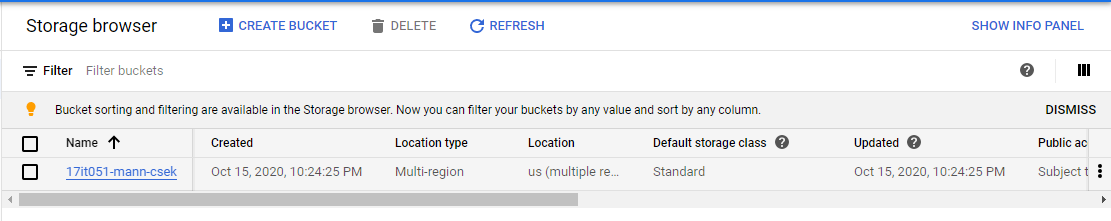
**Configuring Customer-Supplied Encryption Keys**



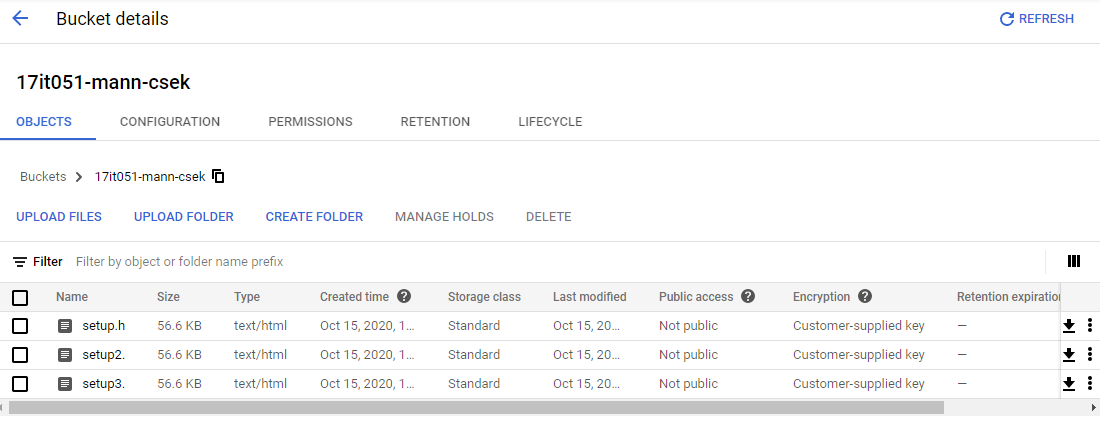
**Adding encryption\_key field in .boto file**



**Creating two setups files**



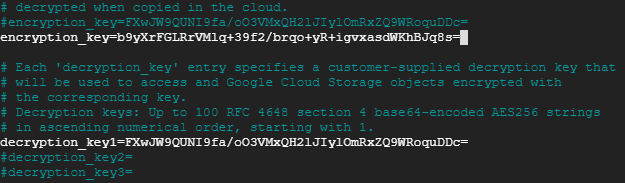
**Bucket successfully created**



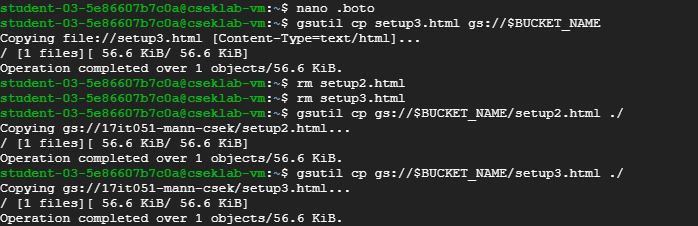
**Observing bucket details which contain two setups files which uploaded via cloud shell**



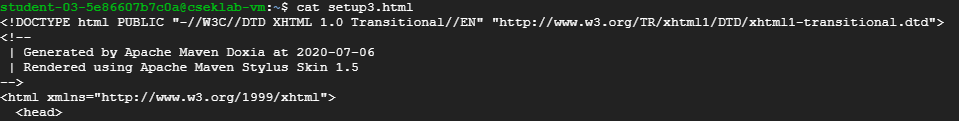
**Generating another key.**



**Rotating keys by adding new key to encryption\_key and replacing decryption\_key with old encryption\_key**

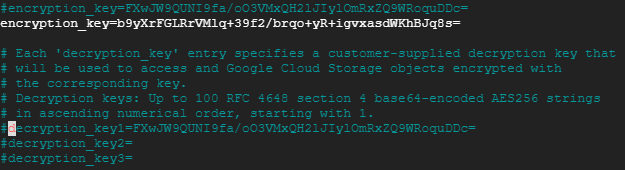


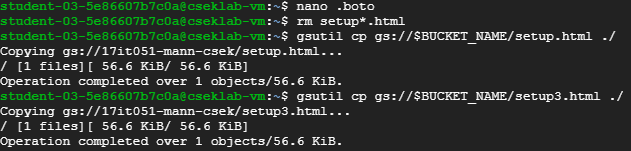




**Creating and observing setup files**

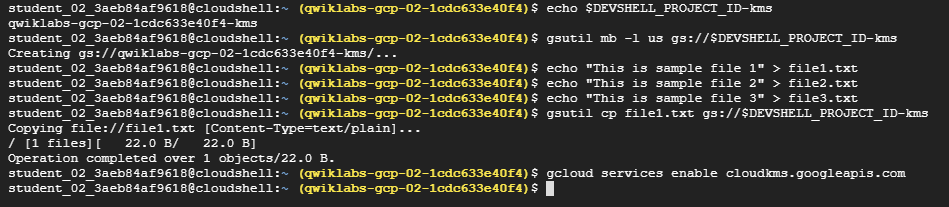




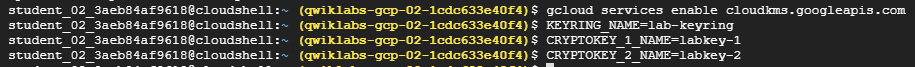


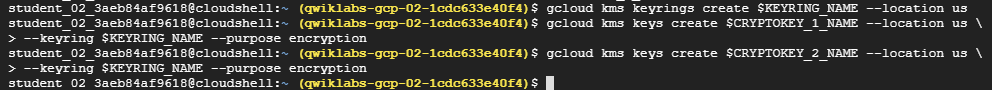
**Rewriting an encrypted file causes the file to be decrypted it using the decryption\_key1 that we previously set and encrypts the file with new encryption\_key**

