Redis:

What is Redis

- Redis is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes with radius queries and streams. Redis has built-in replication, Lua scripting, LRU eviction, transactions and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster.

Re----di-----s

- Re ---→ Remote
- Di ----→ Dictionary
- S ----→ Server

- Redis is a KEY VALUE store : Redis can store data as key value pairs

- Name(Key) = John(Value)

Example :

- SET “name” “John”
- SET “age” “25”
- GET “name”

Features :

- It’s a NO-SQL Database
- Does not have tables/rows/columns/functions/procedures etc. like inn MySQL, Oracle DBs…
- Does not uses any statements like SELECT, INSERT, UPDATE, DELETE..

Features:

- Uses data structures to store data..
- String, Lists, Sets, Sets, Hashes.
- Bitmaps, Hyperloglos, Geopatial indexes

- Interaction with data is command based
- https://redis.io/commands
- Click Documentation
- Click full list of commands

Features:

- It is an in-memory database (with persistence options)
- It’s keeps the data in the memory in the cache and does not writes to the disk and it super fast.
- It has also the options the write the data to the disk.

Creator:

- Salvatore Sanfilippo (original developer)
- May 10, 2009 (first release)
- Cross Platform (written in ANSI C)
- Open source

Some benefits

- Performance
- Flexibility
- Durability
- Language Support
- Compatibility
- Scaling

Performance

- It is super fast
- https://redis.io/topics/benchmarks
- Redis -> Documentstion -> Benchmarks -> Requests per second

Simple and Flexible

- No need of defining any tables, rows, columns.
- No insert, selects , update, delete
- Simple and Straight-forward data read / write

Redis is Durable

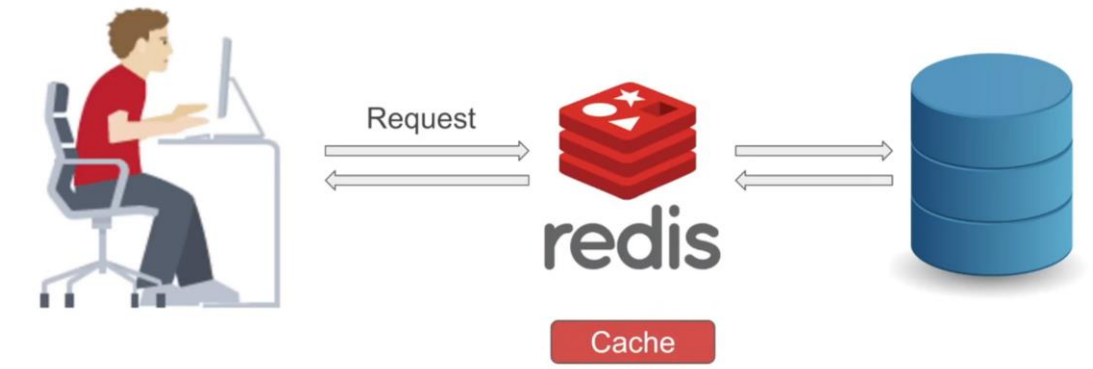
- Has option to write data on disk
- Data write options are configurable
- Can be used as a Caching System OR A full fledged Database.

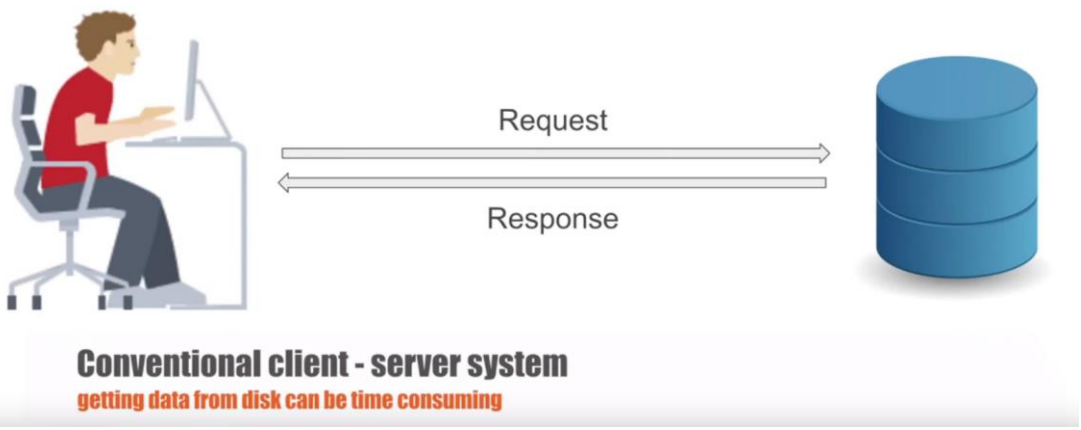
Multi Languages and Platform Support

- https://redis.io/clients
- Redis -> clients

Compatibility

- Can be used as 2nd Database to make transaction faster…
- It can be used along with the existing database


