

Activity1: Redis Installation

1. Go to XpanAI and launch a terminal.
2. Type the following commands to install redis on your machine:

`apt-get update`

`apt-get install redis-server`

```
The following NEW packages will be installed:
  libjemalloc1 redis-server redis-tools
0 upgraded, 3 newly installed, 0 to remove and 67 not upgraded.
Need to get 519 kB of archives.
After this operation, 1507 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

[Type 'y' when prompted]

3. To start redis:

`redis-server`

```
root@expanai-xpanaiadmin-0:~# redis-server
410:C 15 May 08:19:58.464 # Warning: no config file specified, using the default config. In order to specify a config file
use redis-server /path/to/redis.conf

Redis 3.0.6 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 410

http://redis.io

410:M 15 May 08:19:58.465 # Server started, Redis version 3.0.6
410:M 15 May 08:19:58.466 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create
latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_huge
page/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restar
ted after THP is disabled.
410:M 15 May 08:19:58.466 * The server is now ready to accept connections on port 6379
```

4. Start a new terminal and type following command:

`redis-cli`

This will open a redis prompt.

```
127.0.0.1:6379>
```

In the above prompt, 127.0.0.1 is your machine's IP address and 6379 is the port on which Redis server is running.

5. Now type the following PING command.

`ping`

```
127.0.0.1:6379> ping
PONG
```

This shows that Redis is successfully installed on your machine.

Activity2: Strings

Redis string is a sequence of bytes. You can store anything up to 512 megabytes in one string.

1. To assign key "name" with a value of "practical8" using set command.

```
SET name practical8
```

2. To retrieve value of key "name" using get command.

```
GET name
```

3. To delete an entry using del command.

```
DEL name
```

4. The INCR command parses the string value as an integer, increments it by one, and sets the obtained value as the new value.

```
SET counter 100  
INCR counter
```

5. If you do not wish to use an extra line of code to create multiple entries, you can use the extended function mset.

```
MSET name1 john name2 mary name3 peter
```

6. Retrieve the values of several entries at once using mget.

```
MGET name1 name2 name3
```

Exercise

1. Assign following key-value pair.

Key	Value
Module1	English
Module2	Physics
Module3	Programming
Module5	Maths

2. Check if key "Module4" exists or not in the database using EXISTS command. It will return 1 if the key exists, or 0 if the key does not exist.
3. Check the value type stored at key "Module1" using TYPE command.
4. Rename the key "Module5" to "Module4" using RENAME command.
5. Get the first 5 character of key "Module3" using GETRANGE command.

Activity3: Key expiration

Key expiration lets you set a timeout for a key, also known as a “time to live”, or “TTL”. When time to live elapses, the key is automatically destroyed.

1. Set a key's expiration using EXPIRE command.

```
SET key user1
EXPIRE key 5
GET key
```

2. Wait for 5 seconds before you retrieve value of key. The key should have disappeared and return 'nil' now.

```
GET key
```

3. You can also create keys with expires. The following example sets a key with the string value user1, having an expire of ten seconds.

```
SET key user1 EX 10
```

Exercise

1. Set a key with string

Key	Value	Expire Time
Module1	English	10 seconds

2. Check the remaining time to live for the key.
(The command returns -2 if the key does not exist.
The command returns -1 if the key exists but has no associated expire.)

Activity4: Lists

Redis lists are linked lists of string values, sorted by insertion order. You can add elements in Redis lists in the head or the tail of the list.

1. Push two values (100, 101) to head of the list using LPUSH command.

```
LPUSH queue number 100
LPUSH queue number 101
```

2. Push two values (102, 103) to tail of the list using RPUSH command.

```
RPUSH queue number 102
RPUSH queue number 103
```

3. Display values within the range (0 to 10) using lrange command. Both the indexes can be negative, telling Redis to start counting from the end: so -1 is the last element, -2 is the penultimate element of the list, and so forth.

```
LRANGE queue number 0 10
```

4. Remove the value from head of the list (First In First Out)

```
RPOP queuenumber
```

5. Remove the value from tail of the list (First In Last Out)

```
LPOP queuenumber
```

Exercise

1. Create the following key-value pair.

Key	Value
queue	1,3,5,6,7,8

2. Remove values 1,3 and 8 from the list, so that the list now looks like this.

Key	Value
queue	5,6,7

3. Add values to the list, so that the list now looks like this.

Key	Value
queue	3,4,5,6,7,8,9

4. Check the length of the list.
5. Get the third element from the list.
6. Trims the list to contain only 2nd to 5th element.

Activity5: Sets

Redis Sets are an unordered collection of unique strings. Sets do not allow repetition of data in a key.

1. Add new members to a set using SADD command.

```
SADD myset 1 2 3
```

2. Remove the specified member from the set using SREM command.

```
SREM myset 2
```

3. Gets all the members in a set using SMEMBERS command.

```
SMEMBERS myset
```

Exercise

1. Create the following key-value pair.

Key	Value
colorset1	blue orange green
colorset2	red purple yellow

2. Test if “red” value belongs to set “colorset1”.
3. Check the size of the set “colorset2”.
4. Remove value “orange” from the set “colorset1”.
5. Combine “colorset2” and “colorset1” together.

Activity6: Hashes

Hashes are collection of key-value pairs. Every hash can store up to more than 4 billion field-value pairs.

1. Sets the value of one or more fields on a hash using HSET command.

```
HSET user1 username ryan
```

2. Returns the value at a given field using HGET command.

```
HGET user1 username
```

3. Sets the value of one or more fields on a hash using HMSET command.

```
HMSET user2 username nathan birthyear 2005 verified 1
```

4. Returns the values at one on more given fields using HMGET command.

```
HMGET user2 username birthyear verified
```

Exercise

1. Create the following key-value pair.

Key	Value
carspecs	brand: Toyota model: Alphard colour: silver year: 2020

2. Delete hash field “year:2020”.
3. Determine whether hash field “horsepower” exists or not.

Activity7: Sorted Sets

Redis Sorted Sets are similar to Redis Sets with the unique feature of values stored in a set. The difference is, every member of a Sorted Set is associated with a score, that is used in order to take the sorted set ordered, from the smallest to the greatest score.

1. To add members to a sorted set using ZADD command.

```
ZADD tutorials 1 redis
ZADD tutorials 2 mongodb
ZADD tutorials 3 mysql
```

2. You can also update the member value using ZADD command.

```
ZADD tutorials 4 mysql
```

3. To gets the number of members in the sorted sets using ZCARD command.

```
ZCARD tutorials
```

Exercise

1. Assign following key-value pair.

Key	Value	Score
leaderboard	user1	60
	user2	80
	user3	55
	user4	90
	user5	20

2. Get the members in sorted set “leaderboard”, ordered by scores in descending order.
3. How many members in sorted set “leaderboard’ have score between 60 and 90?
4. Increment the score of user3 by 20.
5. Remove user5 from the sorted set.
6. Remove users with score lower than 60 from the sorted set.
7. Get the score of user4 from the sorted set “leaderboard”.