**8.2 Using Customer-Managed Encryption Keys with Cloud Storage and Cloud KMS**

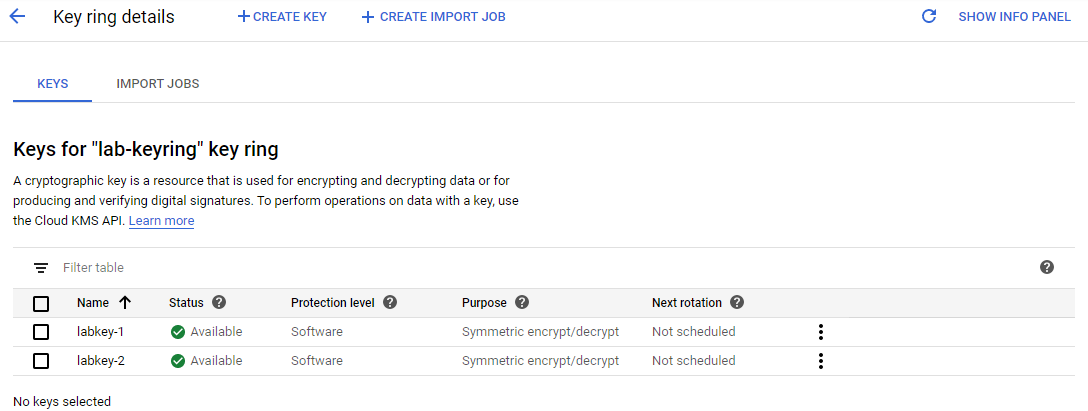


**Creating Cloud Storage bucket and Enabling Cloud KMS**



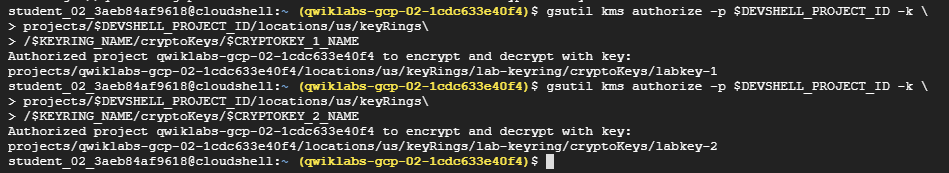


**Creating Keyring and Cryptokey**

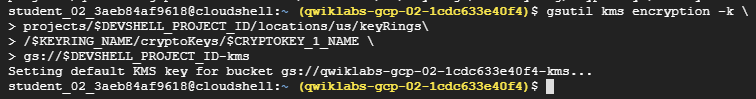


**Created encryption keys**



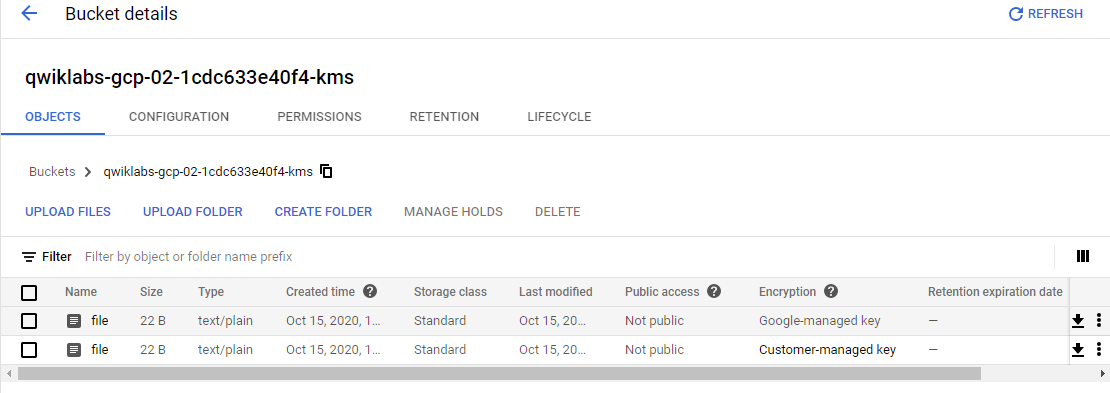


**Adding default key for bucket**

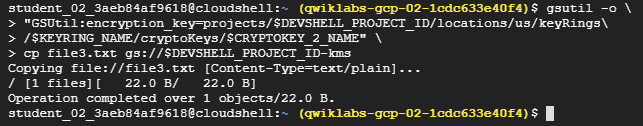




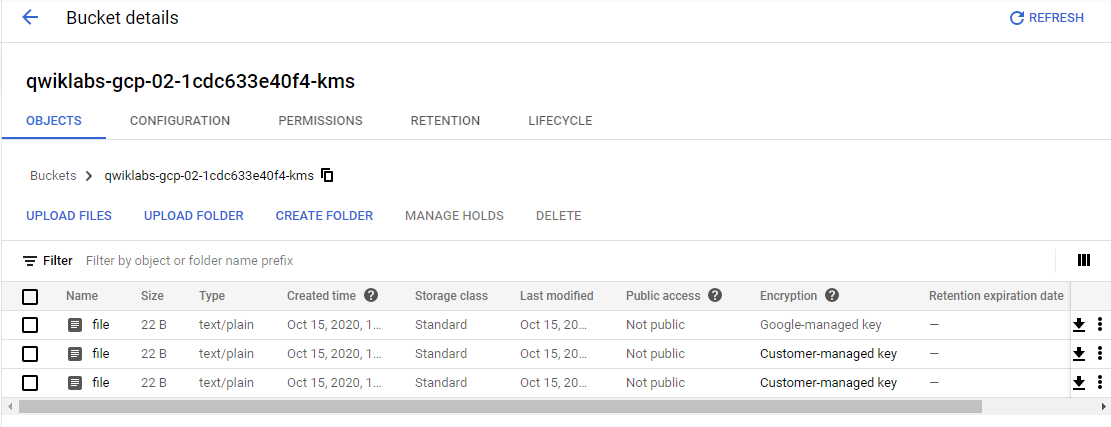
**Setting default key for a bucket**



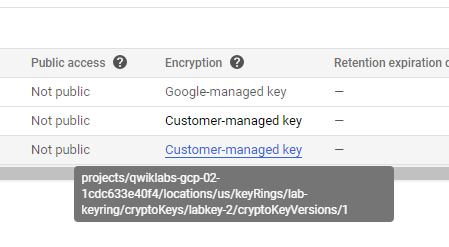
**Files**



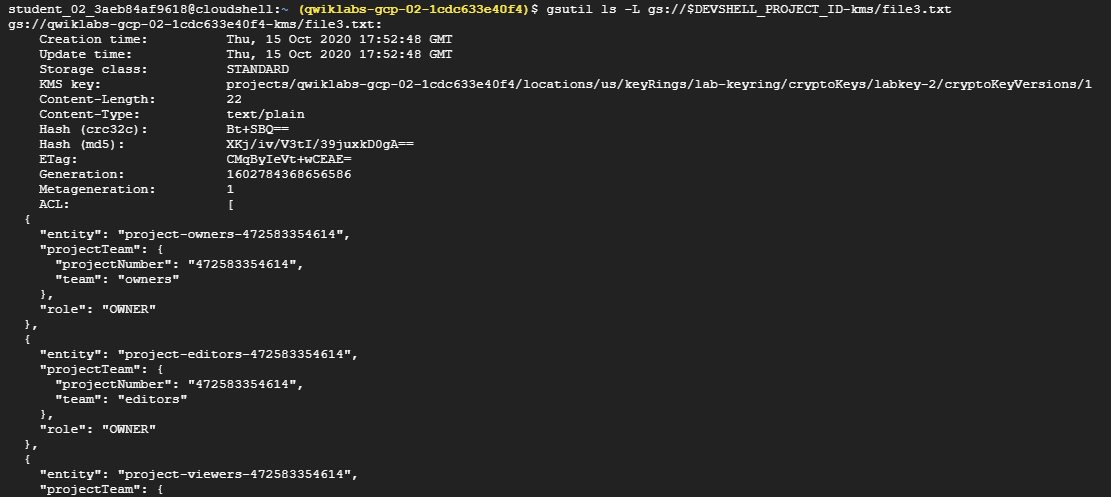
**Encrypting an object with a specific key**



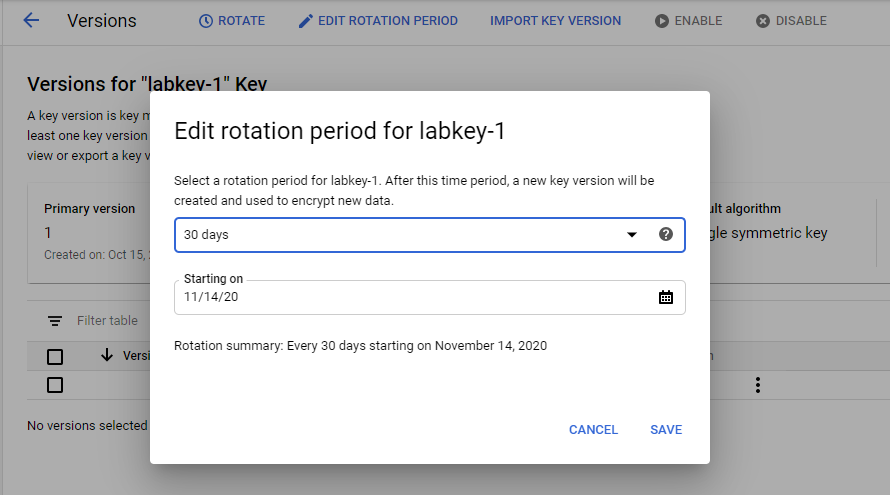
**3 files are added.**



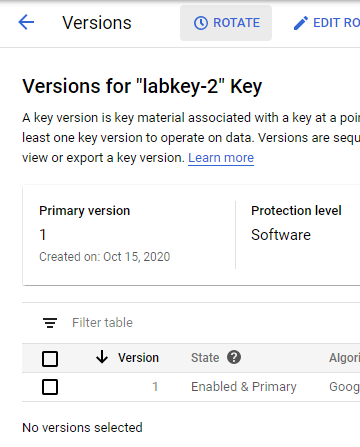
**Observing 2 types of encryption key methods**