Request
redis
Cache


Request
Response
Conventional client - server system
getting data from disk can be time consuming

Request

Service the request with data available in cache

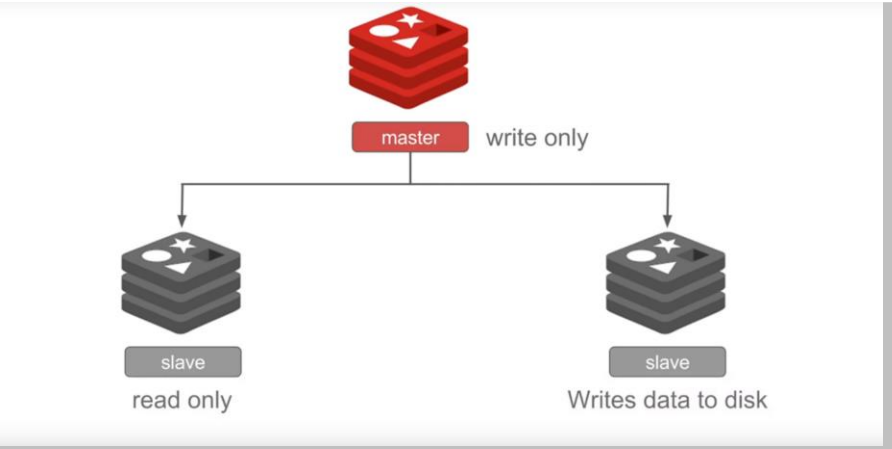
redis

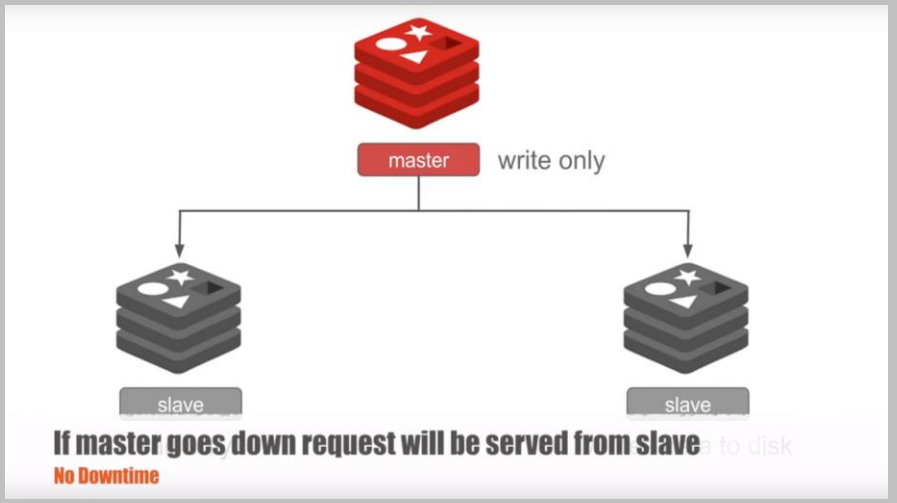
Cache

Get the data from db if data not available in cache

Scaling..

- Redis has master-slave Replication feature
- Performance Optimization
- No Downtime





Some other features

- Has a single text file for all configurations
- Is single Threaded – one action at a time
- Pipelining: You can cluster multiple commands and send them at once.

Who uses Redis…

- Flickr
- Stackoverflow
- Github
- Pinterest
- Twitter
- Docker
- https://redis.io/topics/whos-using-redis

Install Redis in Windows..

- Download redis from :
- https://github.com/MSOpenTech/redis/releases
- https://github.com/MicrosoftArchive/redis/releases

Install Redis in Windows..

- Download redis Zip file.
- Unzip the file
- Click on redis-server.exe
- Click on redis-cli.exe
- Run commands
- Set variable path
- Run it from cmd

Test:

