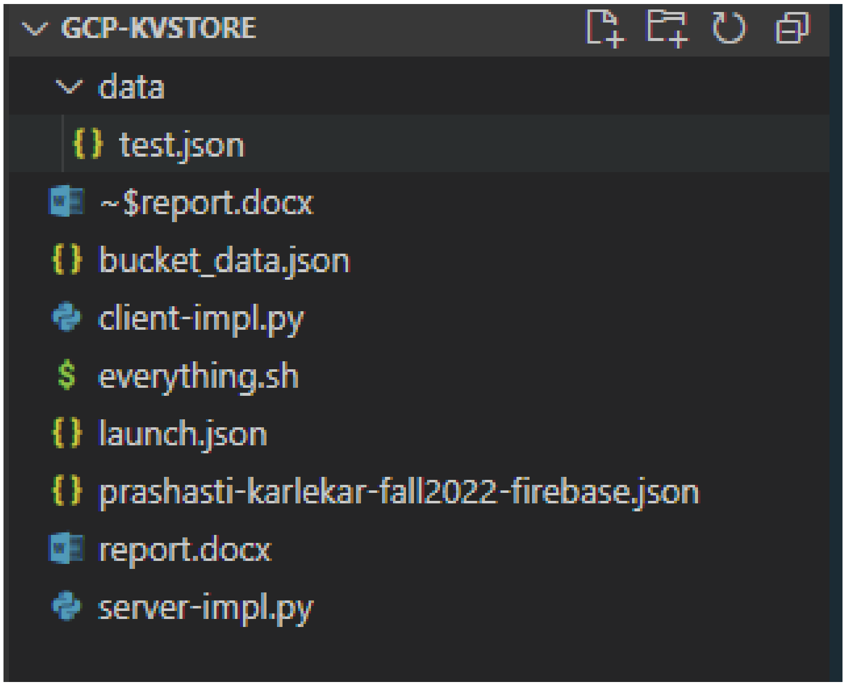
ASSIGNMENT 3

CLOUD KV STORE

ARCHITECTURE:



```
GCP-KVSTORE
  data
    test.json
  ~$report.docx
  bucket_data.json
  client-impl.py
  everything.sh
  launch.json
  prashasti-karlekar-fall2022-firebase.json
  report.docx
  server-impl.py
```

Figure 1 – Folder Structure

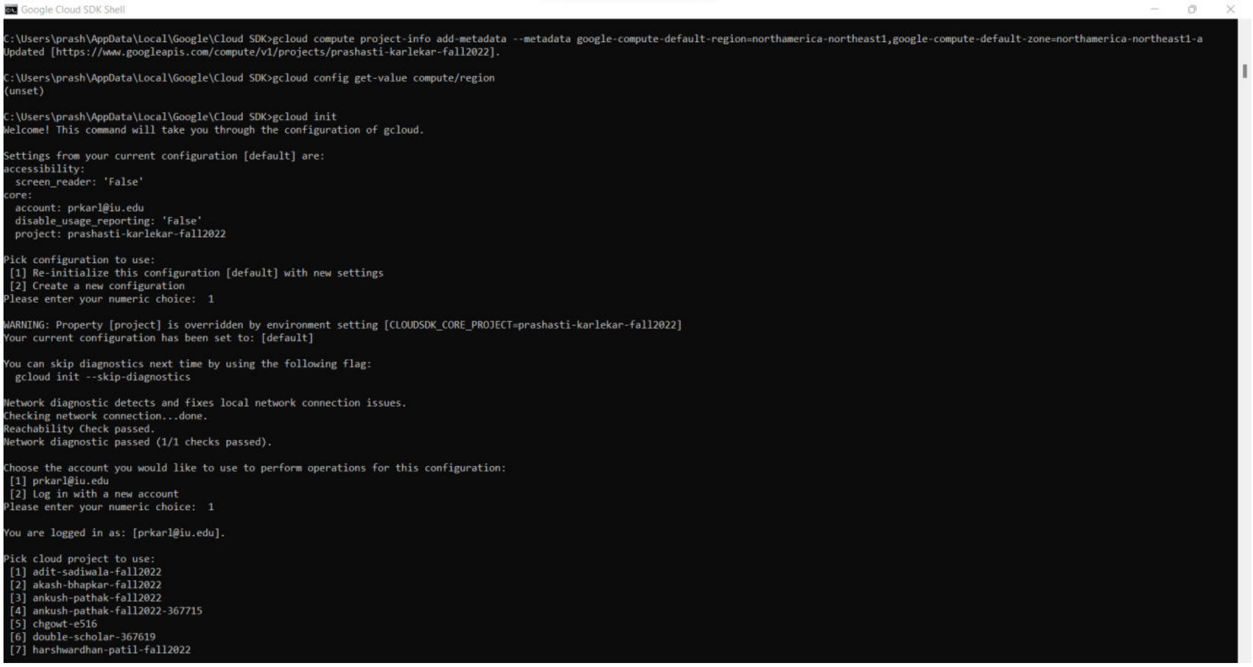
- 1 TCP/IP server built using python3 which interacts with clients and performs the operations
- 1 Database, which is a collection of files (1 file per key) stored in the 'data' folder by the server for native storage
- Client process that initiates set and get requests to the server
- Everything.sh is the script file to run the entire workflow.
- prashasti-karlekar-fall2022-firebase.json contains public/private credentials to connect to the Firestore for my project used for this assignment
- bucket_data.json is a dummy starter JSON file used for bucket storage on google cloud

OPERTIONS PERFORMED:

1. The project "prashasti-karlekar-fall2022" was created with region set to northamerica-northeast1 and zone set to northamerica-northeast1-a and SSH keys were generated using ssh-keygen command, followed by gcloud init to save the changes performed.