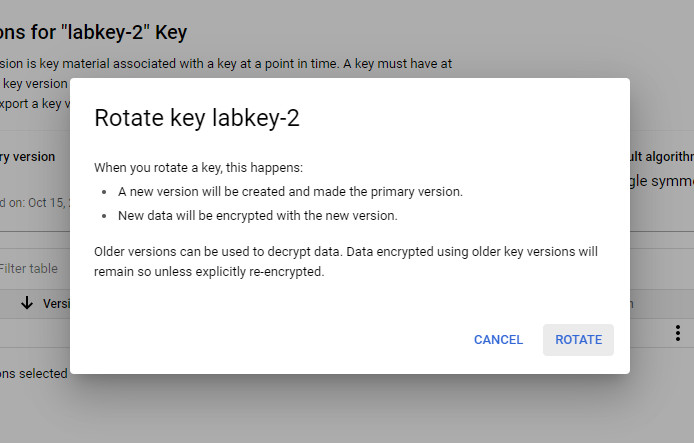


**Identifying the key used to encrypt an object and similarly did for all files**

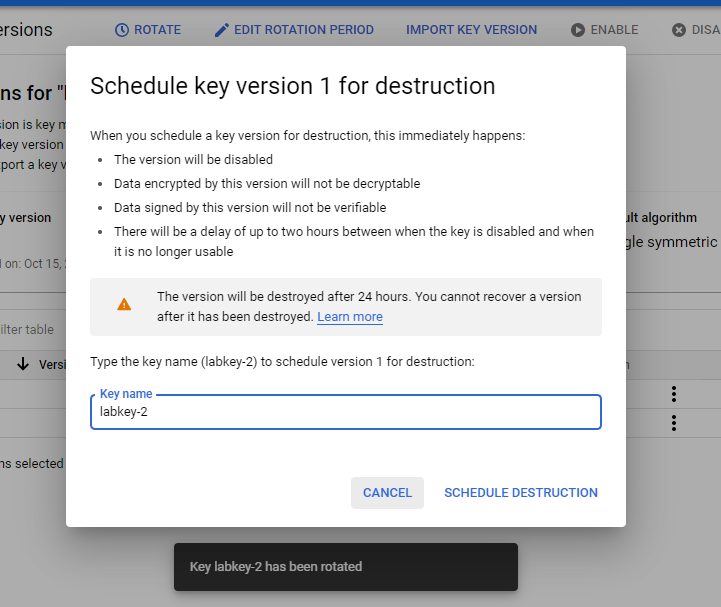


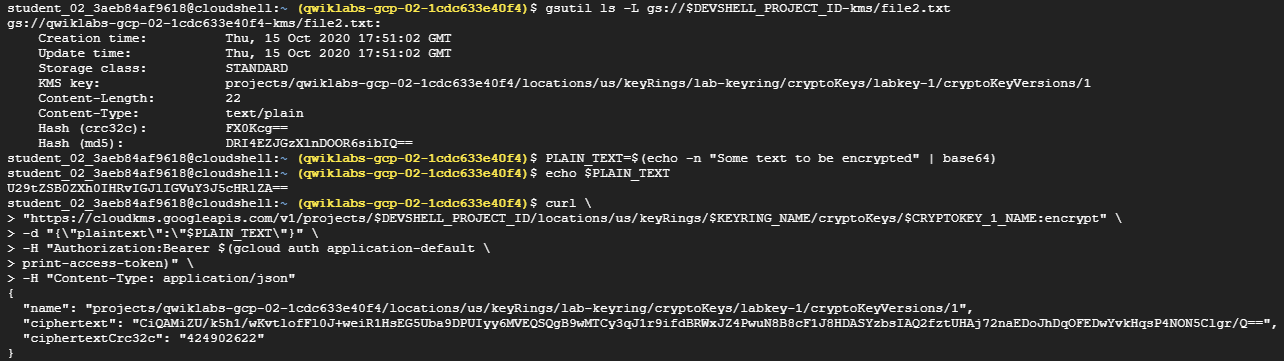
**Setting Rotation period**

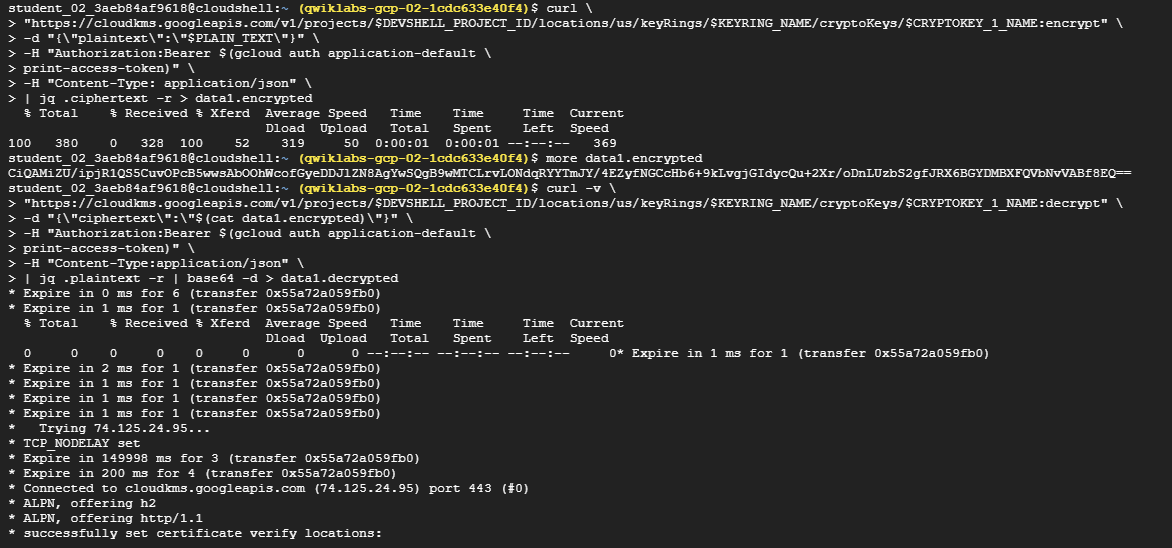




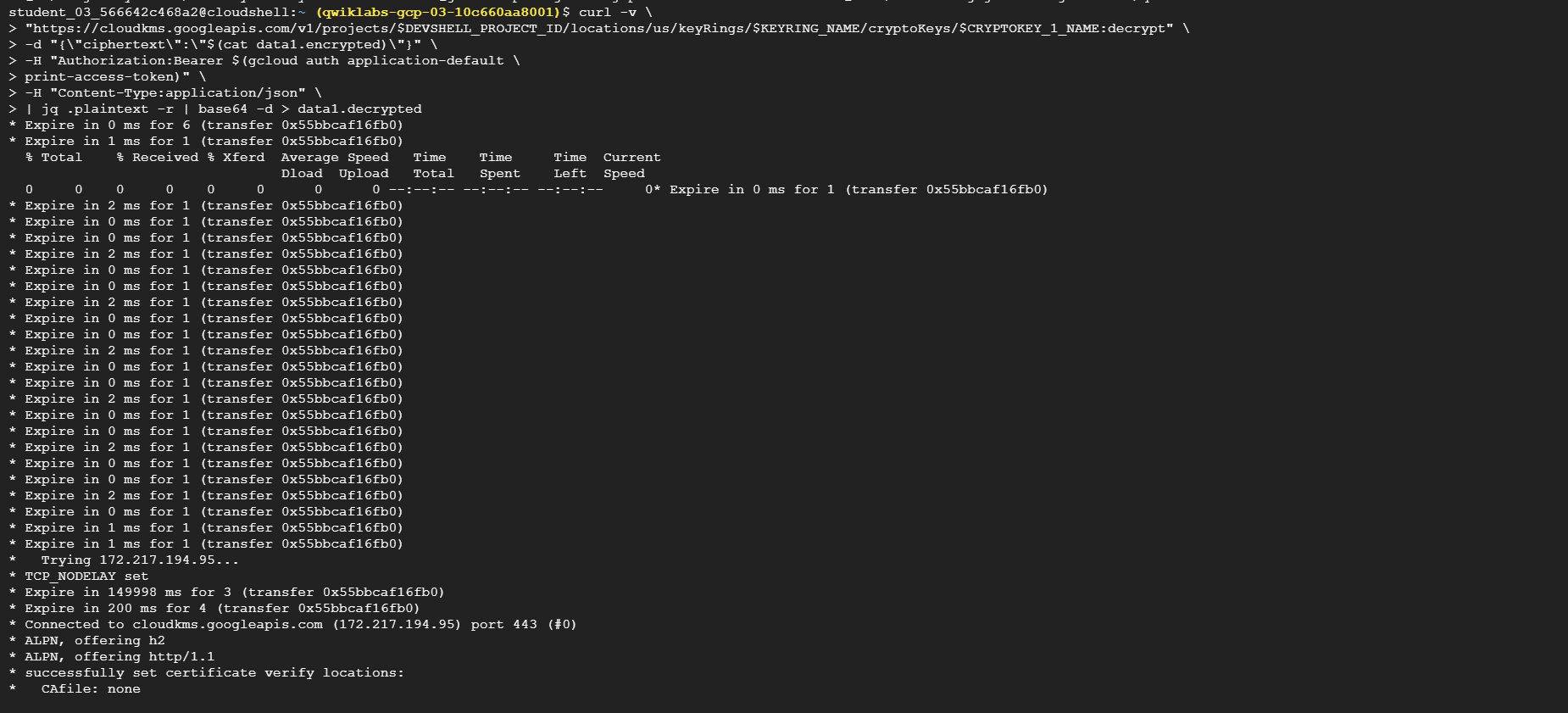
**Manually Rotating the keys**

**Destructing the keys**

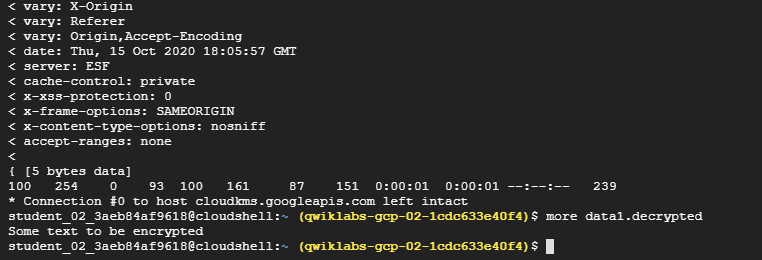




**Encrypting data with REST API**

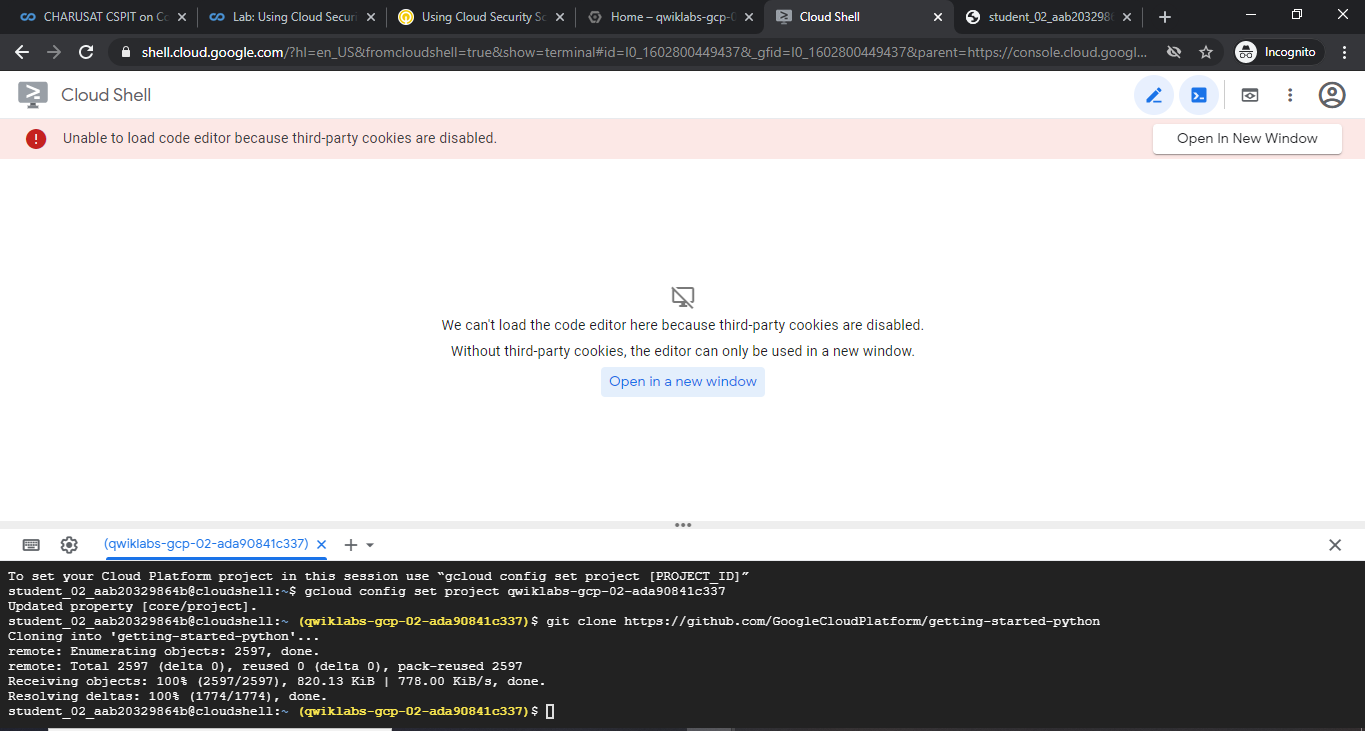


**Decrypting the data**

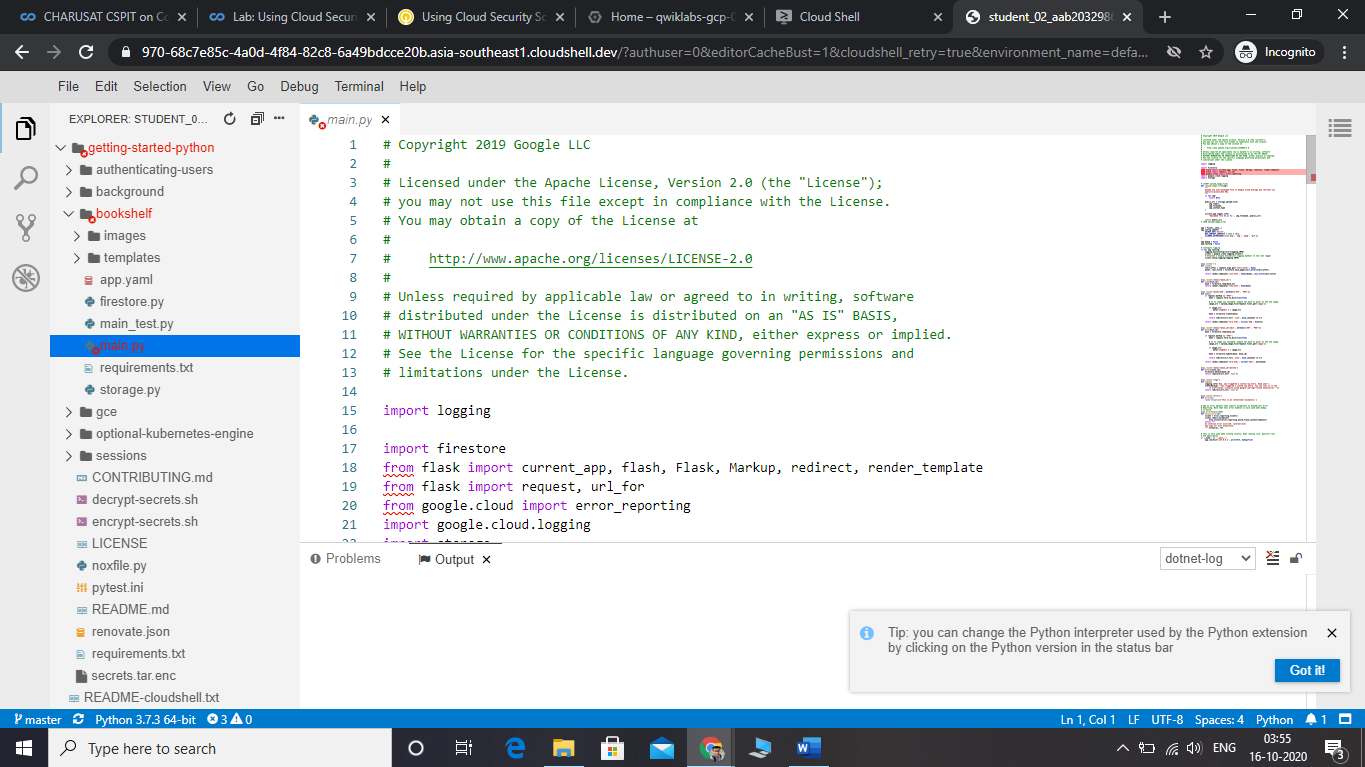