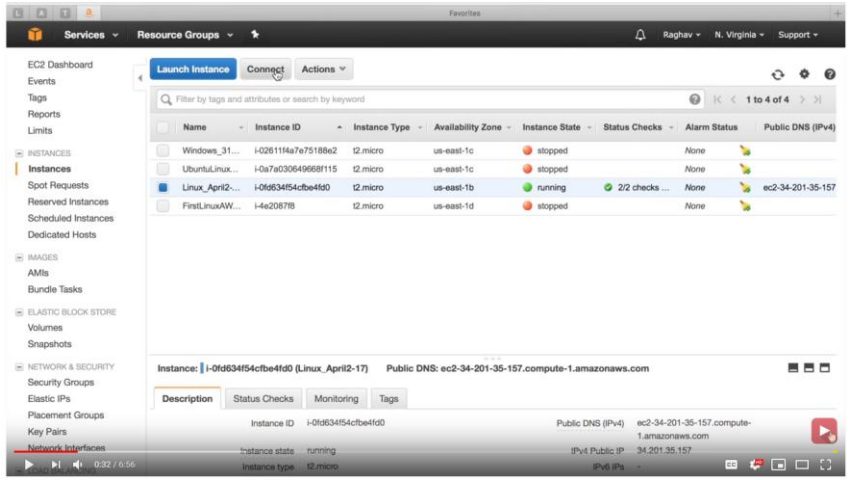
- Ping <enter>
- Ping “Hello World..” <enter>
- Set name “ Rozer “
- Get name

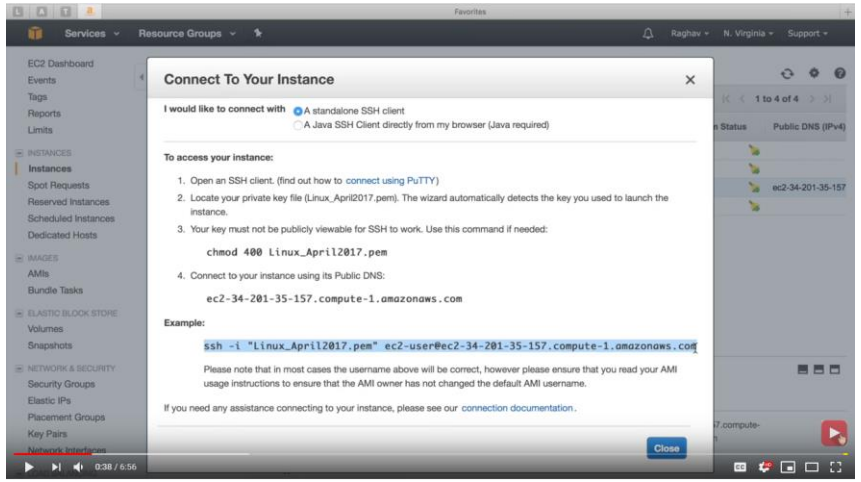
Stop Server :

- How to stop redis server :
- shutdown SAVE
- Ctrl + C

Connect to Linux System :

- AWS Linux ec2
- Go to Terminal and connect
- Make new folder called redis (mkdir)
- Enter the folder
- Run the commands
- Run as super user (sudo)
- Sudo yum groupinstall ‘Development Tools’
- Sudo yum install gcc make
- Sudo make
- Sudo make tets





Connect to Linux System :

- Src/redis-server
- Src/redis-cli
- Sudo make test install
- Redis-server (start redis)
- Go to amazon terminal
- Redis-cli
- Run ping for test
- Shutdown SAVE

Redis Desktop Manager

- GUI Client for Redis
- GUI Client enables interaction with redis via a user interface
- Redis Desktop Manager is a desktop gui client (web based gui clients are also available)
- Open Source | Cross Platform

How to use Redis Desktop manager

- Search on web redis desktop manager
- https://redisdesktop.com/
- Documentation
- https://redisdesktopmanager.readthedocs.io/en/0.9/install/
- Download(0.8.8)
- https://sourceforge.net/projects/redis-desktop-manager.mirror/files/0.8.8/
- Install as any other application
- Launch from icon in Desktop
- Connect to redis server