ASSIGNMENT 3

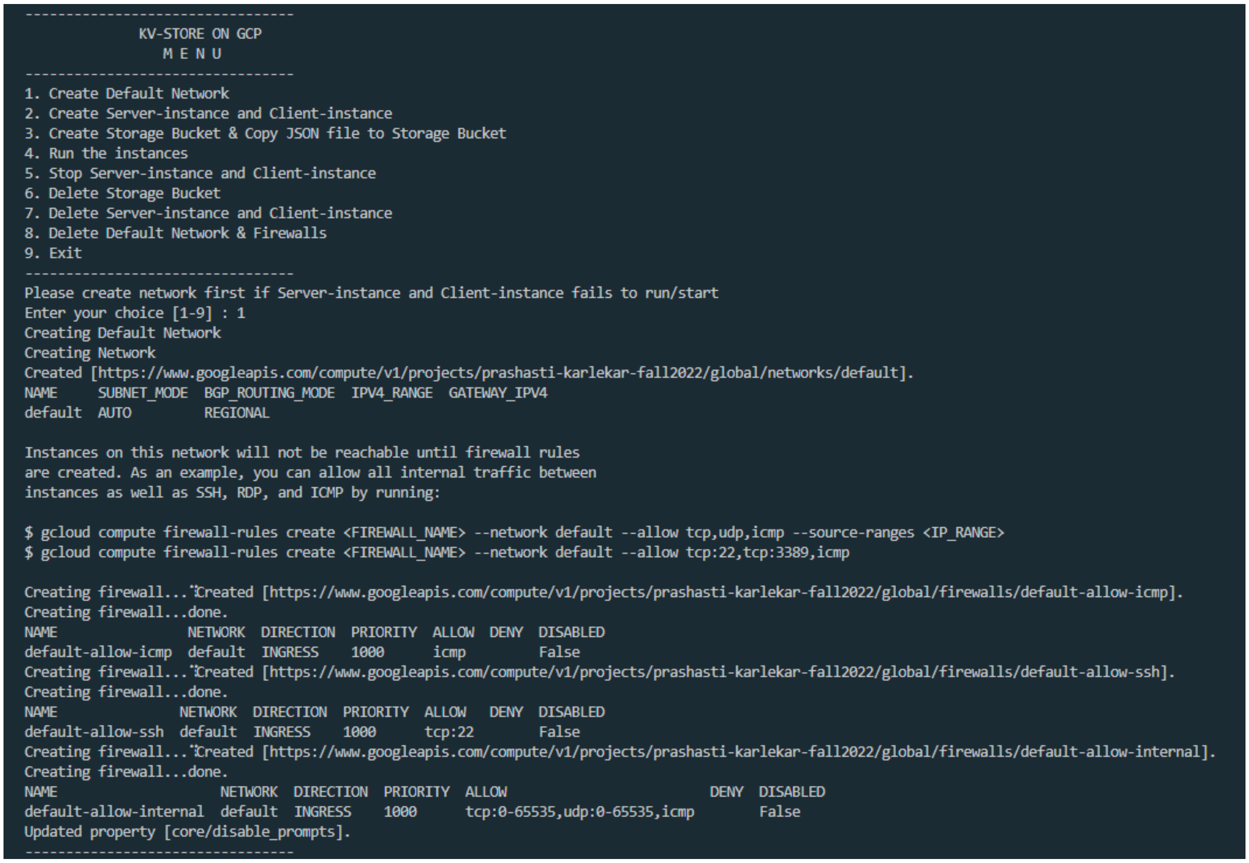
CLOUD KV STORE



Google Cloud SDK Shell

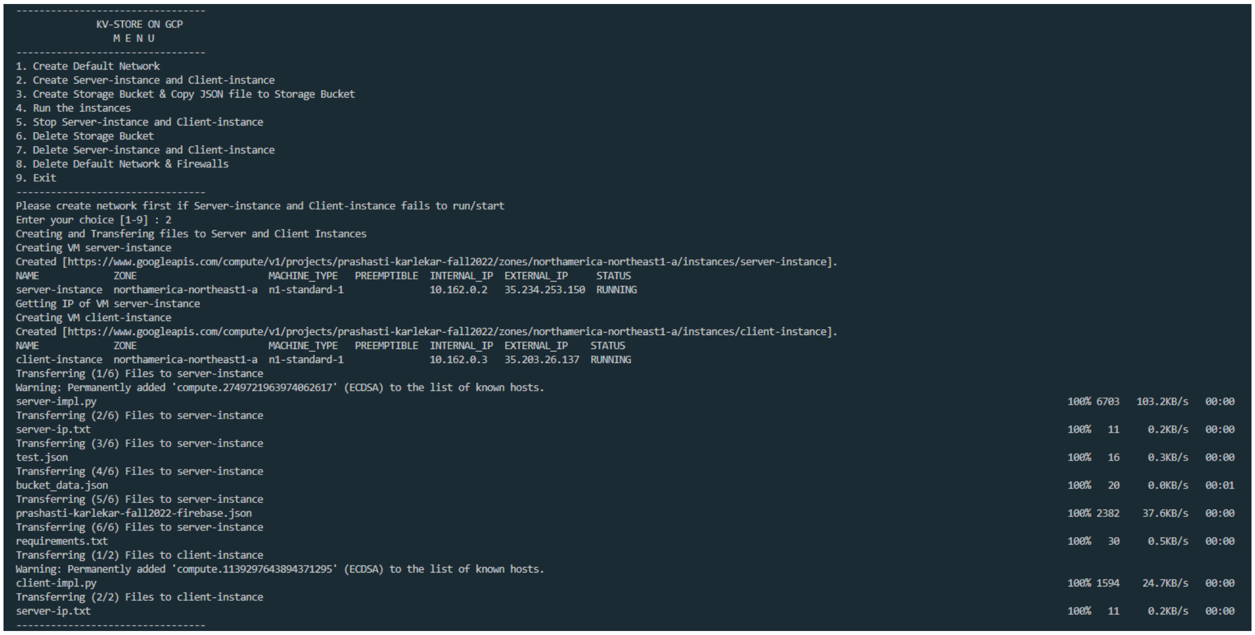
```
C:\Users\prash\AppData\Local\Google\Cloud SDK>gcloud compute project-info add-metadata --metadata google-compute-default-region=northamerica-northeast1,google-compute-default-zone=northamerica-northeast1-a
Updated [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022].

C:\Users\prash\AppData\Local\Google\Cloud SDK>gcloud config get-value compute/region
(unset)

C:\Users\prash\AppData\Local\Google\Cloud SDK>gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
accessibility:
  screen_reader: 'False'
core:
  account: prkarl@iu.edu
  disable_usage_reporting: 'False'
  project: prashasti-karlekar-fall2022

Pick configuration to use:
 [1] Re-initialize this configuration [default] with new settings
 [2] Create a new configuration
Please enter your numeric choice:  1

WARNING: Property [project] is overridden by environment setting [CLOUDSDK_CORE_PROJECT=prashasti-karlekar-fall2022]
Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

Choose the account you would like to use to perform operations for this configuration:
 [1] prkarl@iu.edu
 [2] Log in with a new account
Please enter your numeric choice:  1

You are logged in as: [prkarl@iu.edu].

Pick cloud project to use:
 [1] adit-sadiwala-fall2022
 [2] akash-bhapkar-fall2022
 [3] ankush-pathak-fall2022
 [4] ankush-pathak-fall2022-367715
 [5] chgowt-e516
 [6] double-scholar-367619
 [7] harshwardhan-patil-fall2022
```