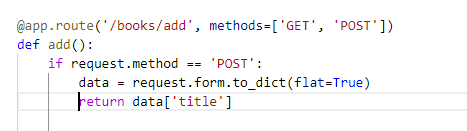
**Observing decrypted data**

**8.3 Using Cloud Security Scanner to find vulnerabilities in an App Engine application**

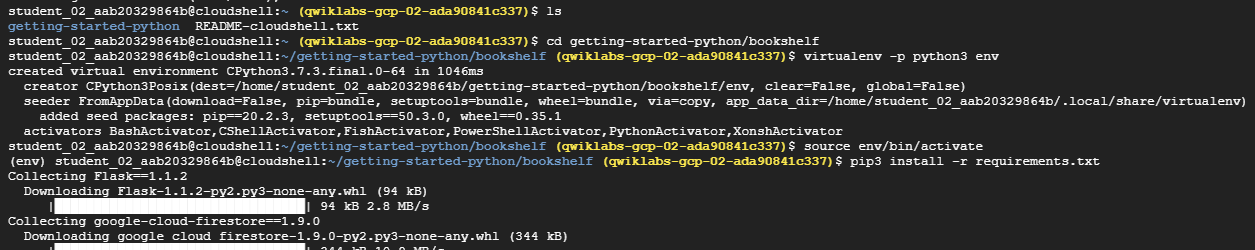


**Getting the sample app**

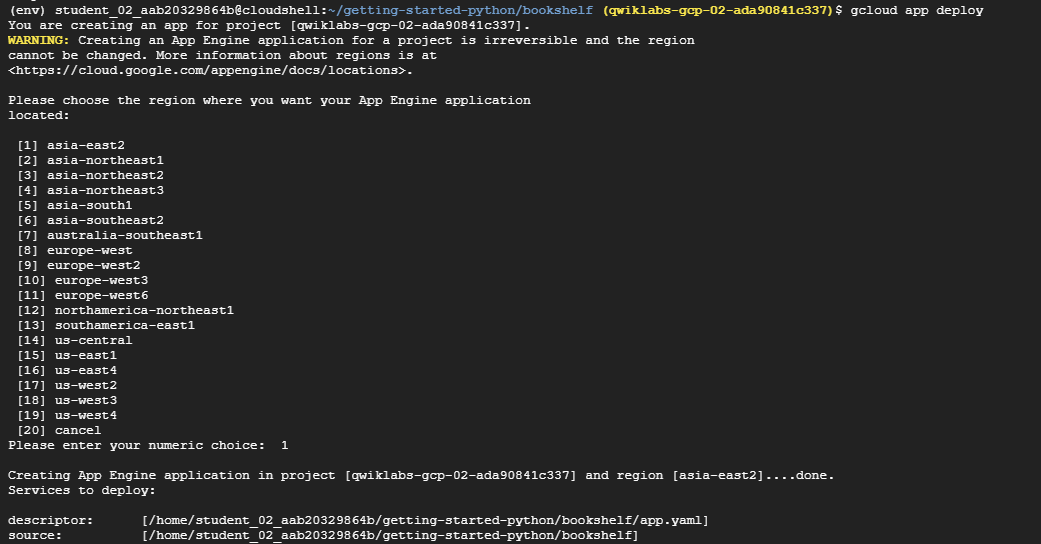




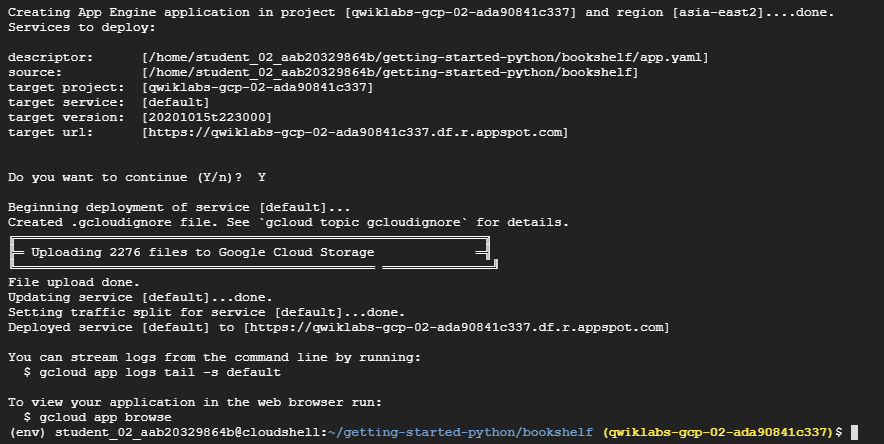
**Adding this line to make site vulnerable using code editor**



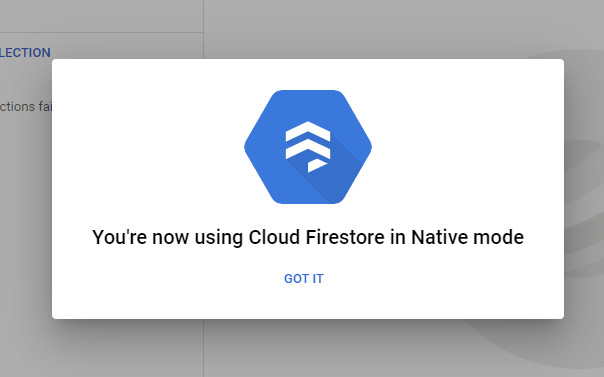
**Deploying an app engine application by installing all the necessary modules**

