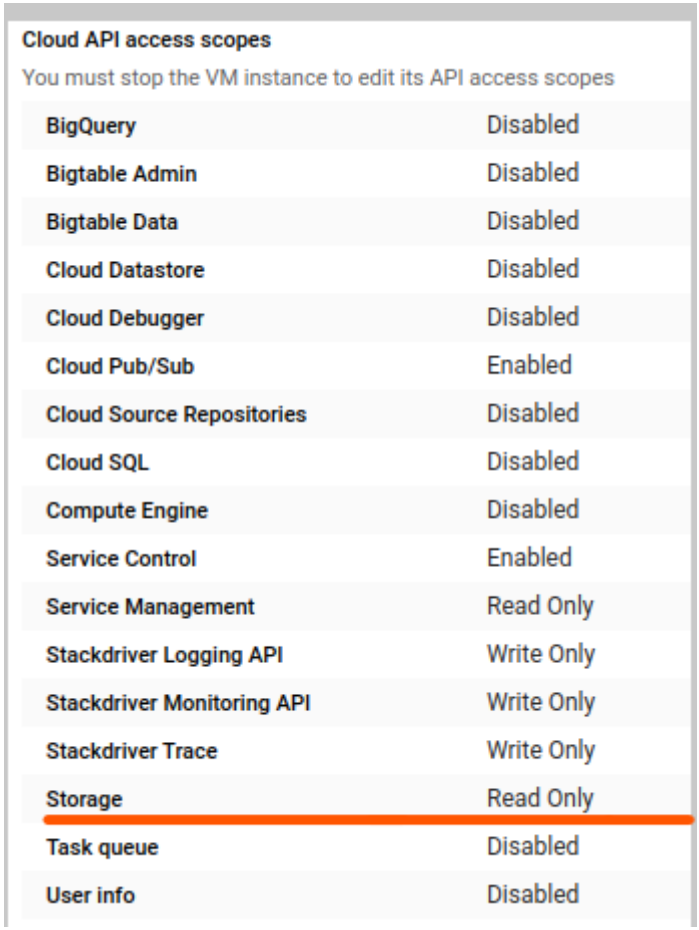
1. **Klient poprosił cię o przygotowanie maszyny dla swoich pracowników, którzy będą mogli pobierać faktury z przygotowanego repozytorium (w naszym przypadku jest to pojemnik Cloud Storage)**

   Domyślnie instancja ma uprawnienia tylko do odczytu Storage:



Cloud API access scopes	
You must stop the VM instance to edit its API access scopes	
BigQuery	Disabled
Bigtable Admin	Disabled
Bigtable Data	Disabled
Cloud Datastore	Disabled
Cloud Debugger	Disabled
Cloud Pub/Sub	Enabled
Cloud Source Repositories	Disabled
Cloud SQL	Disabled
Compute Engine	Disabled
Service Control	Enabled
Service Management	Read Only
Stackdriver Logging API	Write Only
Stackdriver Monitoring API	Write Only
Stackdriver Trace	Write Only
Storage	Read Only
Task queue	Disabled
User info	Disabled

a) Przygotowanie środowiska:
   # Utworzenie maszyny
   gcloud compute instances create vm01 --machine-type=f1-micro --preemptible --zone=europe-west1-b

   # Utworzenie bucket
   gsutil mb gs://gcp-bucket-rk01

   # Stworzenie i skopiowanie przykdowych plikow do bucket
   gsutil cp * gs://gcp-bucket-rk01

b) Operacje:
   # Wyświetlenie zawartości bucket
   gsutil ls gs://gcp-bucket-rk01

```
robert@vm01:~$ gsutil ls  gs://gcp-bucket-rk01
gs://gcp-bucket-rk01/faktura1
gs://gcp-bucket-rk01/faktura2
gs://gcp-bucket-rk01/faktura3
gs://gcp-bucket-rk01/faktura4
gs://gcp-bucket-rk01/faktura5
```

Wyświetlenie zawartości pliku
gsutil cat gs://gcp-bucket-rk01/faktura2

```
robert@vm01:~$ gsutil cat  gs://gcp-bucket-rk01/faktura2
faktura2
```

