

Create 'Replica Of' Redis Open Source server on Redis Enterprise Software.

1. Verify the connectivity to the remote hosts and **reply with confirmation** (or explain the connection issue you are facing):

```
Connection to 3.95.247.143 closed.
jairobrenes@Jairos-MacBook-Pro RedisExam % ssh -i SupportExam-use.pem ubuntu@3.95.247.143
Last login: Fri Apr  4 17:53:09 2025 from 45.65.188.7
ubuntu@ip-10-10-0-19:~$ █
Connection to 18.207.176.232 closed.
jairobrenes@Jairos-MacBook-Pro RedisExam % ssh -i SupportExam-use.pem ubuntu@18.207.176.232
Last login: Fri Apr  4 17:53:17 2025 from 45.65.188.13
ubuntu@ip-10-10-0-185:~$ █
```

2. Install standalone Redis Open Source server **version 3.0.7** on Server A.

```
ubuntu@ip-10-10-0-19:~$ redis-cli
127.0.0.1:6379> info
# Server
redis_version:3.0.7
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:5fb5b7a0c85c00af
redis_mode:standalone
```

3. Use `memtier_benchmark`, a tool created by Redis Ltd, to load data into the Redis database.

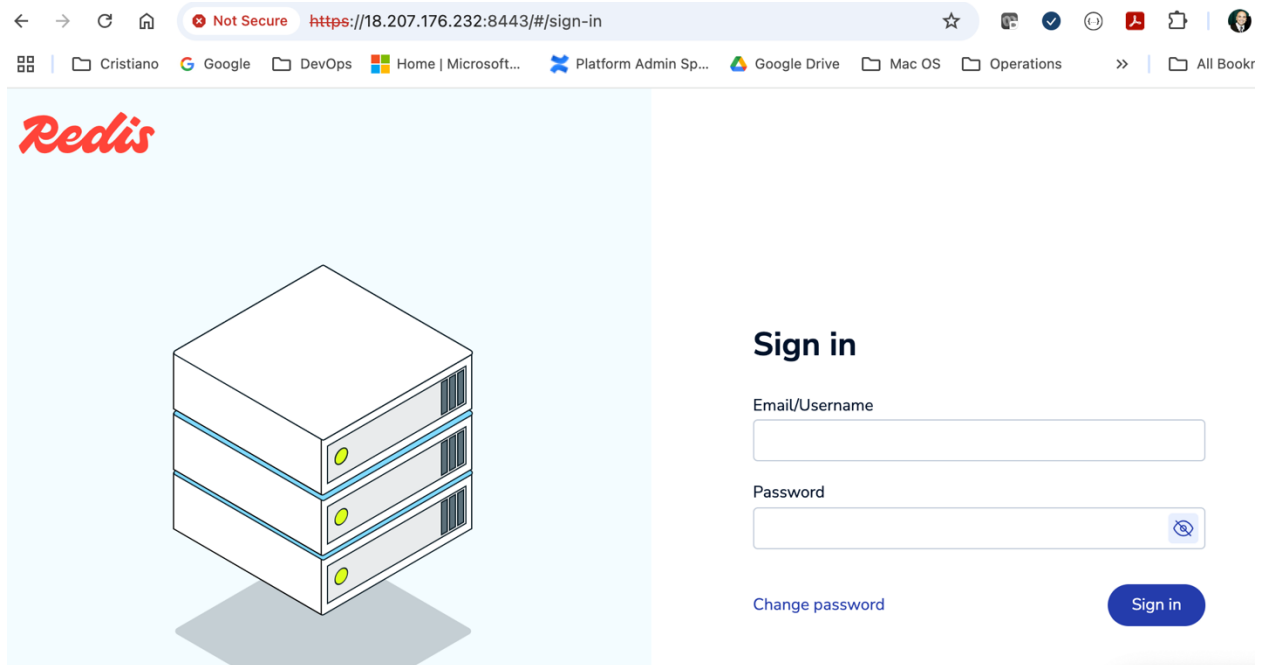
```
[ubuntu@ip-10-10-0-19:~$ memtier_benchmark --help
Usage: memtier_benchmark [options]
A memcache/redis NoSQL traffic generator and performance benchmarking tool.
```

Connection and General Options:

