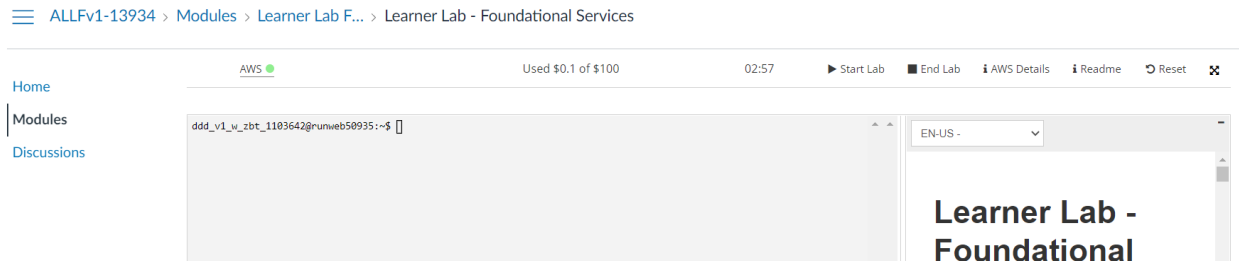


Laboratorio 2 Tópicos especiales en Telemática

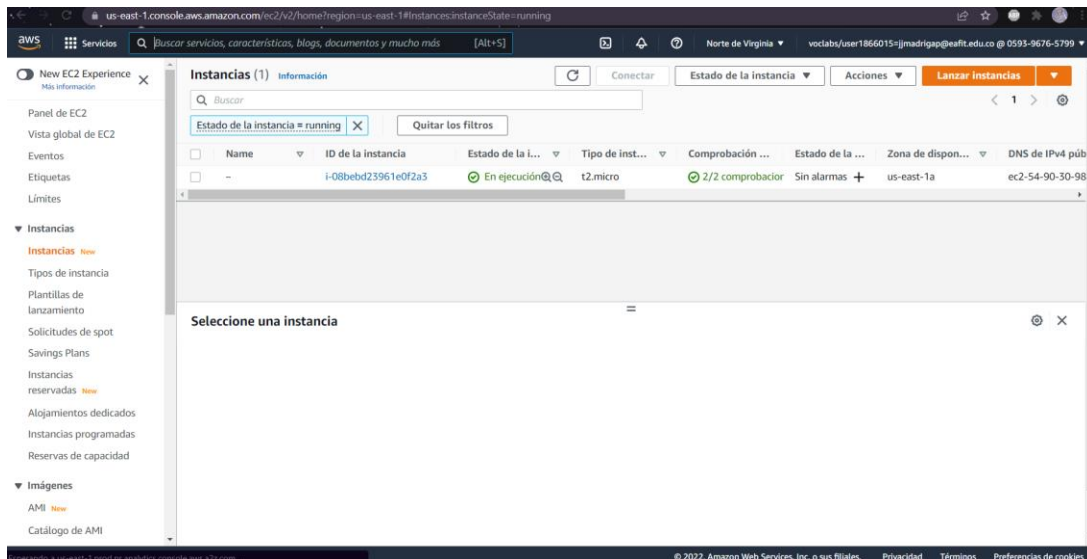
Juan Jose Madrigal Palacio 1000059306

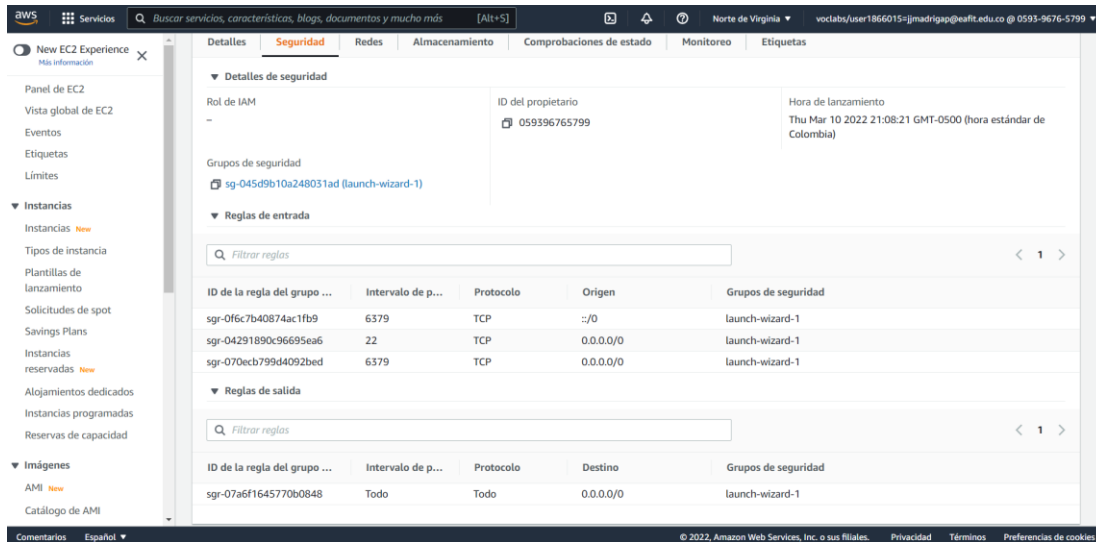
Punto 1 Infraestructura TI para la actividad.

Primero para la realización del laboratorio se fue activo el modulo de aws proporcionado para los laboratorios y luego se accedió a la consola de AWS.



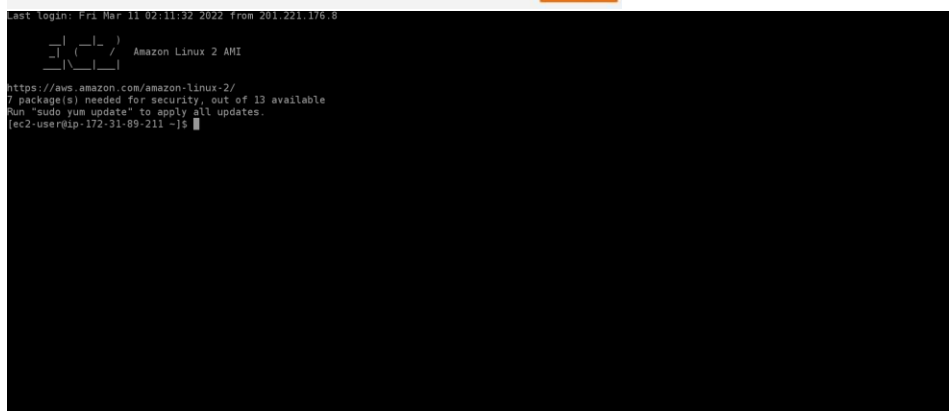
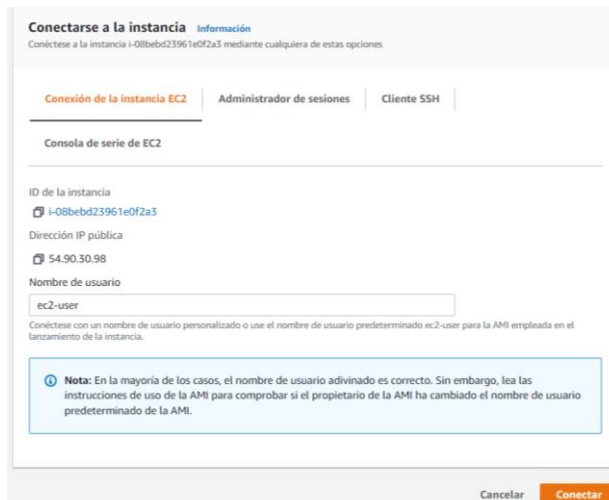
Luego en este se fue hacia la sección de EC2, en la cual se creó una nueva instancia en este con las especificaciones de que este corriera en el puerto tcp 6379





Después se conectó a la instancia al seleccionar la instancia y darle click al botón conectar, y para modificarla o hacer uso de la misma se puede realizar de 2 formas distintas:

1- Dándole al botón de conectar lo cual abre una terminal en una nueva ventana del navegador



i-08bebd23961e0f2a3
Public IPs: 54.90.30.98 Private IPs: 172.31.89.211

2- realizar la conexión mediante ssh, para lo cual primero vamos a la pestaña cliente ssh y en la terminal de nuestro computador usamos el comando que aparece en la parte del ejemplo, el cual vara según la clave de la instancia la cual tenemos descargada en nuestro computador y la IP publica que tenga la instancia en ese momento.