2. Default network was created, followed by creating specific firewall rules.



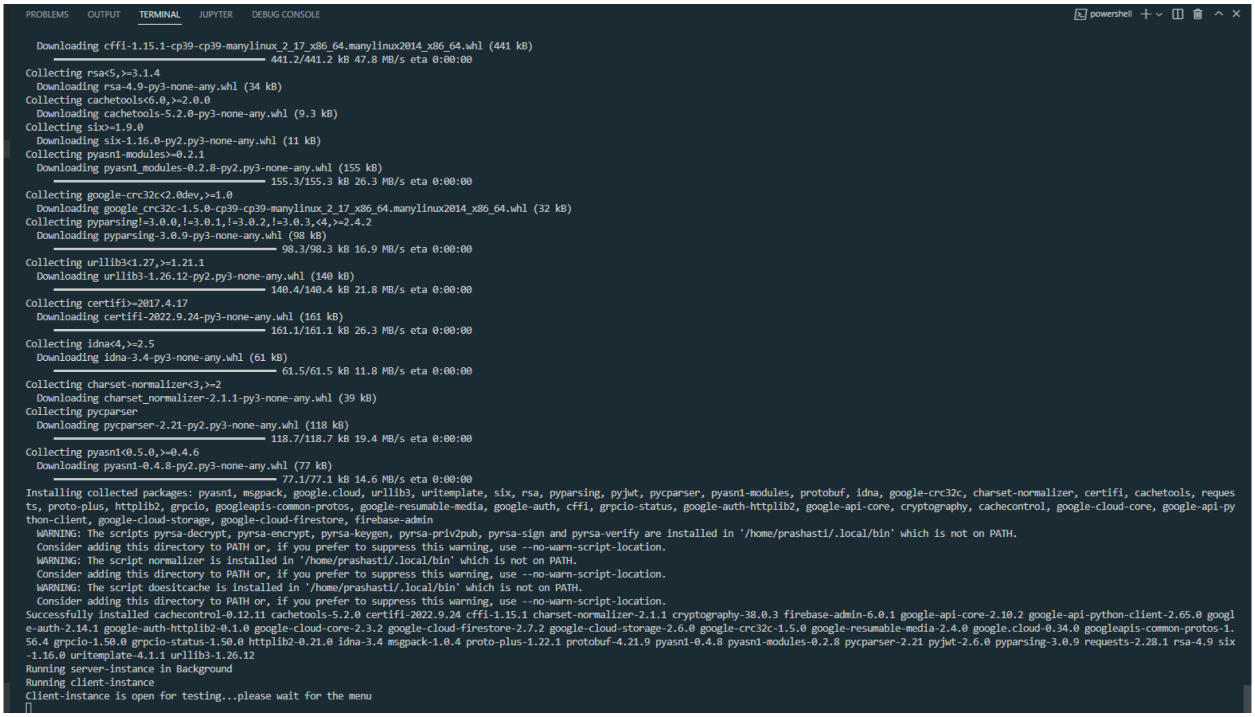
```
--------------------------------
             KV-STORE ON GCP
                M E N U
--------------------------------
1. Create Default Network
2. Create Server-instance and Client-instance
3. Create Storage Bucket & Copy JSON file to Storage Bucket
4. Run the instances
5. Stop Server-instance and Client-instance
6. Delete Storage Bucket
7. Delete Server-instance and Client-instance
8. Delete Default Network & Firewalls
9. Exit
--------------------------------
Please create network first if Server-instance and Client-instance fails to run/start
Enter your choice [1-9] : 1
Creating Default Network
Creating Network
Created [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/global/networks/default].
NAME     SUBNET_MODE  BGP_ROUTING_MODE  IPV4_RANGE  GATEWAY_IPV4
default  AUTO         REGIONAL

Instances on this network will not be reachable until firewall rules
are created. As an example, you can allow all internal traffic between
instances as well as SSH, RDP, and ICMP by running:

$ gcloud compute firewall-rules create <FIREWALL_NAME> --network default --allow tcp,udp,icmp --source-ranges <IP_RANGE>
$ gcloud compute firewall-rules create <FIREWALL_NAME> --network default --allow tcp:22,tcp:3389,icmp

Creating firewall...⠹Created [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/global/firewalls/default-allow-icmp].
Creating firewall...done.
NAME               NETWORK  DIRECTION  PRIORITY  ALLOW  DENY  DISABLED
default-allow-icmp  default  INGRESS    1000      icmp        False
Creating firewall...⠹Created [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/global/firewalls/default-allow-ssh].
Creating firewall...done.
NAME              NETWORK  DIRECTION  PRIORITY  ALLOW   DENY  DISABLED
default-allow-ssh  default  INGRESS    1000      tcp:22        False
Creating firewall...⠹Created [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/global/firewalls/default-allow-internal].
Creating firewall...done.
NAME                   NETWORK  DIRECTION  PRIORITY  ALLOW                         DENY  DISABLED
default-allow-internal  default  INGRESS    1000      tcp:0-65535,udp:0-65535,icmp        False
Updated property [core/disable_prompts].
--------------------------------
```

3. Instances : client-instance and server-instance were created and the required files were transferred onto the VMs.



```
--------------------------------
            KV-STORE ON GCP
               M E N U
--------------------------------
1. Create Default Network
2. Create Server-instance and Client-instance
3. Create Storage Bucket & Copy JSON file to Storage Bucket
4. Run the instances
5. Stop Server-instance and Client-instance
6. Delete Storage Bucket
7. Delete Server-instance and Client-instance
8. Delete Default Network & Firewalls
9. Exit
--------------------------------
Please create network first if Server-instance and Client-instance fails to run/start
Enter your choice [1-9] : 2
Creating and Transfering files to Server and Client Instances
Creating VM server-instance
Created [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/zones/northamerica-northeast1-a/instances/server-instance].
NAME             ZONE                        MACHINE_TYPE   PREEMPTIBLE  INTERNAL_IP  EXTERNAL_IP     STATUS
server-instance  northamerica-northeast1-a  n1-standard-1               10.162.0.2   35.234.253.150  RUNNING
Getting IP of VM server-instance
Creating VM client-instance
Created [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/zones/northamerica-northeast1-a/instances/client-instance].
NAME             ZONE                        MACHINE_TYPE   PREEMPTIBLE  INTERNAL_IP  EXTERNAL_IP    STATUS
client-instance  northamerica-northeast1-a  n1-standard-1               10.162.0.3   35.203.26.137  RUNNING
Transferring (1/6) Files to server-instance
Warning: Permanently added 'compute.2749721963974062617' (ECDSA) to the list of known hosts.
server-impl.py                                                                                        100% 6703   103.2KB/s   00:00
Transferring (2/6) Files to server-instance
server-ip.txt                                                                                         100%   11     0.2KB/s   00:00
Transferring (3/6) Files to server-instance
test.json                                                                                             100%   16     0.3KB/s   00:00
Transferring (4/6) Files to server-instance
bucket_data.json                                                                                      100%   20     0.0KB/s   00:01
Transferring (5/6) Files to server-instance
prashasti-karlekar-fall2022-firebase.json                                                             100% 2382    37.6KB/s   00:00
Transferring (6/6) Files to server-instance
requirements.txt                                                                                      100%   30     0.5KB/s   00:00
Transferring (1/2) Files to client-instance
Warning: Permanently added 'compute.1139297643894371295' (ECDSA) to the list of known hosts.
client-impl.py                                                                                        100% 1594    24.7KB/s   00:00
Transferring (2/2) Files to client-instance
server-ip.txt                                                                                         100%   11     0.2KB/s   00:00
--------------------------------
```