Usunięcie pliku
gsutil rm gs://gcp-bucket-rk01/faktura1

```
robert@vm01:~$ gsutil rm  gs://gcp-bucket-rk01/faktura1
Removing gs://gcp-bucket-rk01/faktura1...
AccessDeniedException: 403 Insufficient Permission
```

2. Dany klient przetrzymuje bardzo ważne dokumenty. Zarząd zdecydował, że wprowadzą szyfrowanie krytycznych dokumentów, które będą mogły zostać odszyfrowane po stronie pracownika, który z danego dokumentu chce skorzystać.

a) Utworzenie nowego service account na potrzeby vm:
gcloud iam service-accounts create crypto-bucket --display-name=crypto-bucket

b) Przypisanie roli do crypto-bucket umożliwiającej szyfowanie, deszyfrowane i odstęp do klucz publicznego. Uprawnień operacji na bucket.
 # role-crypt.yaml
title: "Crypto Role"
description: "Dostep do kluczy szyfrowania"
stage: GA
includedPermissions:
- cloudkms.cryptoKeyVersions.useToDecrypt
- cloudkms.cryptoKeyVersions.useToEncrypt
- cloudkms.cryptoKeyVersions.viewPublicKey

 # Utworzenie custom role
gcloud iam roles create gerwazy --project szkola-chmury-agcp --file role-crypt.yaml
 # bind roli Gerwazy do service account
gcloud projects add-iam-policy-binding szkola-chmury-agcp \
  --member serviceAccount:crypto-bucket@szkola-chmury-agcp.iam.gserviceaccount.com \
  --role projects/szkola-chmury-agcp/roles/gerwazy
 # nadanie uprawnień SA dop operacji na bucket
gsutil iam ch serviceAccount:crypto-bucket@szkola-chmury-agcp.iam.gserviceaccount.com:roles/storage.legacyBucketWriter gs://gcp-bucket-rk01/

c) Utworzenie nowych maszyn pod kontrolą crypto-bucket
#vm01|vm02
gcloud compute instances create vm01 \
  --machine-type=f1-micro --preemptible --zone=europe-west1-b \

--service-account=crypto-bucket@szkola-chmury-agcp.iam.gserviceaccount.com \
--scopes storage-rw,https://www.googleapis.com/auth/cloudkms

d) Instalacja gcsfuse
export GCSFUSE_REPO=gcsfuse-`lsb_release -c -s`

echo "deb http://packages.cloud.google.com/apt $GCSFUSE_REPO main" | sudo tee /etc/apt/sources.list.d/gcsfuse.list
curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

sudo apt-get update
sudo apt-get install gcsfuse

```
robert@vm01:~$ mkdir sejf
robert@vm01:~$ gcsfuse gcp-bucket-rk01  sejf
Using mount point: /home/robert/sejf
Opening GCS connection...
Opening bucket...
Mounting file system...
File system has been successfully mounted.
robert@vm01:~$ cd sejf/
robert@vm01:~/sejf$ ls
crypt.md  faktura1  faktura2  faktura3  faktura4  faktura5
robert@vm01:~/sejf$
```

e) Wygenerowanie kluczy szyfrowania.
 # Utwórz keyring
gcloud kms keyrings create secure-doc --location global

 # utwórz parę kluczy asymetrycznych
gcloud kms keys create key01-doc \
--location global \
--keyring secure-doc \
--purpose asymmetric-encryption \
--default-algorithm rsa-decrypt-oaep-2048-sha256 \
--protection-level software

 # klucz szyfrowania symetrycznego
gcloud kms keys create key02-sym-doc \
--location global \
--keyring secure-doc \
--purpose encryption \
--rotation-period=30d \
--next-rotation-time=2020-01-22T20:00:00.000Z

f) Szyfrowanie asymetryczne
 # pobranie klucza publicznego na vm01
gcloud kms keys versions \
get-public-key 1 \
--location global \
--keyring secure-doc \