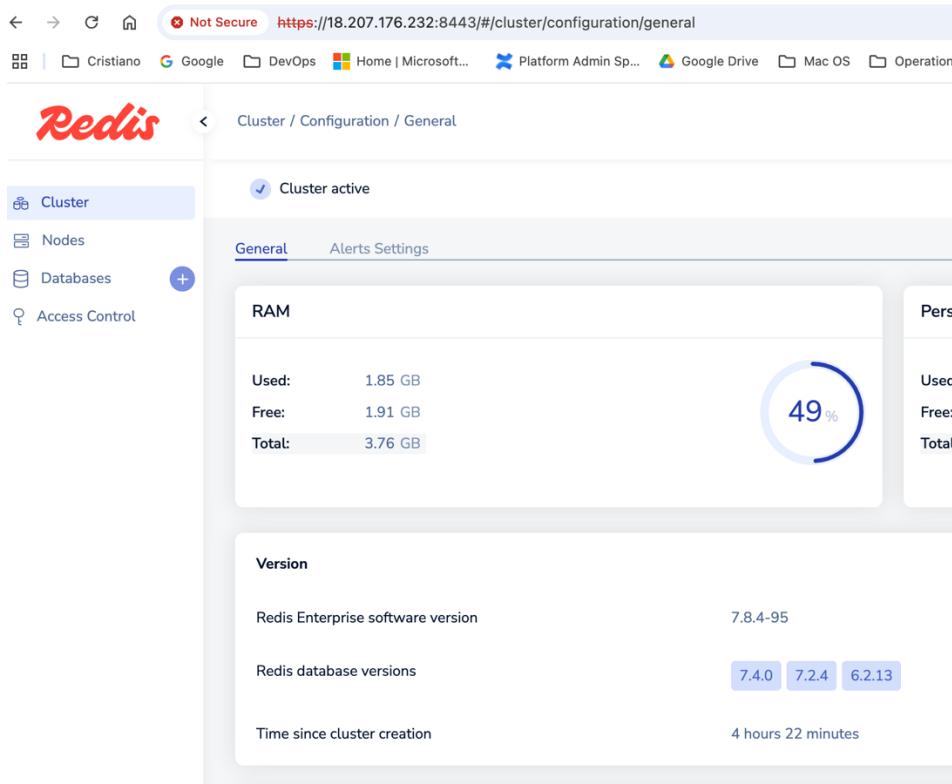
| | |
|--------------------------------|---|
| -h, --host=ADDR | Server address (default: localhost) |
| -s, --server=ADDR | Same as --host |
| -p, --port=PORT | Server port (default: 6379) |
| -S, --unix-socket=SOCKET | UNIX Domain socket name (default: none) |
| -4, --ipv4 | Force IPv4 address resolution. |
| -6, --ipv6 | Force IPv6 address resolution. |
| -P, --protocol=PROTOCOL | Protocol to use (default: redis). other supported protocols are resp2, resp3, memcache_text and memcache_binary. when using one of resp2 or resp3 the redis protocol version will be set via HELLO command. |
| -a, --authenticate=CREDENTIALS | Authenticate using specified credentials. A simple password is used for memcache_text and Redis <= 5.x. <USER>:<PASSWORD> can be specified for memcache_binary or Redis 6.x |

4. Install the latest version of Redis Enterprise Software on Server B.

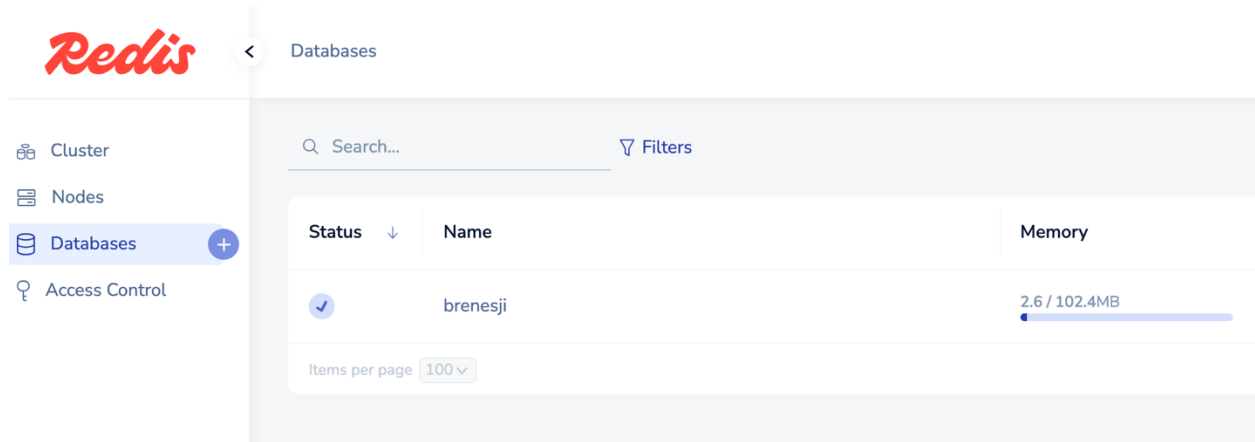
```
[ubuntu@ip-10-10-0-185:~]$ redis-cli -p 12000  
[127.0.0.1:12000> info  
# Server  
redis_version:7.4.0  
redis_git_sha1:00000000  
redis_git_dirty:0  
redis_build_id:0000000000000000000000000000000000000000000000000000000000000000  
redis_mode:standalone  
os:Linux 6.2.0-1017-aws x86_64  
arch_bits:64  
multiplexing_api:epoll  
gcc_version:11.4.0  
process_id:1  
run_id:dcbcd2b07a1d68abb5ab886dbb755c8293e6697d
```