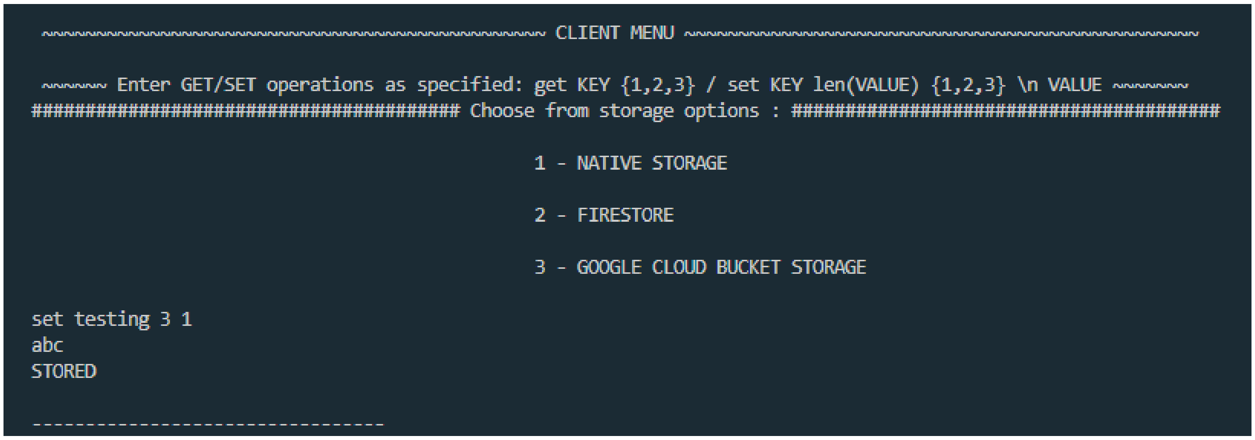
4. To run the VMs select 4. This installs the packages required to run the programs and also runs the server and client programs. The output of this is a client menu which waits for user request.



```
  Downloading cffi-1.15.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (441 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 441.2/441.2 kB 47.8 MB/s eta 0:00:00
Collecting rsa<5,>=3.1.4
  Downloading rsa-4.9-py3-none-any.whl (34 kB)
Collecting cachetools<6.0,>=2.0.0
  Downloading cachetools-5.2.0-py3-none-any.whl (9.3 kB)
Collecting six>=1.9.0
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Collecting pyasn1-modules>=0.2.1
  Downloading pyasn1_modules-0.2.8-py2.py3-none-any.whl (155 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 155.3/155.3 kB 26.3 MB/s eta 0:00:00
Collecting google-crc32c<2.0dev,>=1.0
  Downloading google_crc32c-1.5.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (32 kB)
Collecting pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2
  Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.3/98.3 kB 16.9 MB/s eta 0:00:00
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.12-py2.py3-none-any.whl (140 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 140.4/140.4 kB 21.8 MB/s eta 0:00:00
Collecting certifi>=2017.4.17
  Downloading certifi-2022.9.24-py3-none-any.whl (161 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 161.1/161.1 kB 26.3 MB/s eta 0:00:00
Collecting idna<4,>=2.5
  Downloading idna-3.4-py3-none-any.whl (61 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 61.5/61.5 kB 11.8 MB/s eta 0:00:00
Collecting charset-normalizer<3,>=2
  Downloading charset_normalizer-2.1.1-py3-none-any.whl (39 kB)
Collecting pycparser
  Downloading pycparser-2.21-py2.py3-none-any.whl (118 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 118.7/118.7 kB 19.4 MB/s eta 0:00:00
Collecting pyasn1<0.5.0,>=0.4.6
  Downloading pyasn1-0.4.8-py2.py3-none-any.whl (77 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 77.1/77.1 kB 14.6 MB/s eta 0:00:00
Installing collected packages: pyasn1, msgpack, google.cloud, urllib3, uritemplate, six, rsa, pyparsing, pyjwt, pycparser, pyasn1-modules, protobuf, idna, google-crc32c, charset-normalizer, certifi, cachetools, reques
ts, proto-plus, httplib2, grpcio, googleapis-common-protos, google-resumable-media, google-auth, cffi, grpcio-status, google-auth-httplib2, google-api-core, cryptography, cachecontrol, google-cloud-core, google-api-py
thon-client, google-cloud-storage, google-cloud-firestore, firebase-admin
  WARNING: The scripts pyrsa-decrypt, pyrsa-encrypt, pyrsa-keygen, pyrsa-priv2pub, pyrsa-sign and pyrsa-verify are installed in '/home/prashasti/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The script normalizer is installed in '/home/prashasti/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
  WARNING: The script doesitcache is installed in '/home/prashasti/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed cachecontrol-0.12.11 cachetools-5.2.0 certifi-2022.9.24 cffi-1.15.1 charset-normalizer-2.1.1 cryptography-38.0.3 firebase-admin-6.0.1 google-api-core-2.10.2 google-api-python-client-2.65.0 googl
e-auth-2.14.1 google-auth-httplib2-0.1.0 google-cloud-core-2.3.2 google-cloud-firestore-2.7.2 google-cloud-storage-2.6.0 google-crc32c-1.5.0 google-resumable-media-2.4.0 google.cloud-0.34.0 googleapis-common-protos-1.
56.4 grpcio-1.50.0 grpcio-status-1.50.0 httplib2-0.21.0 idna-3.4 msgpack-1.0.4 proto-plus-1.22.1 protobuf-4.21.9 pyasn1-0.4.8 pyasn1-modules-0.2.8 pycparser-2.21 pyjwt-2.6.0 pyparsing-3.0.9 requests-2.28.1 rsa-4.9 six
-1.16.0 uritemplate-4.1.1 urllib3-1.26.12
Running server-instance in Background
Running client-instance
Client-instance is open for testing...please wait for the menu
▯
```