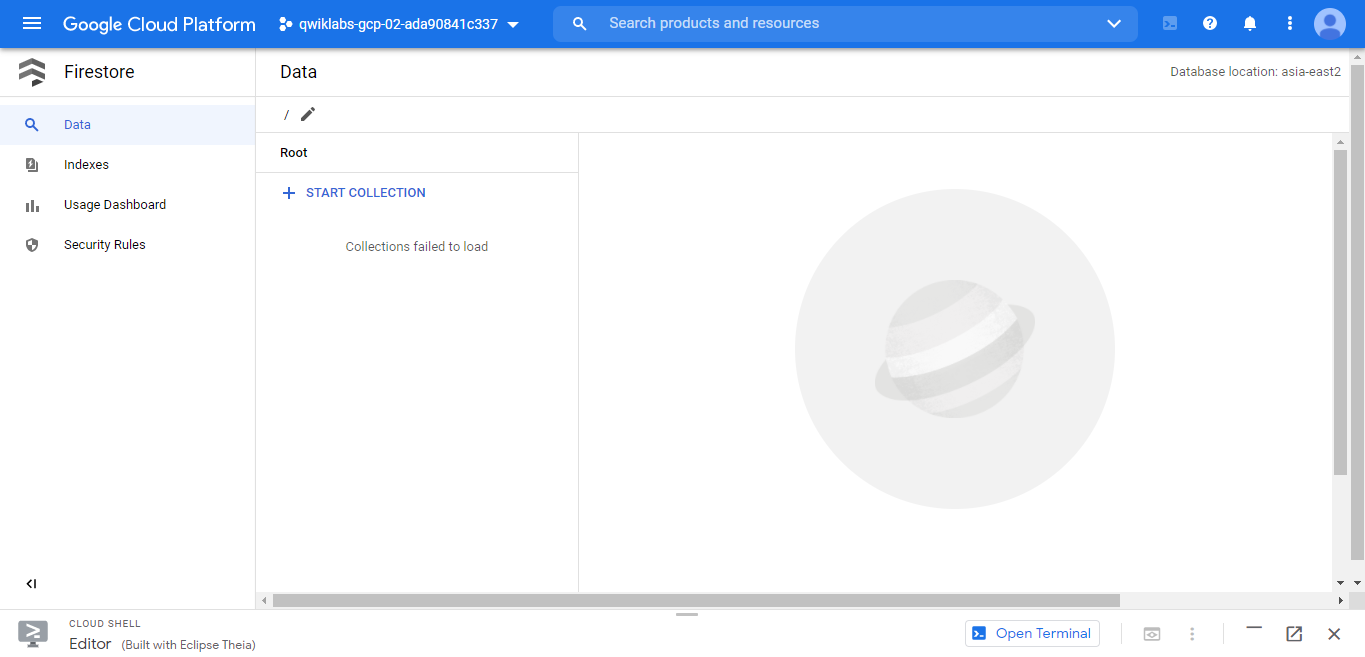


**Deploying app**

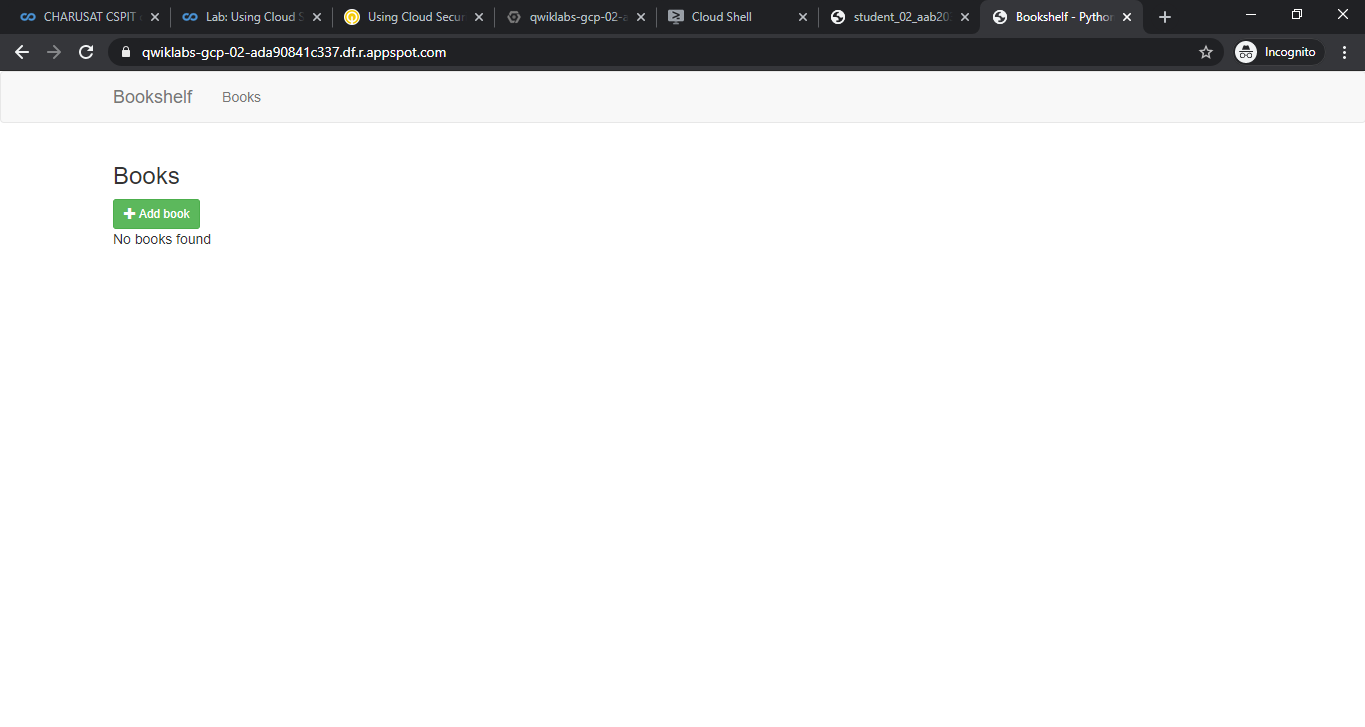


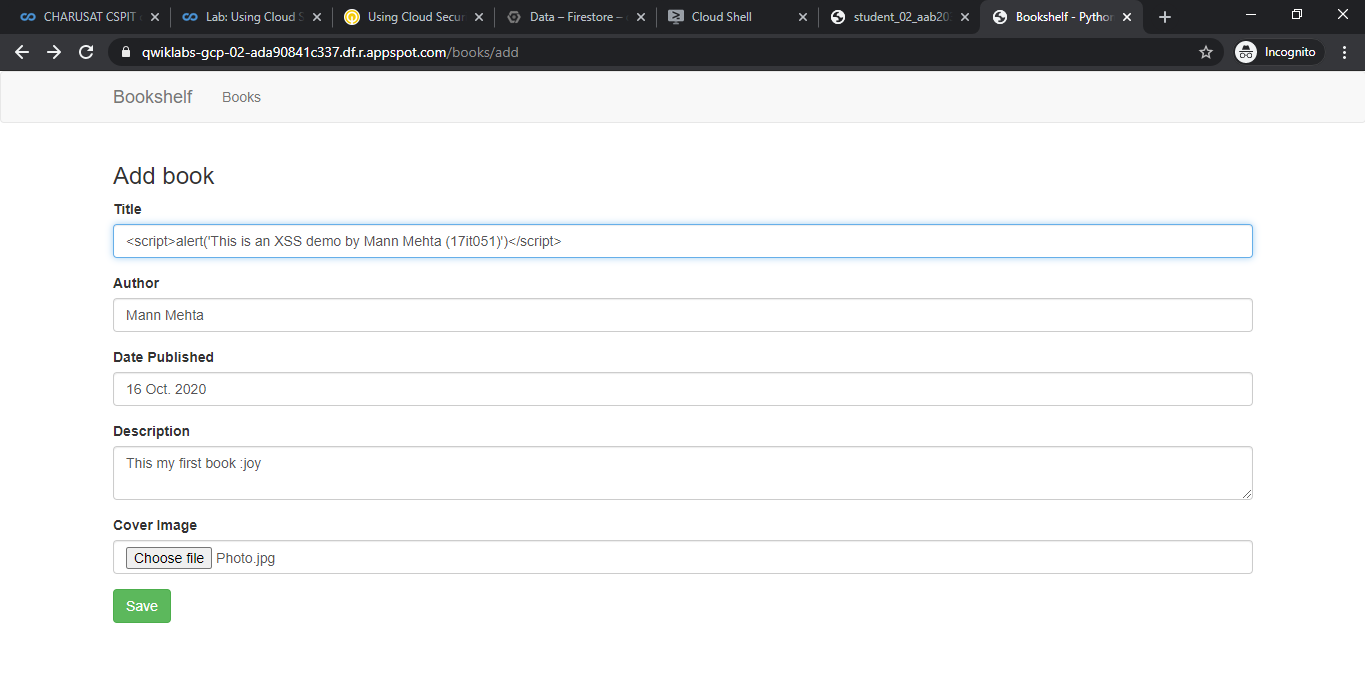
**Deploying the app after adding database**