Conectarse a la instancia [Información](#)

Conéctese a la instancia i-08bebd23961e0f2a3 mediante cualquiera de estas opciones

Conexión de la instancia EC2

Administrador de sesiones

Ciente SSH

Consola de serie de EC2

ID de la instancia

i-08bebd23961e0f2a3

1. Abra un cliente SSH.
2. Localice el archivo de clave privada. La clave utilizada para lanzar esta instancia es kuro-22.pem
3. Ejecute este comando, si es necesario, para garantizar que la clave no se pueda ver públicamente.
 `chmod 400 kuro-22.pem`
4. Conéctese a la instancia mediante su DNS público:
 `ec2-54-90-30-98.compute-1.amazonaws.com`

Ejemplo:

`ssh -i "kuro-22.pem" ec2-user@ec2-54-90-30-98.compute-1.amazonaws.com`

Nota: En la mayoría de los casos, el nombre de usuario adivinado es correcto. Sin embargo, lea las instrucciones de uso de la AMI para comprobar si el propietario de la AMI ha cambiado el nombre de usuario predeterminado de la AMI.

```
PS C:\Users\juan> ssh -i "C:\Users\juan\Desktop\Universidad\Temas Telematicos\Lab 2\kuro-22.pem" ec2-user@ec2-18-209-23-206.compute-1.amazonaws.com
The authenticity of host 'ec2-18-209-23-206.compute-1.amazonaws.com (18.209.23.206)' can't be established.
ECDSA key fingerprint is SHA256:Q2/sTLe5tsMgygrtLUib57BxYVyclzj4FkgbwinQuWw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-209-23-206.compute-1.amazonaws.com,18.209.23.206' (ECDSA) to the list of known hosts.
_ _ _ _ _
| | | | |
|_| ( _/ Amazon Linux 2 AMI
_ _ _ _ _

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 5 packages available
Run "sudo yum update" to apply all updates.
```

Cuando ya la tengamos iniciamos a continuación la creación de la base de datos redis por ahora en el modo single-node

Lo primero que se hace en el nodo es instalar gcc como requisito para luego la instalación y creación de redis

```
[ec2-user@ip-172-31-22-133 ~]$ sudo yum install gcc
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                     | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
--> Package gcc.x86_64 0:7.3.1-13.amzn2 will be installed
--> Processing Dependency: cpp = 7.3.1-13.amzn2 for package: gcc-7.3.1-13.amzn2.x86_64
--> Processing Dependency: libsanitizer >= 7.3.1-13.amzn2 for package: gcc-7.3.1-13.amzn2.x86_64
--> Processing Dependency: libquadmath >= 7.3.1-13.amzn2 for package: gcc-7.3.1-13.amzn2.x86_64
--> Processing Dependency: libmpx >= 7.3.1-13.amzn2 for package: gcc-7.3.1-13.amzn2.x86_64
--> Processing Dependency: libitm >= 7.3.1-13.amzn2 for package: gcc-7.3.1-13.amzn2.x86_64
--> Processing Dependency: libcilkrts >= 7.3.1-13.amzn2 for package: gcc-7.3.1-13.amzn2.x86_64
--> Processing Dependency: libatomic >= 7.3.1-13.amzn2 for package: gcc-7.3.1-13.amzn2.x86_64
--> Processing Dependency: glibc-devel >= 2.2.90-12 for package: gcc-7.3.1-13.amzn2.x86_64
--> Processing Dependency: libmpfr.so.4()(64bit) for package: gcc-7.3.1-13.amzn2.x86_64
--> Processing Dependency: libmpc.so.3()(64bit) for package: gcc-7.3.1-13.amzn2.x86_64
```