--key key01-doc \
--output-file ~/key01-doc.pub

 # wygenerowanie pliku do szyfrowania na vm01

echo "Tajna wiadomosc" > crypt-mini.md

```
robert@vm01:~/sejf$ gcloud kms keys versions \
>   get-public-key 1 \
>   --location global \
>   --keyring secure-doc \
>   --key key01-doc \
>   --output-file ~/key01-doc.pub
robert@vm01:~/sejf$ echo "Tajna wiadomosc" > crypt-mini.md
robert@vm01:~/sejf$ cat crypt-mini.md
Tajna wiadomosc
robert@vm01:~/sejf$ cat ~/key01-doc.pub
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4PigczNwuDNjeU5x9OGl
HstYDuFV92aMQ/s96gIVM3z4u3FoHvstUjzxPcz1Z6TREvYo8iCsWoOS6pHGScIQ
4Wy3i5qIbL1HMOG3Upon9z9AAfE6nQHYC/dWAkO9KVykSHHwEGo0xNd9nKMCwVuT
S90n4jCXZ0+mSei6zUziKewJt3DDXEKMiNd8vE1oHQ/5ctom4ZWFhlI/4XpdITKx
Wa7KcWlj09GVk6K6poCLJboCUY1L15ReF4stl3ChGCa1gRRZDMkk0IYR75Z5IwHx
EwltSBwNXtQBryz+ROflSK0aV+y0Wbk/dDvAY7pAWqTINTrGsIpvEav+tzNSQyWi
fQIDAQAB
-----END PUBLIC KEY-----
```

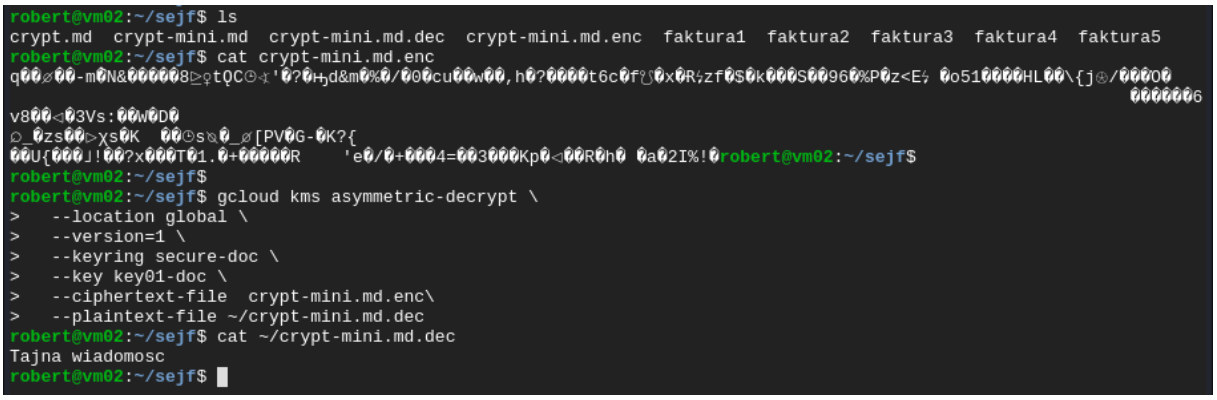
 # zaszyfrowanie wiadomości na vm01

openssl pkeyutl -in crypt-mini.md \
-encrypt -pubin \
-inkey ~/key01-doc.pub \
-pkeyopt rsa_padding_mode:oaep \
-pkeyopt rsa_oaep_md:sha256 \
-pkeyopt rsa_mgf1_md:sha256 > crypt-mini.md.enc

```
robert@vm01:~/sejf$ openssl pkeyutl -in crypt-mini.md \
>   -encrypt -pubin \
>   -inkey ~/key01-doc.pub \
>   -pkeyopt rsa_padding_mode:oaep \
>   -pkeyopt rsa_oaep_md:sha256 \
>   -pkeyopt rsa_mgf1_md:sha256 > crypt-mini.md.enc
robert@vm01:~/sejf$ ls
crypt.md  crypt-mini.md  crypt-mini.md.enc  faktura1  faktura2  faktura3  faktura4  faktura5
robert@vm01:~/sejf$ cat crypt-mini.md.enc
q��⌀��-m@N&@@@@@8⊵♀tQC◷↯'@?@ʰ♄d&m@%@/@@cu@@w@@,h@?@@@@t6c@f♌@x@R↯zf@S@k@@@S@@96@%P@z<E↯ @o51@@@@HL@@\{j⊙/@@@O@
                                                                                                                    @@@@@@6
v8@@◁@3Vs:@@w@D@
⌕_@zs@@▷χs@K  @@◷s⊗@_⌀[PV@G-@K?{
@@U{@@@J!@@?x@@@T@1.@+@@@@@R    'e@/@+@@@4=@@3@@@Kp@◁@@R@h@ @a@2I%!@robert@vm01:~/sejf$
```