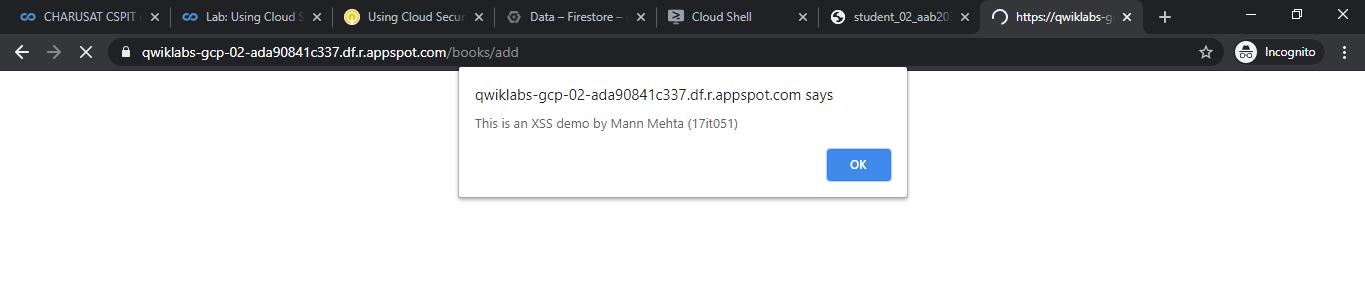


**Swithcing to cloud firestore store native mode.**

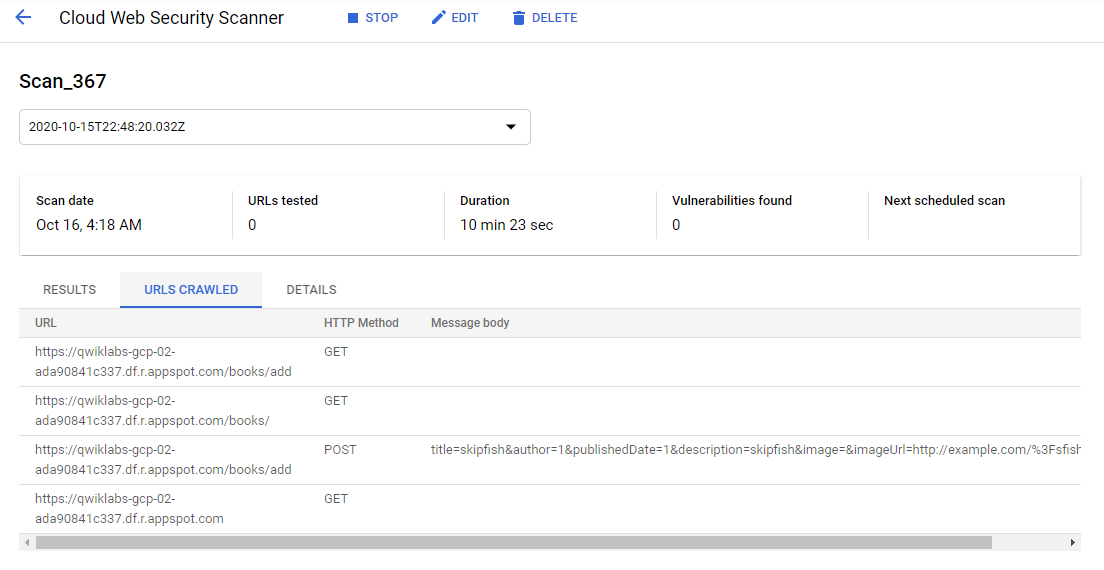




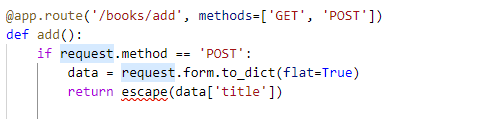
**App Engine application deployed successfully and been able to view it**



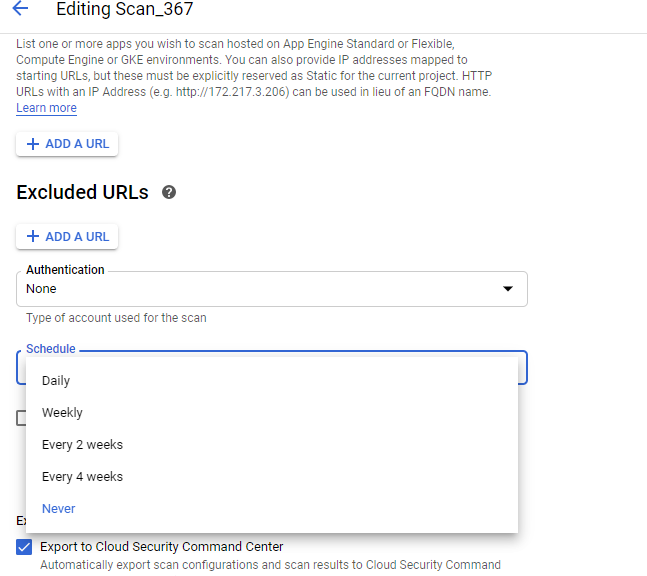
**Site is vulnerable to XSS**



**Observing vulnerabilities and crawled urls**

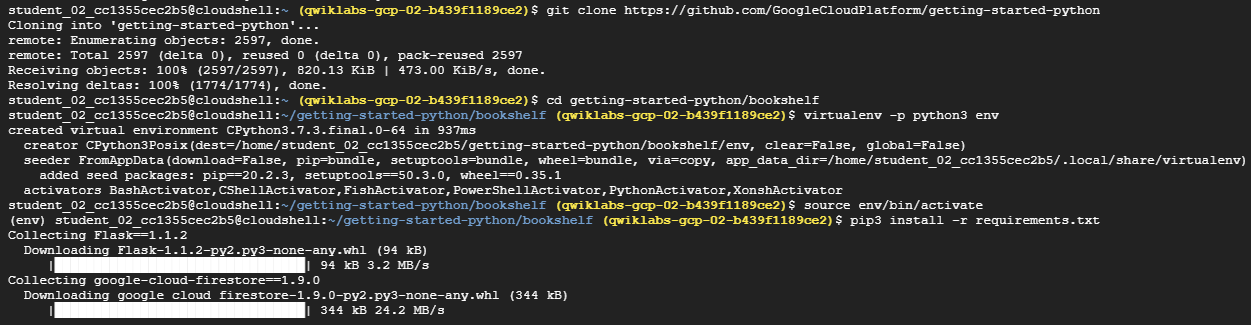


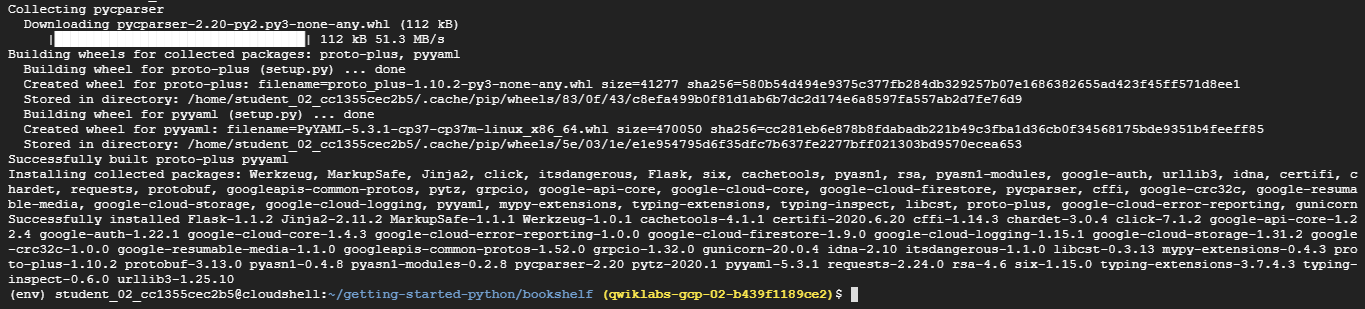
**Correcting the vulnerability in the application**



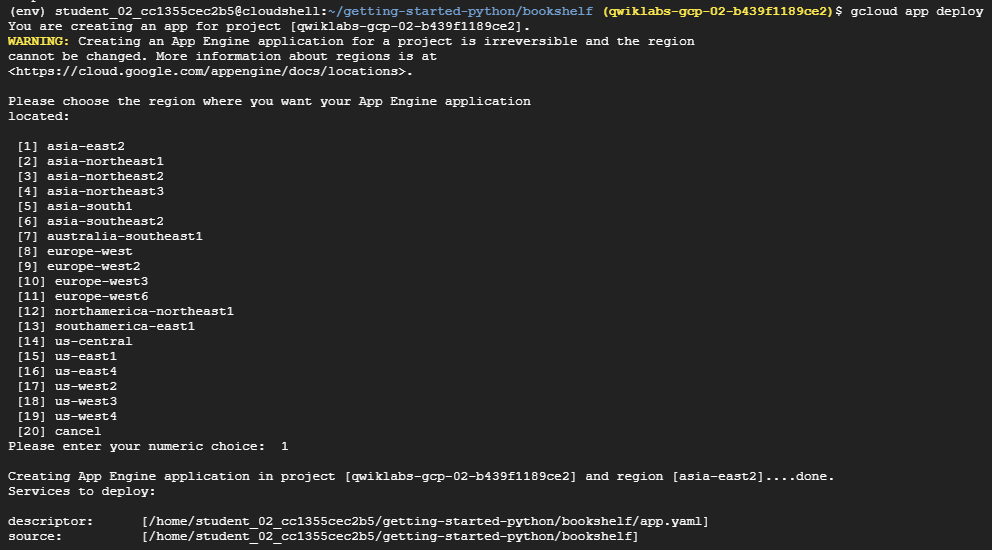
**Scheduling the scans**

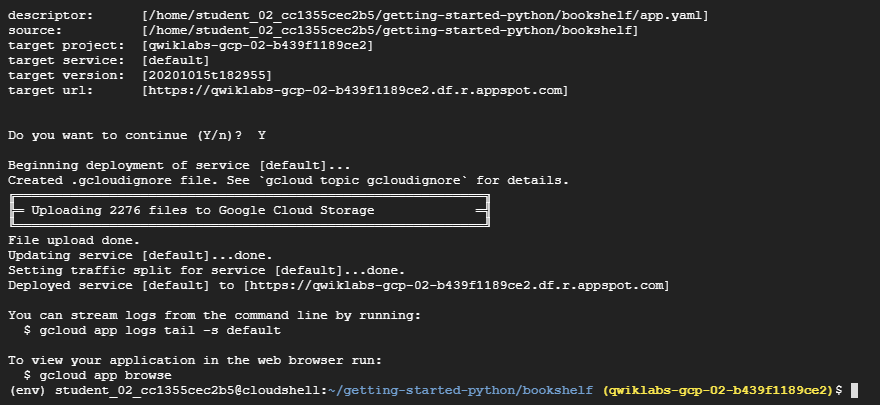
**8.4 Configuring Identity Aware Proxy to Protect a Project**