Luego como indica la creación de una base de datos redis se usa el comando

wget http://download.redis.io/redis-stable.tar.gz

```
[ec2-user@ip-172-31-22-133 ~]$ wget http://download.redis.io/redis-stable.tar.gz
--2022-03-10 22:02:34-- http://download.redis.io/redis-stable.tar.gz
Resolving download.redis.io (download.redis.io)... 45.60.121.1
Connecting to download.redis.io (download.redis.io)[45.60.121.1]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2527624 (2.4M) [application/octet-stream]
Saving to: 'redis-stable.tar.gz'

100%[=====>] 2,527,624 --.-K/s in 0.1s

2022-03-10 22:02:34 (17.4 MB/s) - 'redis-stable.tar.gz' saved [2527624/2527624]
```

tar xvf redis-stable.tar.gz

```
[ec2-user@ip-172-31-22-133 ~]$ tar xvf redis-stable.tar.gz
redis-stable/
redis-stable/CONTRIBUTING
redis-stable/src/
redis-stable/src/geohash.h
redis-stable/src/redis-trib.rb
```

cd redis-stable

make

```
[ec2-user@ip-172-31-22-133 ~]$ cd redis-stable
[ec2-user@ip-172-31-22-133 redis-stable]$ make
cd src && make all
make[1]: Entering directory '/home/ec2-user/redis-stable/src'
cc Makefile.dep
make[1]: Leaving directory '/home/ec2-user/redis-stable/src'
make[1]: Entering directory '/home/ec2-user/redis-stable/src'
rm -rf redis-server redis-sentinel redis-cli redis-benchmark redis-check-rdb redis-check-aof *.o *.gcda *.gco *.gcov redis.info lcov-html Makefile.dep
rm -f adlist.d quicklist.d ae.d anet.d dict.d server.d sds.d zmalloc.d lzf.c.d lzf.d.d pgsort.d zipmap.d sha1.d ziplist.d release.d networking.d util.d obje
ct.d db.d replication.d rdb.d t_string.d t_list.d t_set.d t_zset.d t_hash.d config.d aof.d pubsub.d multi.d debug.d sort.d intset.d syncio.d cluster.d crc16
.d endianconv.d slowlog.d scripting.d bio.d rio.d rand.d memtest.d crcspeed.d crc64.d bitops.d sentinel.d notify.d setproctitle.d blocked.d hyperloglog.d la
```

y por último: *sudo make install.*

Cuando ya tenemos listo la base de datos redis en nuestra instancia la podemos ejecutar con este comando: *redis-server*

Pero primero se va a configurar la contraseña para ingresar por lo que vamos a la carpeta *\redis-table* y buscamos el archivo *redis.conf* el editaremos con por ejemplo vim o nano, dentro de este buscaremos una parte llamada *requirepass* y le quitaremos el '#' para que deje de ser un comentario y cambiaremos lo que sigue por la contraseña de nuestra preferencia.

```
# IMPORTANT NOTE: starting with Redis 6 "requirepass" is just a compatibility
# layer on top of the new ACL system. The option effect will be just setting
# the password for the default user. Clients will still authenticate using
# AUTH <password> as usually, or more explicitly with AUTH default <password>
# if they follow the new protocol: both will work.
#
# The requirepass is not compatible with aclfile option and the ACL LOAD
# command, these will cause requirepass to be ignored.
#
requirepass madrigall123

# New users are initialized with restrictive permissions by default, via the
# equivalent of this ACL rule 'off resetkeys -@all'. Starting with Redis 6.2, it
# is possible to manage access to Pub/Sub channels with ACL rules as well. The
# default Pub/Sub channels permission if new users is controlled by the
# acl-pubsub-default configuration directive, which accepts one of these values:
#
"redis.conf" 2051L, 93726B written
[ec2-user@ip-172-31-89-211 redis-stable]$ redis-cli
```