 # odszyfrowanie wiadomości na drugiej maszynie

gcloud kms asymmetric-decrypt \
--location global \
--version=1 \
--keyring secure-doc \
--key key01-doc \
--ciphertext-file crypt-mini.md.enc\
--plaintext-file ~/crypt-mini.md.dec

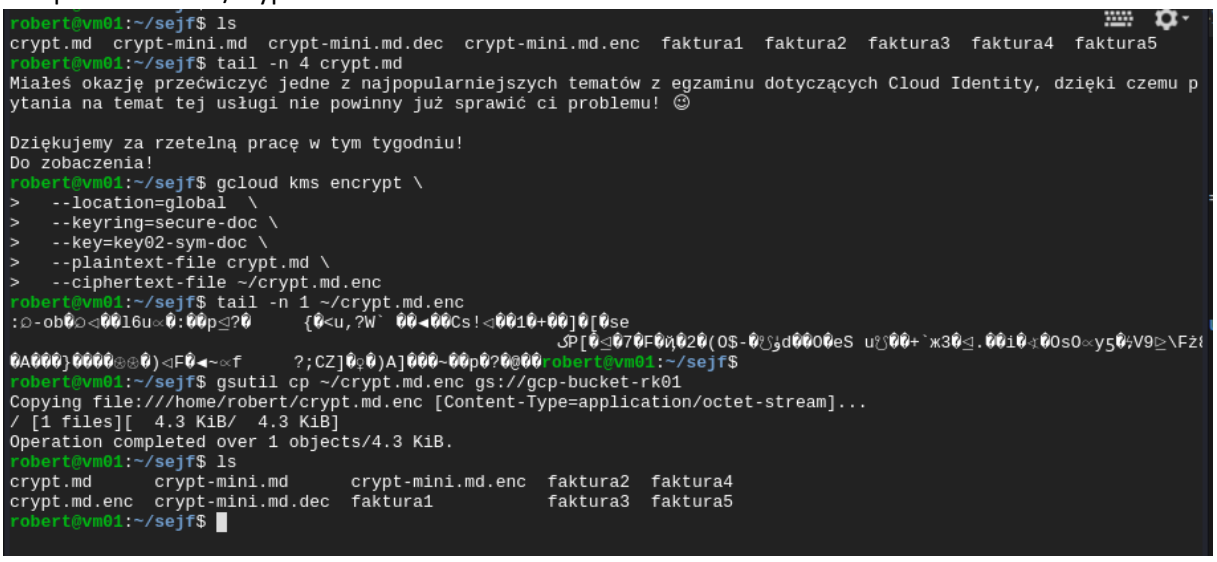


#uwagi: gcsfuze nie radzi sobie z formatem plików wynikowych.