ASSIGNMENT 3

CLOUD KV STORE



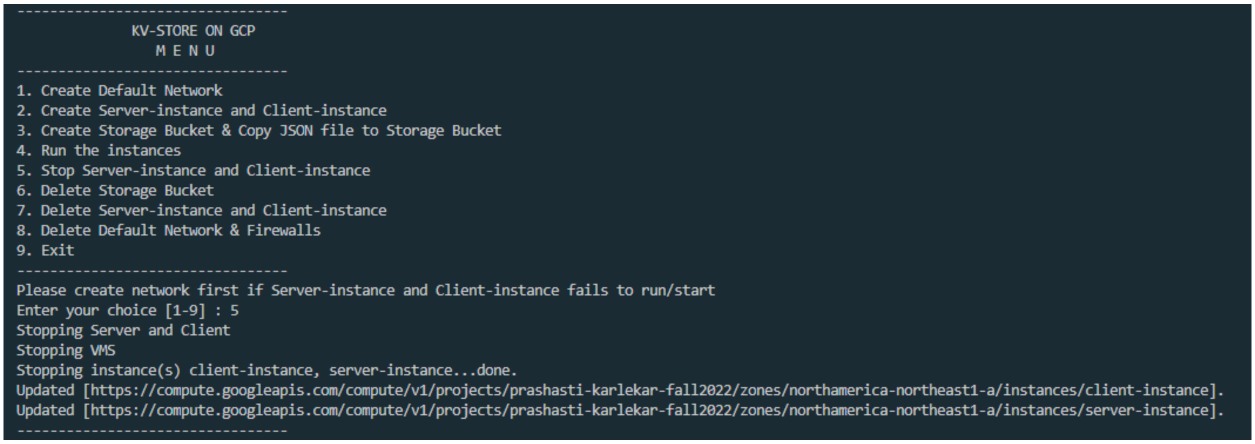
```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ CLIENT MENU ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 ~~~~~~ Enter GET/SET operations as specified: get KEY {1,2,3} / set KEY len(VALUE) {1,2,3} \n VALUE ~~~~~~~
######################################## Choose from storage options : ########################################

                                              1 - NATIVE STORAGE

                                              2 - FIRESTORE

                                              3 - GOOGLE CLOUD BUCKET STORAGE

set testing 3 1
abc
STORED

--------------------------------
```

5. For the native storage, the set operation writes to a newly created file with filename of the key in the data directory and the get operation reads the requested key's value from the data directory
6. For Firestore, the cloud firestore "Prashasti-karlekar-fall2022" was created. The credentials file for this store was then downloaded and is present in the project folder. The database contains a collection "KVStore" which then contains a document called "test". This document is written to/read from the server program to utilize firestore as a database for key-value store.
7. For Google Cloud bucket storage, the commands in script file in create_bucket() was performed.

   ```
   gcloud storage buckets create gs://prashasti_kvstore --location=US-EAST1 -
   -uniform-bucket-level-access
   ```

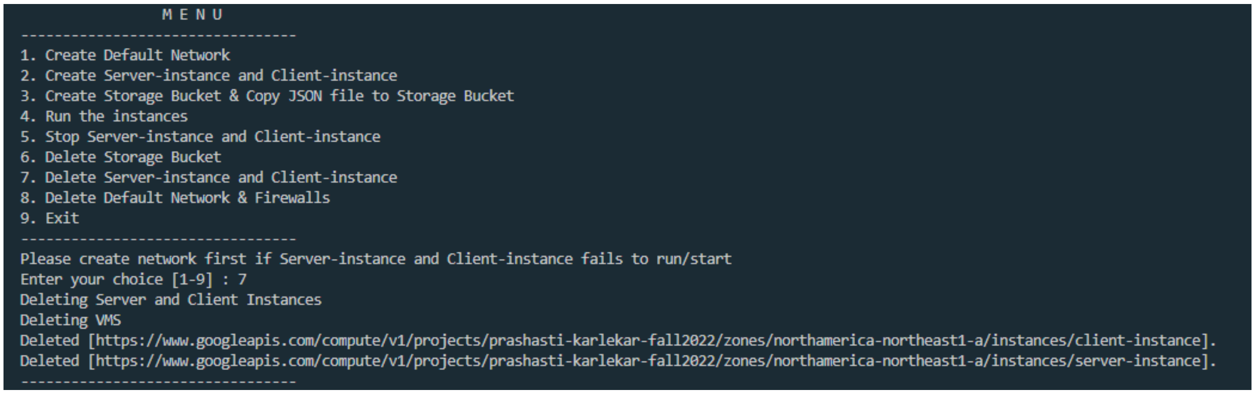
   This creates a bucket in Google cloud with name prashasti_kvstore.
   Also, an empty JSON file was copied onto the bucket for its storage utilization.

8. The client feeds a command with a specific storage option from {1,2,3} where 1 is native storage, 2 is Firestore and 3 is Google Cloud Bucket. Accordingly the server processes the client request and sends the output to the client.
9. To stop the VMs, enter option 5. This stops the client-instance and server-instance.