5. Setup Redis Enterprise Software.



6. Create a Redis Database on Redis Enterprise Software



7. Send an email (Reply All) with a few details on how it is going so far.



Jairo Brenes <jairo.brenes@gmail.com>

to John, Steven, Ella, William ▾

Hi John and all

I have reached number 7 and test connectivity between servers, I was about to shot the email

Right now I'm working on 9 and 10.

Will let you know as soon as I finish.

Thanks!



8. Enable **Unidirectional Replica Of (Active-Passive)** between the two Redis servers:

```
# Replication
role:master
connected_slaves:1
slave0:ip=10.10.0.185,port=0,state=online,offset=51854,lag=1
master_repl_offset:51854
repl_backlog_active:1
repl_backlog_size:1048576
repl_backlog_first_byte_offset:2
repl_backlog_histlen:51853
```

| Replica Of | | | | |
|---|----------------------|----------------------|-------------|--------------------|
| Use this database as target | | | | |
| Connect source databases; Data on the source databases will be replicated to the target database. | | | | |
| Status | Source database name | Source database path | Compression | TLS authentication |
| ✓ Synced | N/A | N/A | N/A | Disabled |

9. Choose any technology to implement the following, using **the most efficient Redis Data Type**:

- Insert 100 random (not predictable) values to the Redis Open Source server.

```
[ubuntu@ip-10-10-0-19: ~]$ ./populatedatabase.sh
(integer) 1
(integer) 2
(integer) 3
(integer) 4
(integer) 5
(integer) 6
(integer) 7
(integer) 8
(integer) 9
```

```
cat names.txt | while read word; do redis-cli LPUSH names "$word"; done
```

- **Read and print the values in reverse insertion order** from the Redis Enterprise Software.

```
[ubuntu@ip-10-10-0-185:~/code$ ./readnormal.sh
1) "Tiffany"
2) "Steven"
3) "Sarah"
4) "Robert"
5) "Nicole"
6) "Michael"
7) "Elizabeth"
8) "David"
9) "Chelsea"
10) "Brian"
11) "Ashley"
12) "Eric"
13) "Danielle"
14) "Christopher"
15) "Brittany"
16) "Andrew"
17) "Zoe"
18) "Xavier"
19) "William"
20) "Vanessa"

[ubuntu@ip-10-10-0-185:~/code$ ./readrecursive.sh
Alice
Bob
Charlie
David
Eve
Frank
Grace
Henry
Ivy
Jack
Kate
Leo
Mia
Noah
Olivia
Peter
Quinn
Rose
Sam
Tom
Ava
Ben
Chloe
Daniel
Emily
```

```
[ubuntu@ip-10-10-0-185:~/code$ cat readnormal.sh
redis-cli -p 12000 LRange names 0 100
[ubuntu@ip-10-10-0-185:~/code$ cat readrecursive.sh
redis-cli -p 12000 LRange names 0 -1 | tac
ubuntu@ip-10-10-0-185:~/code$
```

- **Explain why it is the most efficient Redis Data Type for this task**

- In my opinion the best data type depends on the use case, for this I picked list because it needed to be not predictable hence this way we can read from bottom to top (I could have used sorted sets but this is more like when we need to do comparison between values.)

10. Keep the Redis servers running on both hosts after you exit.

Done.

- Username username: brenes.jairo@hotmail.com
- password: Redis6379