g) Szyfrowanie kluczem symetrycznym

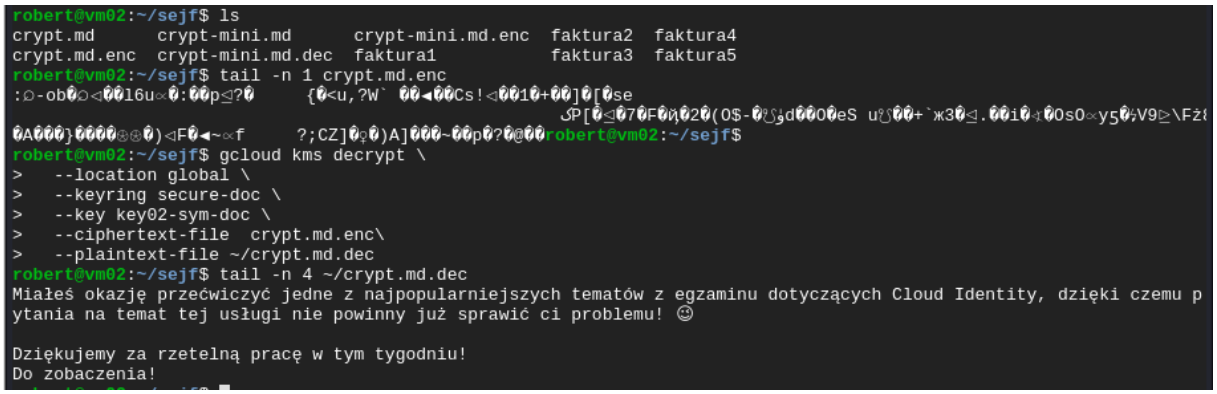
 # zaszyfrowanie wiadomości na maszynie vm01

```
gcloud kms encrypt \
 --location=global  \
 --keyring=secure-doc \
 --key=key02-sym-doc \
 --plaintext-file crypt.md \
 --ciphertext-file ~/crypt.md.enc
```



odszyfrowanie wiadomości na maszynie vm02

```
gcloud kms decrypt \
 --location global \
 --keyring secure-doc \
 --key key02-sym-doc \
 --ciphertext-file  crypt.md.enc\
 --plaintext-file ~/crypt.md.dec
```



#podsumowanie: gcsfuse sprawdza się jako dostawca plików do wymiany jednak nie radzi sobie z operacjami z wygenerowanym formatem pliku.

3. **Firma zdecydowała się już na ostatni krok ... zbudowanie niestandardowej roli za pomocą, której połączą możliwości szyfrowania oraz odszyfrowywania danych za pomocą KMS oraz dostępu do danych w Cloud Storage na poziomie READ.**

a) Utworzenie i przypisanie roli.

 # role-crypto-cs.yaml
title: "Crypto and CS ro Role"
description: "Dostep do kluczy szyfrowania oraz odczytu CloudStorage"
stage: GA
includedPermissions:
- cloudkms.cryptoKeyVersions.useToDecrypt
- cloudkms.cryptoKeyVersions.useToEncrypt
- cloudkms.cryptoKeyVersions.viewPublicKey
- storage.objects.get
- storage.objects.list

 # utworzenie roli
gcloud iam roles create cryptoCSviewer --project szkola-chmury-agcp --file role-crypto-cs.yaml

 # przypisanie roli do użytkownika
gcloud projects add-iam-policy-binding szkola-chmury-agcp \
--member user:testuser@overseer.eu \
--role projects/szkola-chmury-agcp/roles/cryptoCSviewer

b) Sprawdzenie uprawnień.

 # utworzenie maszyny bez scope
gcloud compute instances create vm03 \
--machine-type=f1-micro --preemptible --zone=europe-west1-b \
--no-scopes
 # próba pobrania klucza publicznego
gcloud kms keys versions get-public-key 1 --location global --keyring secure-doc --key key01-doc --output-file ~/keyy01-doc.pub


testuser@vm03:~$ gcloud kms keys versions get-public-key 1 --location global --keyring secure-doc --key key01-doc --output-file ~/keyy01-doc.pub
testuser@vm03:~$ ls
keyy01-doc.pub
testuser@vm03:~$ cat keyy01-doc.pub
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4PigczNwuDNjeU5x9OGl
HstYDuFV92aMQ/s96gIVM3z4u3FoHvstUjzxPcz1Z6TREvYo8iCsWoOS6pHGScIQ
4Wy3i5qIbL1HMOG3Upon9z9AAfE6nQHYC/dWAkO9KVykSHHwEGo0xNd9nKMCwVuT
S90n4jCXZ0+mSei6zUziKewJt3DDXEKMiNd8vE1oHQ/5ctom4ZWFhlI/4XpdITKx
Wa7KcWlj09GVk6K6poCLJboCUY1L15ReF4stl3ChGCa1gRRZDMkk0IYR75Z5IwHx
EwltSBwNXtQBryz+ROflSK0aV+y0Wbk/dDvAY7pAWqTINTrGsIpvEav+tzNSQyWi
fQIDAQAB
-----END PUBLIC KEY-----
testuser@vm03:~$