**Title:** Redis PSYNC

***Excerpt:*** The PSYNC command is issued by the replica instances to fetch incremental datasets and write commands from the master. It uses master replication id and offset values to obtain only the incremental chunk of data which is low latency and high performance.

**Permalink:** redis-psync

**Category:** Redis

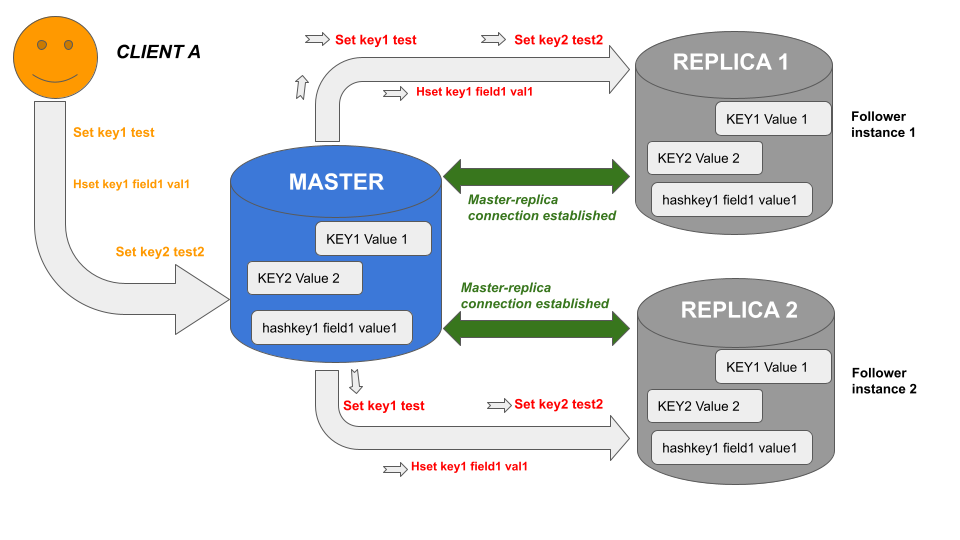
# 

# 

# 

# Introduction to Redis Replication

Redis is built with various features to ensure the high availability of its instances. Apart from advanced high availability features like clustering, and sentinel, the master-replica replication is available to configure and use with minimal effort. A master(leader) instance can have multiple follower instances where each follower is an exact copy of the leader instance. Whenever a replica gets disconnected from its leader instance, it will retry to establish the connection again and clone the master automatically.



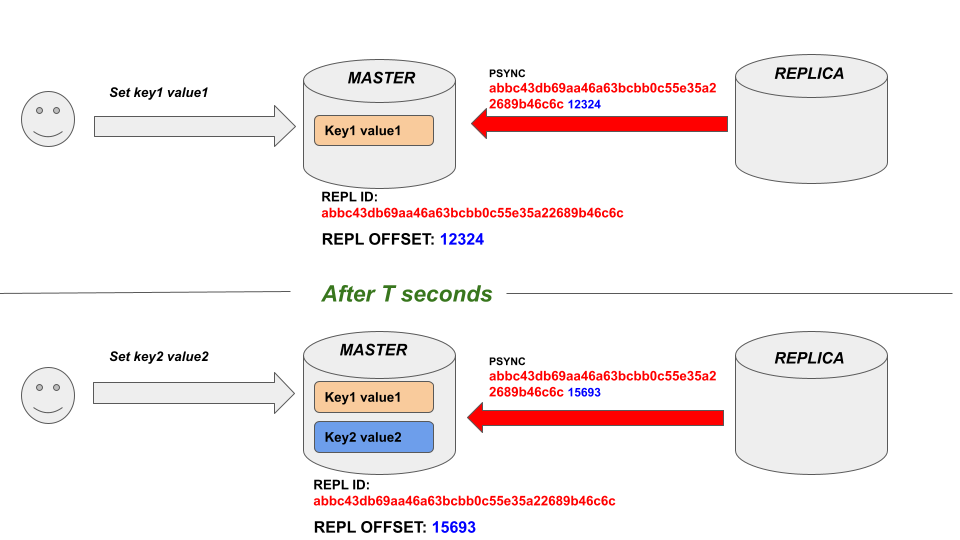
As you can see in the above illustration, when the master and replicas are well connected, the master sends all the write commands and the actions that change the master dataset to the replicas asynchronously. This process is low latency and performance-optimized.