Ahora para ejecutar la base de datos y que se efectuó el cambio de la contraseña debemos ejecutar el comando como sigue: *redis-server /home/ec2-user/redis-stable/redis.conf*

```
[ec2-user@ip-172-31-89-211 redis-stable]$ redis-server /home/ec2-user/redis-stable/redis.conf
10555:C 08 Mar 2022 02:53:15.335 # o000o000o000o Redis is starting o000o000o000o
10555:C 08 Mar 2022 02:53:15.335 # Redis version=6.2.6, bits=64, commit=00000000, modified=0, pid=10555, just started
10555:C 08 Mar 2022 02:53:15.335 # Configuration loaded
10555:M 08 Mar 2022 02:53:15.336 * monotonic clock: POSIX clock_gettime

Redis 6.2.6 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 10555

https://redis.io

10555:M 08 Mar 2022 02:53:15.336 # Server initialized
10555:M 08 Mar 2022 02:53:15.337 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
10555:M 08 Mar 2022 02:53:15.337 * Loading RDB produced by version 6.2.6
10555:M 08 Mar 2022 02:53:15.337 * RDB age 126 seconds
10555:M 08 Mar 2022 02:53:15.337 * RDB memory usage when created 0.77 Mb
10555:M 08 Mar 2022 02:53:15.337 # Done loading RDB, keys loaded: 0, keys expired: 0.
10555:M 08 Mar 2022 02:53:15.337 * DB loaded from disk: 0.000 seconds
10555:M 08 Mar 2022 02:53:15.337 * Ready to accept connections
```

Por último, para probar la conexión es necesario que abramos una nueva terminal en la cual para probar el funcionamiento primero ingresamos a la *redis-cli* usando el comando del mismo nombre, se hace *AUTH* (contraseña) para ingresar y luego se usa el comando *ping* y si nos regresa un *PONG* para verificar que hay conexión.

```
console.aws.amazon.com/ec2/v2/connect/ec2-user/i-08bebd23961e0f2a3
Last login: Tue Mar 8 02:53:24 2022 from ec2-18-206-107-25.compute-1.amazonaws.com

 _ _ | ( _ _ |
 _ _ | _ _ |
 _ _ | _ _ |

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-89-211 ~]$ redis-cli
127.0.0.1:6379> PING
(error) NOAUTH Authentication required.
127.0.0.1:6379> AUTH madrigal1123
OK
127.0.0.1:6379> PING
PONG
```

Punto 2 Operaciones de ejemplo CRUD en redis-cli

Dentro de la terminal de la instancia se realizaron las siguientes actividades en la redis-cli, las cuales aparecían dentro del libro “Siete bases de datos en Siete Semanas” Capitulo 8

```
console.aws.amazon.com/ec2/v2/connect/ec2-user/i-08bebd23961e0f2a3
Last login: Tue Mar  8 02:53:24 2022 from ec2-18-206-107-25.compute-1.amazonaws.com

 _ _ | _ _ | _ _ )
 _ | ( _ | _ /   Amazon Linux 2 AMI
 _ | \ _ | _ | _

https://aws.amazon.com/amazon-linux-2/
No packages needed for security; 1 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-89-211 ~]$ redis-cli
127.0.0.1:6379> PING
(error) NOAUTH Authentication required.
127.0.0.1:6379> AUTH madrigal1123
OK
127.0.0.1:6379> PING
PONG
127.0.0.1:6379>
127.0.0.1:6379>
127.0.0.1:6379> HELP
```

```
127.0.0.1:6379> HELP
redis-cli 6.2.6
To get help about Redis commands type:
"help @<group>" to get a list of commands in <group>
"help <command>" for help on <command>
"help <tab>" to get a list of possible help topics
"quit" to exit

To set redis-cli preferences:
":set hints" enable online hints
":set nohints" disable online hints
Set your preferences in ~/.redisclirc
127.0.0.1:6379> █
```

i-08bebd23961e0f2a3

Public IPs: 54.235.228.23 Private IPs: 172.31.89.211

```
127.0.0.1:6379> SET 7wks http://www.sevenweeks.org/
OK
127.0.0.1:6379> GET 7wks
"http://www.sevenweeks.org/"
127.0.0.1:6379>
127.0.0.1:6379> MSET gog http://www.google.com.ezproxy.eafit.edu.co yah http://www.yahoo.com
OK
127.0.0.1:6379> MGET gog yah
1) "http://www.google.com.ezproxy.eafit.edu.co"
2) "http://www.yahoo.com"
127.0.0.1:6379>
127.0.0.1:6379>
127.0.0.1:6379> SET count 2
OK
127.0.0.1:6379> INCR count
(integer) 3
127.0.0.1:6379> GET count
"3"
127.0.0.1:6379> █
```