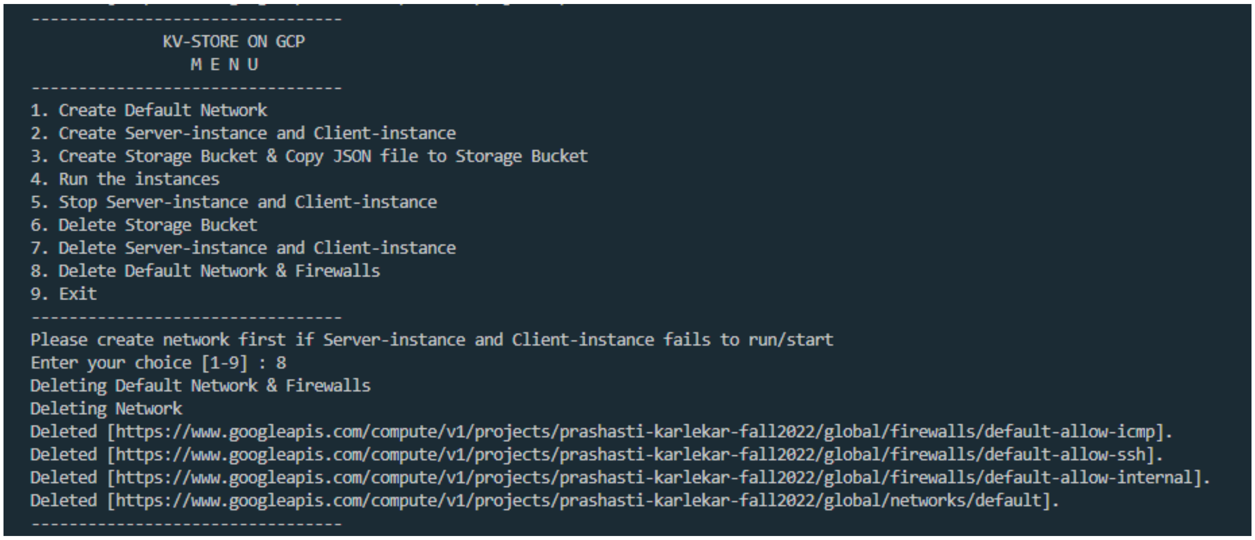
```
--------------------------------
            KV-STORE ON GCP
               M E N U
--------------------------------
1. Create Default Network
2. Create Server-instance and Client-instance
3. Create Storage Bucket & Copy JSON file to Storage Bucket
4. Run the instances
5. Stop Server-instance and Client-instance
6. Delete Storage Bucket
7. Delete Server-instance and Client-instance
8. Delete Default Network & Firewalls
9. Exit
--------------------------------
Please create network first if Server-instance and Client-instance fails to run/start
Enter your choice [1-9] : 5
Stopping Server and Client
Stopping VMS
Stopping instance(s) client-instance, server-instance...done.
Updated [https://compute.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/zones/northamerica-northeast1-a/instances/client-instance].
Updated [https://compute.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/zones/northamerica-northeast1-a/instances/server-instance].
--------------------------------
```

10. To delete the storage bucket, enter 6.
11. To delete the server-instance and client-instance, enter 7.



```
               M E N U
--------------------------------
1. Create Default Network
2. Create Server-instance and Client-instance
3. Create Storage Bucket & Copy JSON file to Storage Bucket
4. Run the instances
5. Stop Server-instance and Client-instance
6. Delete Storage Bucket
7. Delete Server-instance and Client-instance
8. Delete Default Network & Firewalls
9. Exit
--------------------------------
Please create network first if Server-instance and Client-instance fails to run/start
Enter your choice [1-9] : 7
Deleting Server and Client Instances
Deleting VMS
Deleted [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/zones/northamerica-northeast1-a/instances/client-instance].
Deleted [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/zones/northamerica-northeast1-a/instances/server-instance].
--------------------------------
```

12. To delete default network and set firewall rules, enter 8.



```
--------------------------------
            KV-STORE ON GCP
               M E N U
--------------------------------
1. Create Default Network
2. Create Server-instance and Client-instance
3. Create Storage Bucket & Copy JSON file to Storage Bucket
4. Run the instances
5. Stop Server-instance and Client-instance
6. Delete Storage Bucket
7. Delete Server-instance and Client-instance
8. Delete Default Network & Firewalls
9. Exit
--------------------------------
Please create network first if Server-instance and Client-instance fails to run/start
Enter your choice [1-9] : 8
Deleting Default Network & Firewalls
Deleting Network
Deleted [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/global/firewalls/default-allow-icmp].
Deleted [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/global/firewalls/default-allow-ssh].
Deleted [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/global/firewalls/default-allow-internal].
Deleted [https://www.googleapis.com/compute/v1/projects/prashasti-karlekar-fall2022/global/networks/default].
--------------------------------
```

ASSIGNMENT 3

CLOUD KV STORE

IMPLEMENTATION:

1. TCP- socket server:

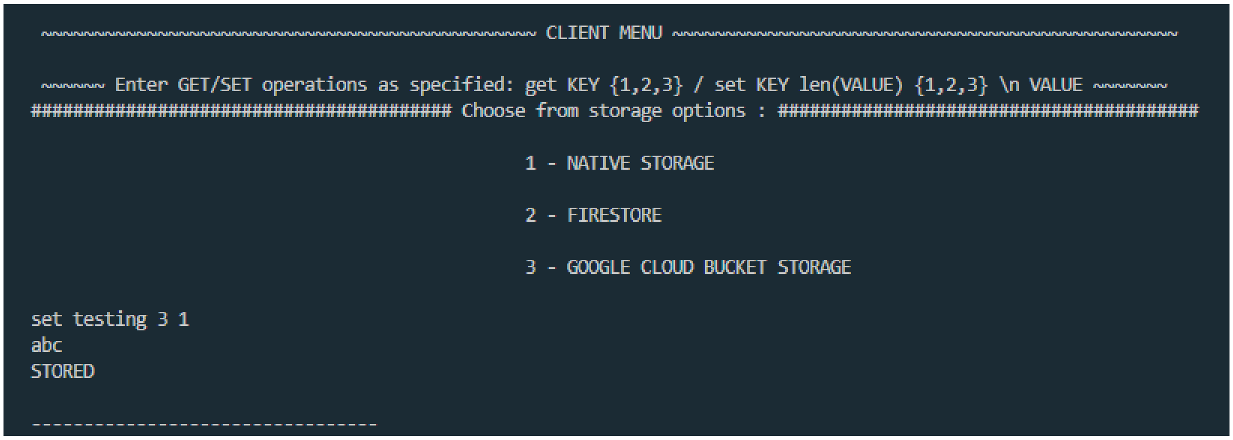
The server was implemented using the socket library offered by python. Upon creation of the server socket, it is associated with a specific network interface of server-instance running on Google cloud and port number. Once it accepts a connection from a client, it returns a new socket object representing the connection and a tuple holding the address of the client i.e. (host, port). This socket is then used to communicate with the client. Based on the client request, the set or get operation is performed for the specified storage option and a response for the request is sent back to the client.

- SET COMMAND:
  This stores the value for later retrieval.

  set <key> <value-size-bytes> <storage_option> \r\n

  <value> \r\n

  The set command is whitespace delimited. It takes the key to be stored, the length of the value associated with the key, the storage option where this key needs to be stored and the value itself as the arguments. The server responds with either "STORED\r\n", or "NOT-STORED\r\n" depending on whether the key was stored or not.