Furthermore, in the event of a master-replica break, replicas will reconnect and try to acquire only the missed commands due to the disconnection. It is called partial synchronization. In some cases, partial sync is not possible. Hence, a full resynchronization will happen where the master instance generates a snapshot of its data, pushes it to the replicas, and sends all the commands that affect the dataset modifications.

# The PSYNC/SYNC Command

When replicas connect to a master, they use the PSYNC command to communicate with the master instance. In this partial synchronization process, the given replica sends the master replication ID and offset values to the master server. At the receiving end, these two pieces of information will be used to transmit the incremental changes of the dataset from the master to the replica instance.

In short, the master replication ID and offset represents an exact version of the master dataset. Every master node is assigned a unique pseudo-random string which is called replication ID. In addition, the master keeps an offset value that is incremented for each chunk of data produced to be pushed to the replicas.



As shown in the above illustration, since the master dataset has been changed due to the client's writes, the replication offset value has changed from 12324 to 15693. Hence, the replica calls the PSYNC command with the updated offset value.

## Syntax

The PSYNC command has the following syntax.

PSYNC *master\_replication\_id* *master\_replication\_offset*

***master\_replication\_id:*** This is the pseudo-random string generated which represents a given dataset.

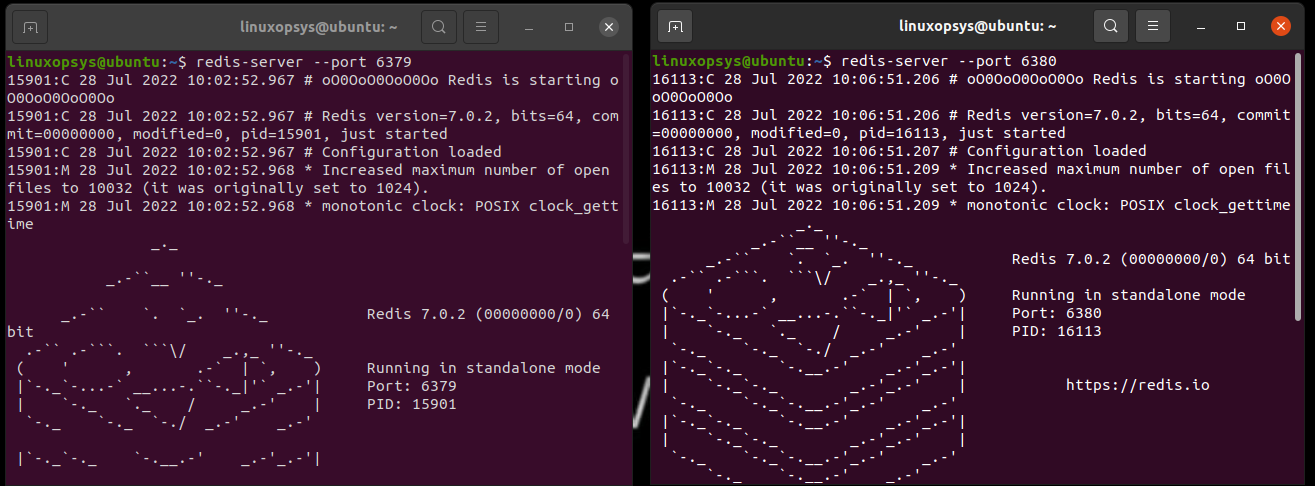
***master\_replication\_offset:*** The offset value which is incremented per each chunk of byte generated to send to the replicas.

This command returns a piece of data and writes from the master to a replica instance. In short, the PSYNC command starts a replication stream from the leader instance to the replica instance which transfers the dataset and client write requests changed from the previous offset.