i-08bebd23961e0f2a3

Public IPs: 54.235.228.23 Private IPs: 172.31.89.211

```
127.0.0.1:6379>
127.0.0.1:6379> MULTI
OK
127.0.0.1:6379(TX)> SET prag http://pragprog.com
QUEUED
127.0.0.1:6379(TX)> INCR count
QUEUED
127.0.0.1:6379(TX)> EXEC
1) OK
2) (integer) 4
127.0.0.1:6379>
```

Punto 3 Programas de CRUD hechos en Python

Para hacer las actividades en Python primero se verifico que se tuviera Python instalado y luego con pip se instalaron las dependencias necesarias de redis.

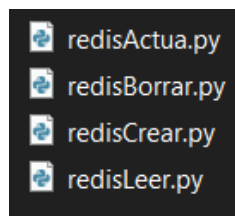
```
console.aws.amazon.com/ec2/v2/connect/ec2-user/i-08bebd23961e0f2a3
[ec2-user@ip-172-31-89-211 ~]$ python3 -V
Python 3.7.10
[ec2-user@ip-172-31-89-211 ~]$ pip3 install redis
Defaulting to user installation because normal site-packages is not writeable
Collecting redis
  Downloading redis-4.1.4-py3-none-any.whl (175 kB)
    | 175 kB 23.7 MB/s
Collecting deprecated>=1.2.3
  Downloading Deprecated-1.2.13-py2.py3-none-any.whl (9.6 kB)
Collecting packaging>=20.4
  Downloading packaging-21.3-py3-none-any.whl (40 kB)
    | 40 kB 8.8 MB/s
Collecting importlib-metadata>=1.0; python_version < "3.8"
  Downloading importlib_metadata-4.11.2-py3-none-any.whl (17 kB)
Collecting wrapt<2,>=1.10
  Downloading wrapt-1.13.3-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2010_x86_64.whl (79 kB)
    | 79 kB 12.8 MB/s
Collecting pyparsing!<3.0.5,>=2.0.2
```

Luego para crear los archivos.py en Windows se realizo la conexión por medio de la terminal (método 2), luego se creó una virtual env para trabajar con los comandos que se muestran abajo a la cual también se le instalo el redis

```
PS C:\Users\juan\Desktop\Universidad\Temas Telematicos\Lab 2\pythoncode> python -m virtualenv venv
created virtual environment CPython3.10.0.final.0-64 in 6419ms
creator CPython3Windows(dest=C:\Users\juan\Desktop\Universidad\Temas Telematicos\Lab 2\pythoncode\venv, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\juan\AppData\Local\pypa\virtualenv)
added seed packages: pip==22.0.4, setuptools==60.9.3, wheel==0.37.1
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
PS C:\Users\juan\Desktop\Universidad\Temas Telematicos\Lab 2\pythoncode>
PS C:\Users\juan\Desktop\Universidad\Temas Telematicos\Lab 2\pythoncode> Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope CurrentUser
PS C:\Users\juan\Desktop\Universidad\Temas Telematicos\Lab 2\pythoncode> .\venv\Scripts\activate
(venv) PS C:\Users\juan\Desktop\Universidad\Temas Telematicos\Lab 2\pythoncode> pip install redis
Collecting redis
  Downloading redis-4.1.4-py3-none-any.whl (175 kB)
```

Luego como prueba se corrio uno de los archivos que había en el repositorio mandado

```
(venv) PS C:\Users\juan\Desktop\Universidad\Temas Telematicos\Lab 2\pythoncode> python .\redisHelloWorld.py 52.23.226.19 madrigal1123
Displaying current welcome message...
message = None
Writing "Hello world from Python!" to Redis...
Displaying current welcome message...
message = b'Hello world from Python!'
```