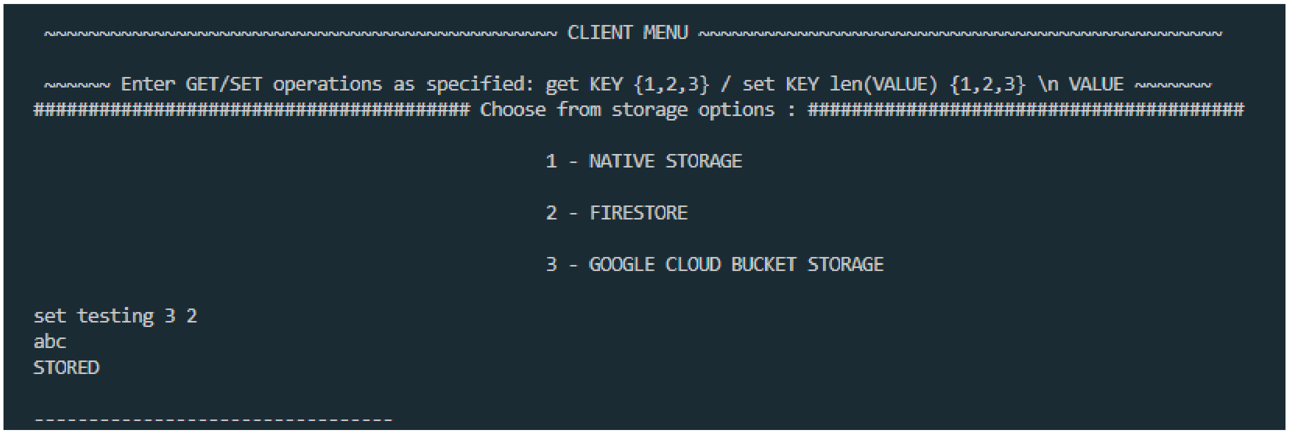


```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ CLIENT MENU ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 ~~~~~~ Enter GET/SET operations as specified: get KEY {1,2,3} / set KEY len(VALUE) {1,2,3} \n VALUE ~~~~~~~
######################################## Choose from storage options : ########################################

                                              1 - NATIVE STORAGE

                                              2 - FIRESTORE

                                              3 - GOOGLE CLOUD BUCKET STORAGE

set testing 3 1
abc
STORED

---------------------------------
```

SET operation for Native Storage

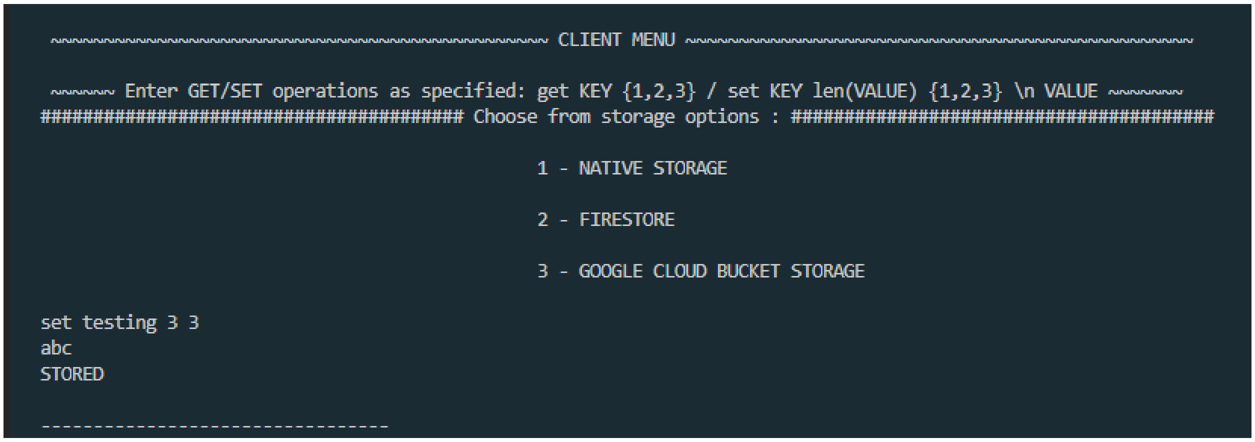
ASSIGNMENT 3

CLOUD KV STORE



```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ CLIENT MENU ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 ~~~~~~~ Enter GET/SET operations as specified: get KEY {1,2,3} / set KEY len(VALUE) {1,2,3} \n VALUE ~~~~~~~
######################################## Choose from storage options : ########################################

                                              1 - NATIVE STORAGE

                                              2 - FIRESTORE

                                              3 - GOOGLE CLOUD BUCKET STORAGE

set testing 3 2
abc
STORED

---------------------------------
```

SET operation for Firestore



```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ CLIENT MENU ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 ~~~~~~~ Enter GET/SET operations as specified: get KEY {1,2,3} / set KEY len(VALUE) {1,2,3} \n VALUE ~~~~~~~
######################################## Choose from storage options : ########################################

                                              1 - NATIVE STORAGE

                                              2 - FIRESTORE

                                              3 - GOOGLE CLOUD BUCKET STORAGE

set testing 3 3
abc
STORED

---------------------------------
```

SET operation for Google Cloud Bucket Storage

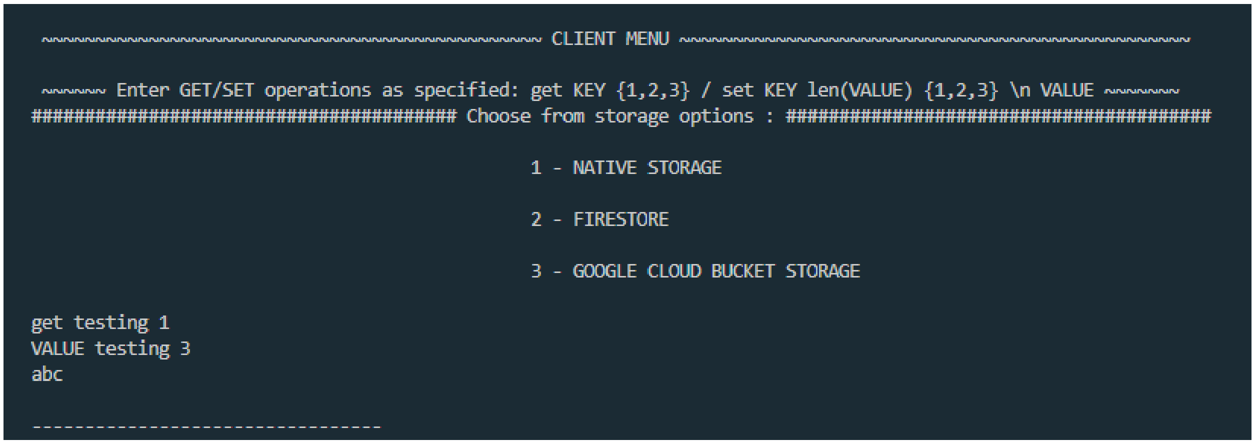
- GET COMMAND:
  This will fetch the data corresponding to the key and return it to the client

  get <key> <storage_option>\r\n

  The get command gets the value associated with the given key from selected storage. It accepts the key name and storage option as the argument.