## Use Case - Syncing Replica with Master

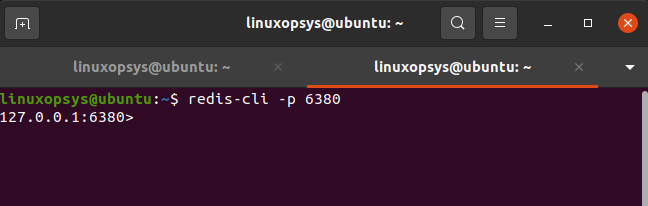
Let’s start with a couple of Redis server instances in different ports 6379 and 6380. You can use the following command.

| redis-server --port 6379  Redis-server --port 6380 |
| --- |

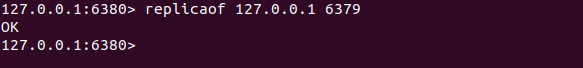


Next, we will convert one Redis instance to a replica using the REPLICAOF command. Before that, we need to connect to that instance with Redis-CLI as shown in the following.

| redis-cli -p 6380 |
| --- |



Now the connection has been made to the Redis server instance up and running in the 6380 port. Let’s make this instance a replica of the other Redis instance that runs at port 6379.



Now we can test whether this instance has become a replica by issuing the following command.



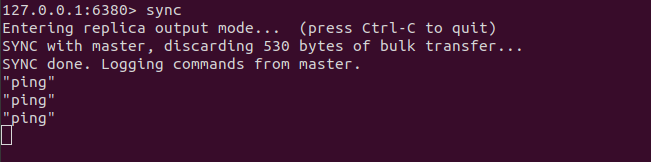
As you could see the role is displayed as a slave for this instance. Let’s run the same command against the other instance and inspect.



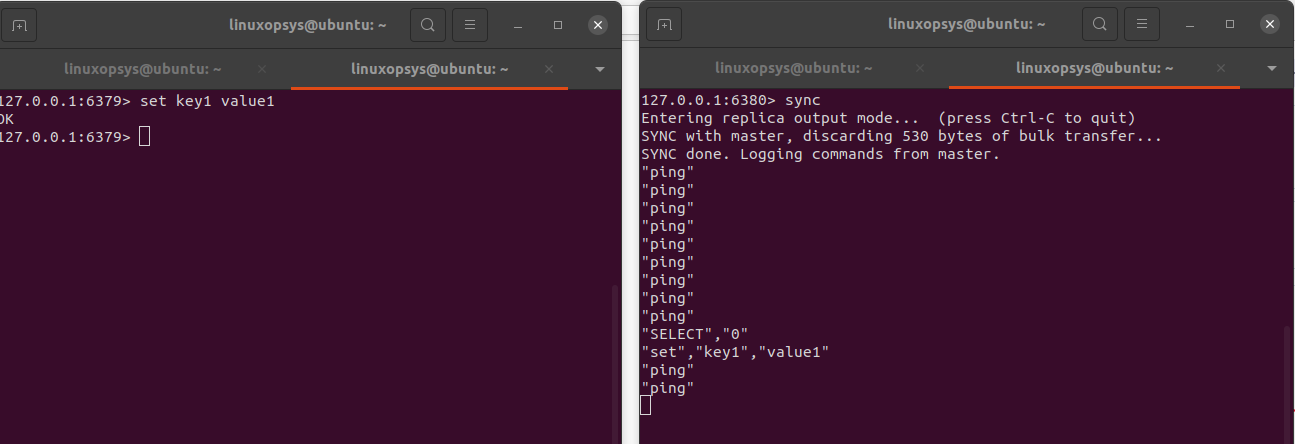
It is a master node as shown in the above output window.

Let’s closely inspect how the synchronization happens between master and slave instances. For demonstration purposes, the SYNC command is used. It behaves the same as the PSYNC command except that partial synchronization is not allowed.

Let’s issue the SYNC command from the replica.



Now, we will issue a write command to the master instance via the CLI.



As expected, the write command has been transferred to the replica immediately followed by PINGs. The PSYNC command works the same as the SYNC command but with partial synchronization enabled.

# Conclusion

In summary, Redis comes with different features that enable high availability. The master-replica pattern is one of the techniques supported by Redis to make sure at least one instance of the Redis server is available at a given time. Every replica is synchronized with its master and issues the PSYNC command to create a replication stream that transmits the changed dataset and writes requests from the master. As mentioned, the PSYNC command behaves the same as the old SYNC command but with partial synchronization enabled.