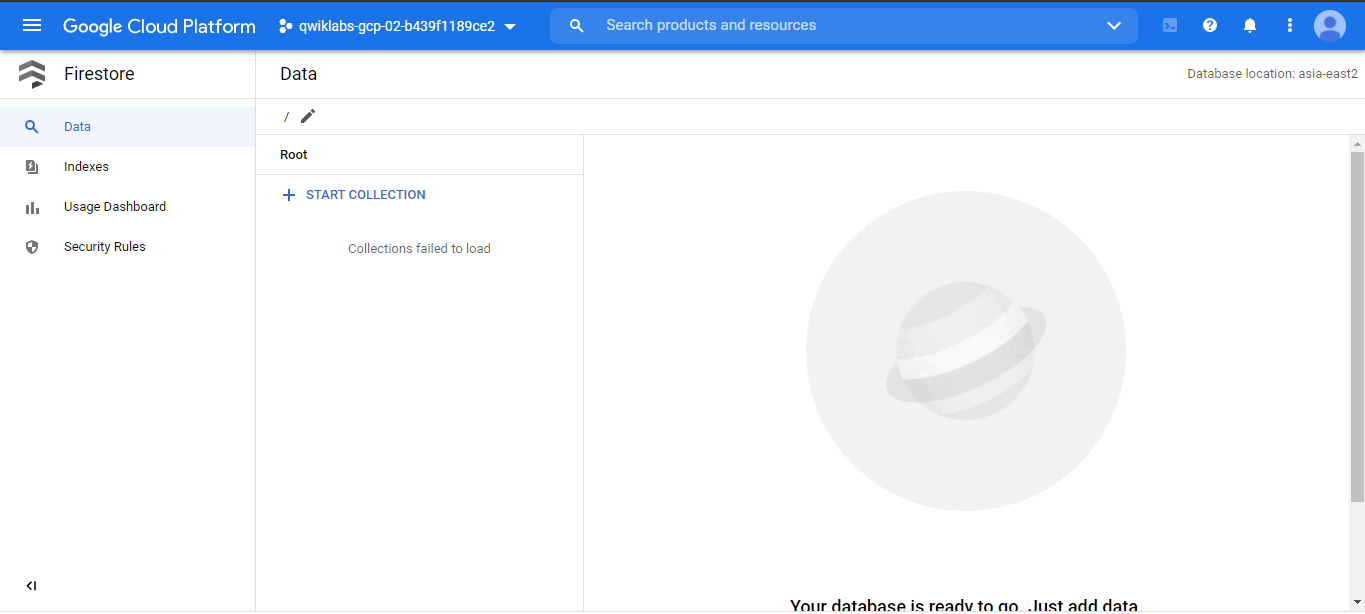


**Cloning the application and installing the requirements**

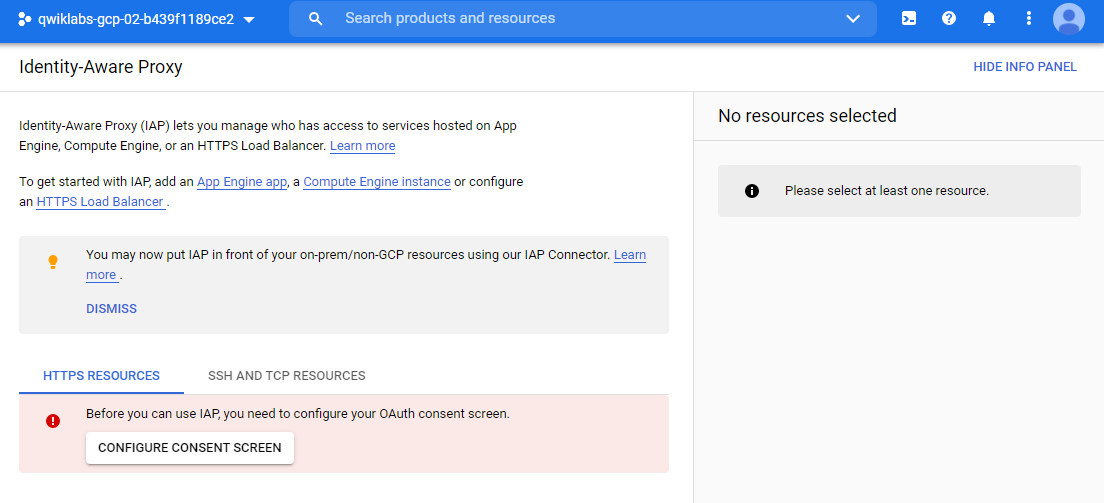


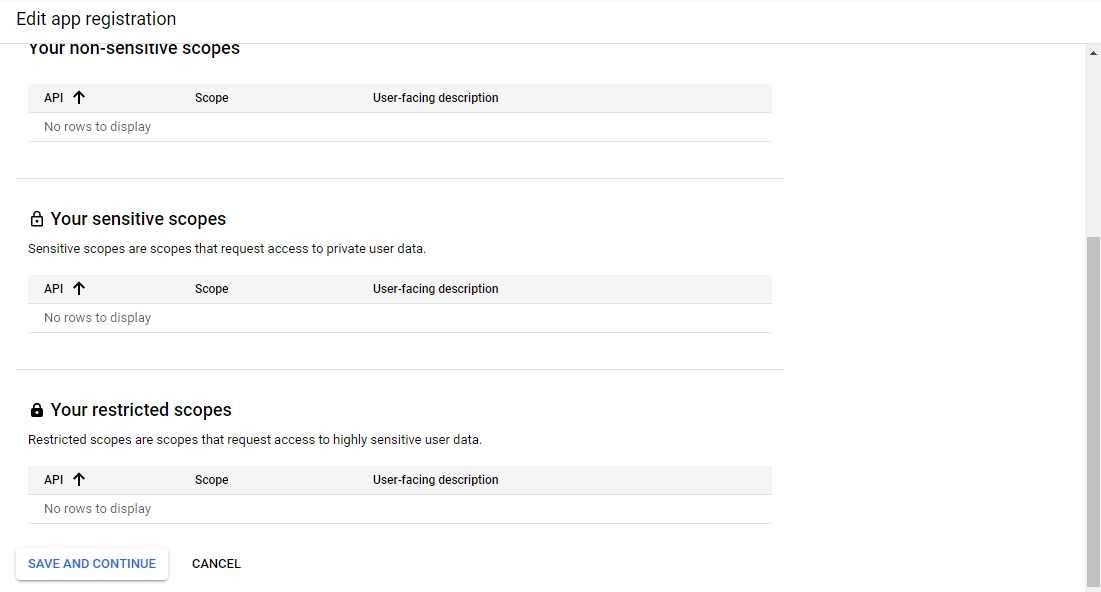


**Deploying the application**

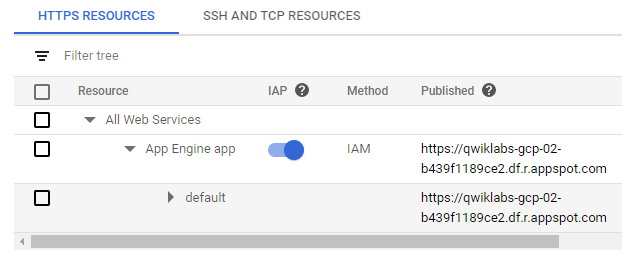


**Firestore setting up**

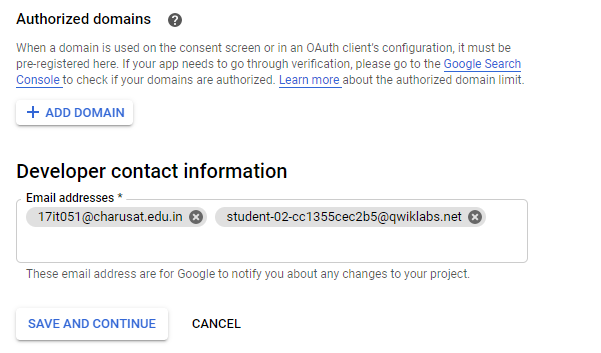




**Setting up the OAuth Consent screen**

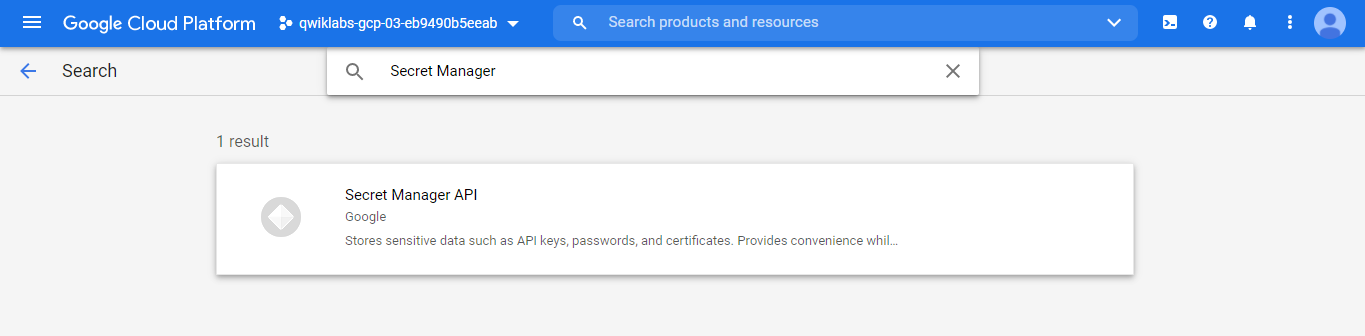


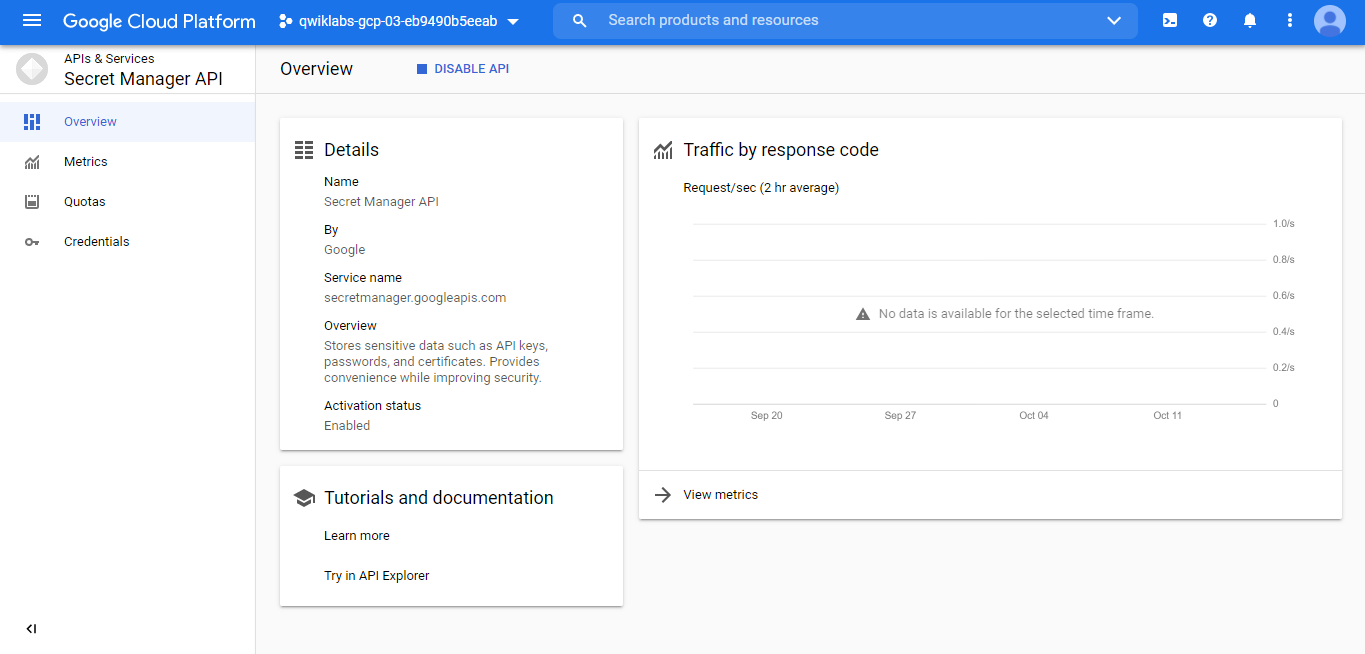
**Configured the OAuth Consent Screen**



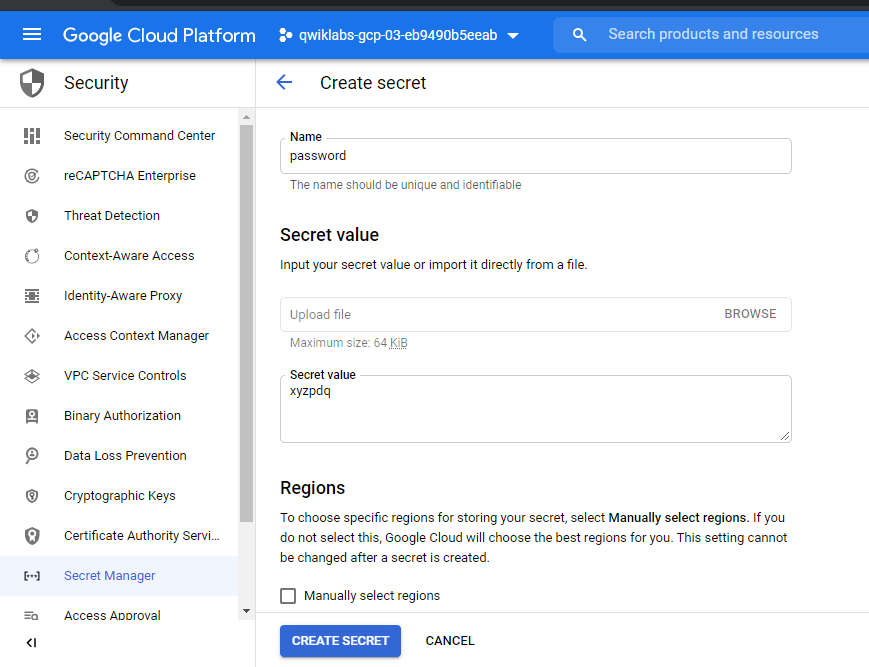
**Adding only admin/specific members to access the application**

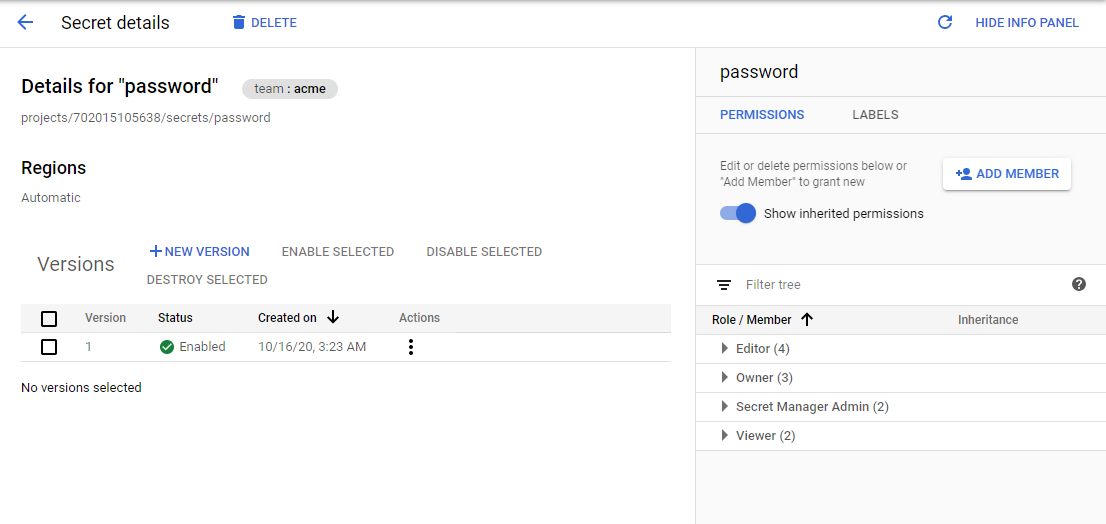
**8.5 Configuring and Using Credentials with Secret Manager**





**Enabling Secret Manager API**

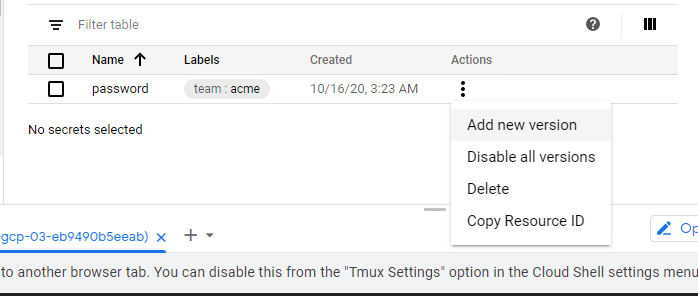


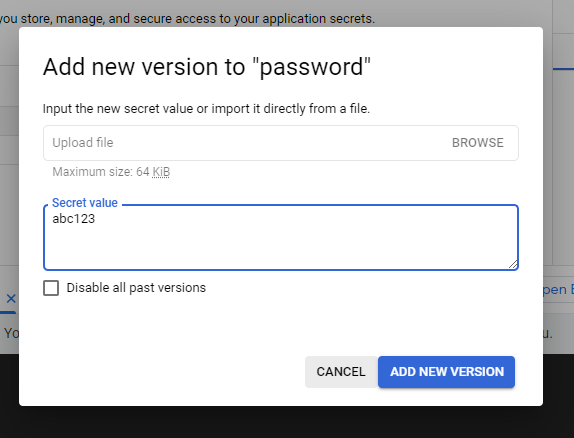


**Creating new secret with name “password”**



**Accessing secret with name “password” from shell.**

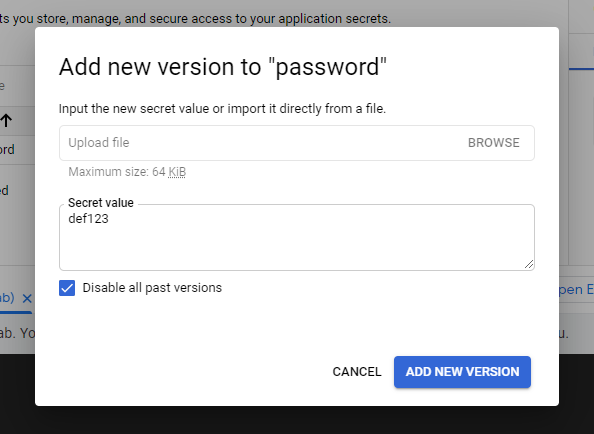




**Adding New Secret Version**



**Checking for new version**



**Adding new version and along with checking disable past versions**

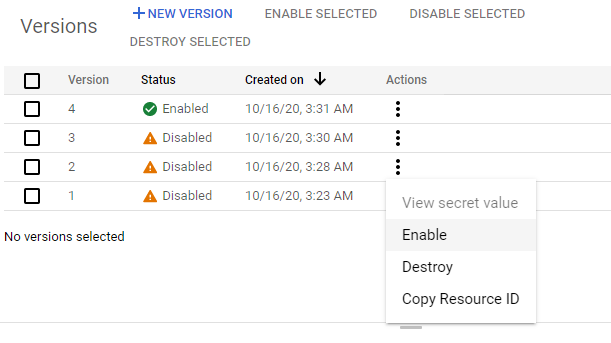


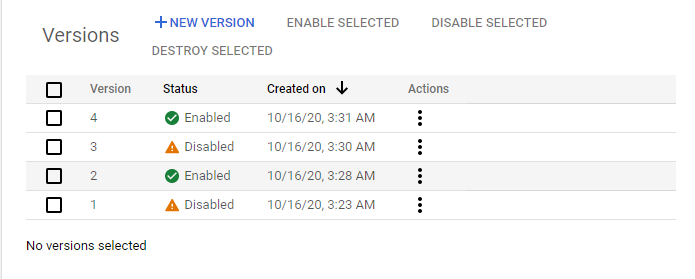


**Accessing the new version and also trying to access old version and getting errors**



**Accessing version 2**





**Old version 2 enabled again**

**LATEST APPLICATIONS:**

Providing security controls and techniques on real time applications and also managing timely updating and rotating the keys for accessing complete benefits of gcp security.

**LEARNING OUTCOME:**

Configure CSEK for google cloud storage, uploading files on storage bucket and generate and rotate the encryption keys then generate manage and encrypted data using cloud kms, create key rings and crypto keys then install sample app engine and sue cloud security scanner to scan application and find vulnerabilities along with scheduling the scans then configure OAuth consent screen and setup cloud IAP access and enable cloud API.

**REFERENCE:**

1. https://googlecoursera.qwiklabs.com/focuses/11115438?parent=lti\_session
2. https://googlecoursera.qwiklabs.com/focuses/11115450?parent=lti\_session
3. https://googlecoursera.qwiklabs.com/focuses/11115455?parent=lti\_session
4. https://googlecoursera.qwiklabs.com/focuses/11115459?parent=lti\_session
5. https://googlecoursera.qwiklabs.com/focuses/11115466?parent=lti\_session