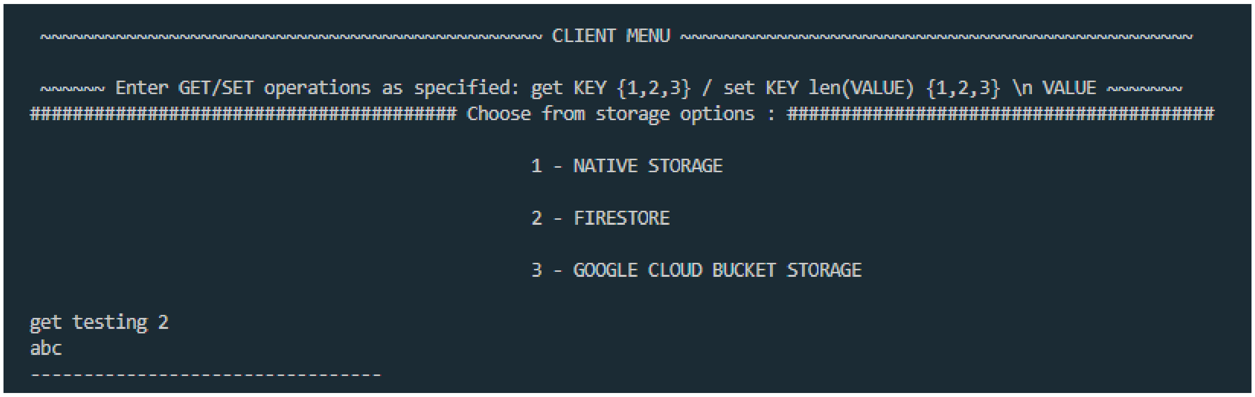
ASSIGNMENT 3

CLOUD KV STORE



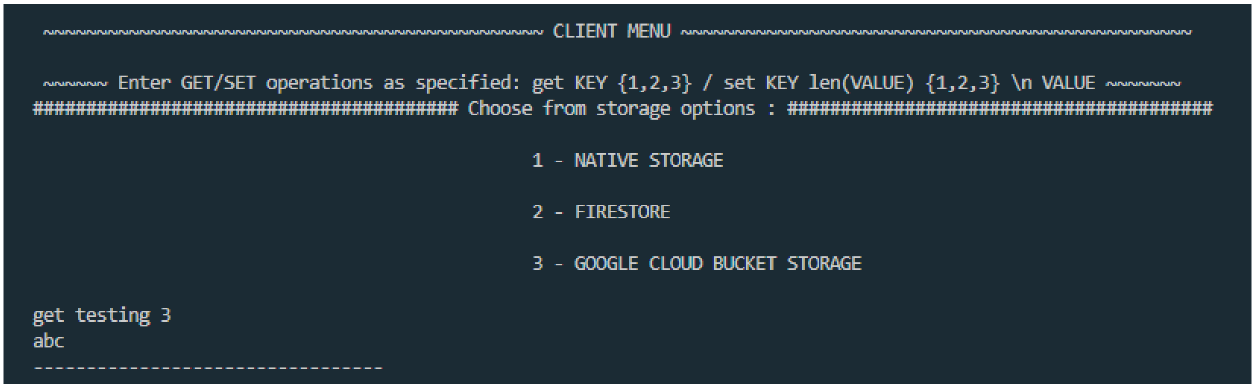
```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ CLIENT MENU ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 ~~~~~~~ Enter GET/SET operations as specified: get KEY {1,2,3} / set KEY len(VALUE) {1,2,3} \n VALUE ~~~~~~~~
######################################## Choose from storage options : ########################################

                                               1 - NATIVE STORAGE

                                               2 - FIRESTORE

                                               3 - GOOGLE CLOUD BUCKET STORAGE

get testing 1
VALUE testing 3
abc

--------------------------------
```

GET operation for Native Storage



```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ CLIENT MENU ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 ~~~~~~~ Enter GET/SET operations as specified: get KEY {1,2,3} / set KEY len(VALUE) {1,2,3} \n VALUE ~~~~~~~~
######################################## Choose from storage options : ########################################

                                               1 - NATIVE STORAGE

                                               2 - FIRESTORE

                                               3 - GOOGLE CLOUD BUCKET STORAGE

get testing 2
abc
--------------------------------
```

GET operation for Firestore



```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ CLIENT MENU ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

 ~~~~~~~ Enter GET/SET operations as specified: get KEY {1,2,3} / set KEY len(VALUE) {1,2,3} \n VALUE ~~~~~~~~
######################################## Choose from storage options : ########################################

                                               1 - NATIVE STORAGE

                                               2 - FIRESTORE

                                               3 - GOOGLE CLOUD BUCKET STORAGE

get testing 3
abc
--------------------------------
```

GET operation for Google Cloud Bucket Storage

2. Client:
   The client program connects to the server and makes get or set requests. It sends a command line/data block and receives a response from the server which indicates the success or failure of request made.

STORAGE BACKENDS - WHAT & WHY:

Native Storage:

When a large number of written and read from the data store, affects the set operation in terms of space requirements. But the get operation is fast. With the increase in the number of keys, the space requirements increase and hence does not make this model very scalable. Few limitations and improvements of native storage:

1. The current implementation of the server has a single process concurrency which allows easy caching and database access but has potential disadvantages like blockage of services when busy.
2. In-memory data structure stores like Redis can be implemented for better performance and improve the scalability.
3. The response time of server is merely 1/100th of a second for few requests, but the same can be tested for 1000s of client requests.
4. There are size limits associated with the value provided and this can be enhanced by accepting input files from the system.

Cloud Firestore:

Why I chose this:

This NoSQL data model stores data as collections of documents which is easier to compose as a key-value store. The documents store key-value pairs in dictionary form.

Advantages:

- Ideal for key-value pairs since the structure is similar to JSON format
- Performs better when scaled largely as it used subcollections within documents
- Since the queries are indexed, the performance of the query does not depend on the data set but instead on the result set

Total costs:

Pay for what you use. According to the official site, it is $11.10/month for read/write costs + $1.04/month for storage/networking costs

ASSIGNMENT 3

CLOUD KV STORE

Performance:

- Firestore is highly scalable and supports parallel reads.
- Retries the failed transactions to deal with transient errors
- Updating to a single document more than once per second leads to high latency and timeouts.

Google Cloud Bucket:

Why I chose this:

Buckets are the basic containers that holds the data. Google Cloud provides advantages like easy identity management, object versioning and authentication.

Advantages:

- With multiple storage bucket locations, it allows multiple redundancy options.
- Provides a low cost for data backup and archival
- Provides uniformly control access to Cloud storage resources
- Enable Pub/Sub notifications for Cloud Storage