Luego se crearon 4 archivos .py los cuales se pueden ver en la imagen, cada uno de estos se ejecuto en en entorno virtual para lo cual se ejecuto en la terminal un comando el cual priemro va el nombre del archivo seguido de la direccion ip de la instancia aws y por ultimo la contraseña de la instancia como se puede ver: *python .\redisHelloWorld.py ###.###.### madrigal1123*

En la siguiente imagen se puede ver la forma de usar el código y su ejecución.

```

(venv) PS C:\Users\juan\Desktop\Universidad\Topicos Telematica\Lab 2\pythoncode> python .\redisCrear.py 54.90.30.98 madrigal1123
Se va a escribir en redis pofavor ingrese los datos
Ingrese la llave/key para la escritura: 22
Ingrese el el dato/value que quiera escribir: holahola
Se logro escribir correctamente
(venv) PS C:\Users\juan\Desktop\Universidad\Topicos Telematica\Lab 2\pythoncode> python .\redisLeer.py 54.90.30.98 madrigal1123
Se va a leer de redis pofavor ingrese los datos
Ingrese la llave/key que quiere leer: 22
Key: 22, valor= b'holahola'
Se leyo correctamente
(venv) PS C:\Users\juan\Desktop\Universidad\Topicos Telematica\Lab 2\pythoncode> python .\redisActua.py 54.90.30.98 madrigal1123
Se va a actualizar un valor en redis pofavor ingrese los datos
Ingrese la llave/key a actualizar: 22
Ingrese el el dato/value con el que lo quiere actualizar: jajajajaja
Se logro actualizar correctamente
(venv) PS C:\Users\juan\Desktop\Universidad\Topicos Telematica\Lab 2\pythoncode> python .\redisLeer.py 54.90.30.98 madrigal1123
Se va a leer de redis pofavor ingrese los datos
Ingrese la llave/key que quiere leer: 22
Key: 22, valor= b'jajajajaja'
Se leyo correctamente
(venv) PS C:\Users\juan\Desktop\Universidad\Topicos Telematica\Lab 2\pythoncode> python .\redisBorrar.py 54.90.30.98 madrigal1123
Se va a borrar de redis pofavor ingrese los datos
Ingrese la llave/key que quiere borrar: 22
Se lobro borrar correctamente
(venv) PS C:\Users\juan\Desktop\Universidad\Topicos Telematica\Lab 2\pythoncode> python .\redisLeer.py 54.90.30.98 madrigal1123
Se va a leer de redis pofavor ingrese los datos
Ingrese la llave/key que quiere leer: 22
Key: 22, valor= None
Se leyo correctamente

```

Punto 4 Creación del clúster

Debido a fallos y problemas con la realización de este punto no se logró hacer a tiempo

Punto 5 Conclusiones

Para el primer proyecto queremos crear una API (la especificación puede ser encontrada en el repositorio de Github en la carpeta Project 1) y un cliente en Python similar a como se usó con StrictRedis. Pensamos que implementar solo las operaciones básicas de CRUD debería ser relativamente sencillo, a excepción de eliminar dependiendo del método de guardado en disco que decidamos utilizar.

Una de las especificaciones no funcionales para este proyecto que con total certeza usaremos será particionamiento (aunque solo estático) usando un algoritmo DHT simple. Replicación no será una prioridad, entendemos lo útil que es sin embargo creemos que deberíamos centrar nuestra atención en otras características. Nuestro proyecto debería ser consistente dado que solo habrá 3 nodos de datos y no es muy seguro que se realice la replicación. La tolerancia a fallos por ende no estará garantizada, sin embargo, no será completamente ignorada.

De este laboratorio con Redis tuvimos un buen acercamiento al concepto tanto de base de datos key-value como a base de datos distribuida, si bien pueden parecer relativamente más sencillas de usar dada su aparente simplicidad, no se deben subestimar en lo absoluto. Son herramientas ciertamente poderosas para las funciones que están pensadas a desempeñar y la observación de los diferentes métodos que contiene Redis nos ha dado una buena idea de que tipo de operaciones deberíamos manejar nosotros y de qué forma presentarlas.