Name: my redis

Host : localhost

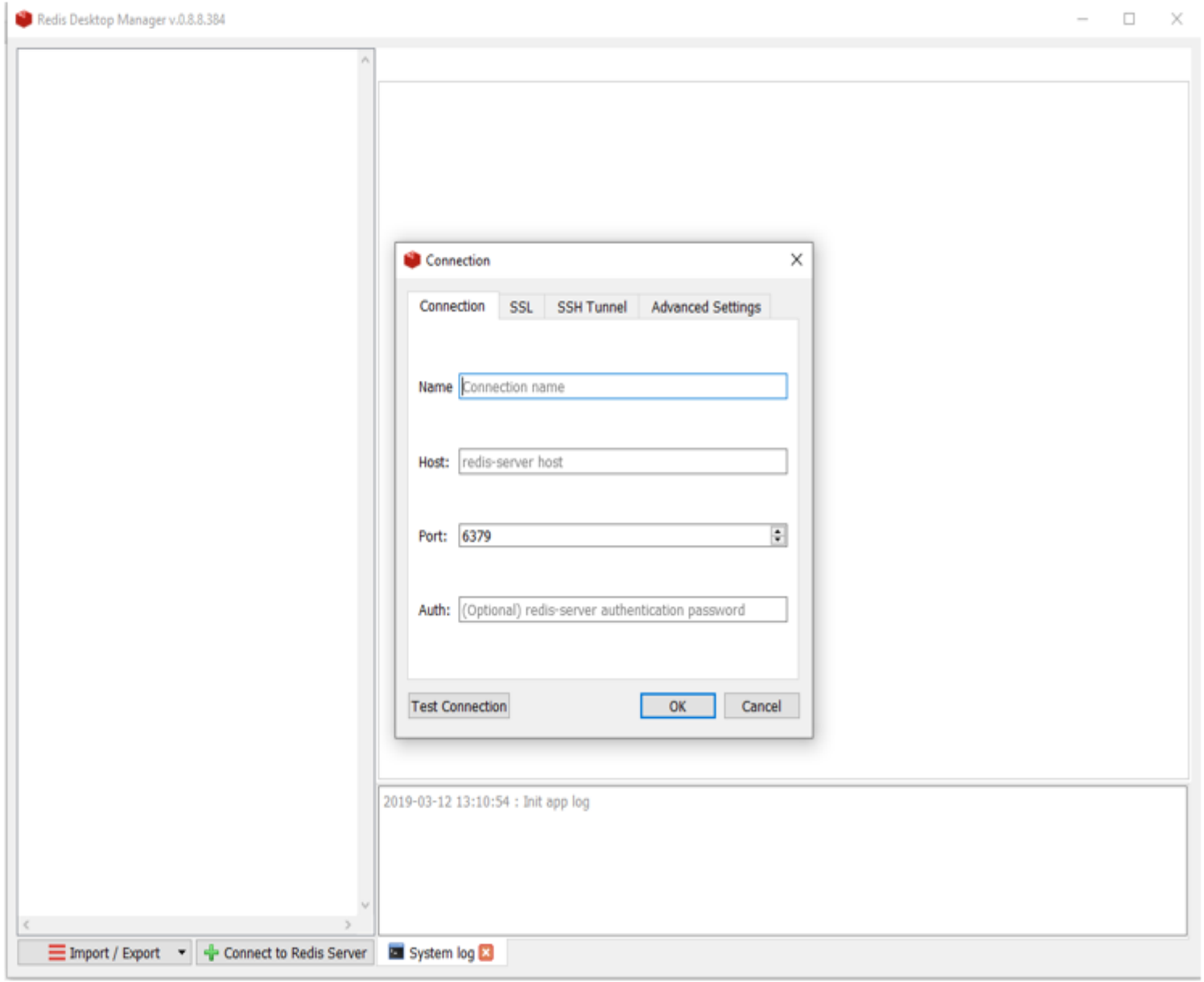
Click on Test Connection

Click on OK Button

Click on name on the left side box

Expand the box

Change value directly



Run Commands on the Remote Server:

- To run commands on Redis remote server, you need to connect to the server by the same client **redis-cli.**
- $ redis-cli -h host -p port -a password (syntax)
- Example:
- $redis-cli -h 127.0.0.1 -p 6379 -a "mypass"
- redis 127.0.0.1:6379>
- redis 127.0.0.1:6379> PING
- PONG

Redis keys:

- Redis 127.0.0.1:6379> COMMAND KEY_NAME (syntax)
- redis 127.0.0.1:6379> SET Innopolis redis
- OK
- redis 127.0.0.1:6379> DEL Innopolis
- (integer) 1

Redis String:

- redis 127.0.0.1:6379> COMMAND KEY_NAME (syntax)
- redis 127.0.0.1:6379> SET innopolis redis
- OK
- redis 127.0.0.1:6379> GET innopolis
- "redis"

- **SET key value** : This command sets the value at the specified key.
- **GET key** : Gets the value of a key.

Redis List:

- Redis Lists are simply lists of strings, sorted by insertion order. You can add elements in Redis lists in the head or the tail of the list.
- Maximum length of a list is 2^{32} - 1 elements (4294967295, more than 4 billion of elements per list).
- redis 127.0.0.1:6379> LPUSH tutorials redis
- (integer) 1
- redis 127.0.0.1:6379> LPUSH tutorials mongodb
- (integer) 2
- redis 127.0.0.1:6379> LPUSH tutorials mysql
- (integer) 3
- redis 127.0.0.1:6379> LRANGE tutorials 0 10
- 1) "mysql"
- 2) "mongodb"
- 3) "redis"

Redis sets:

- Redis Sets are an unordered collection of unique strings. Unique means sets does not allow repetition of data in a key.
- redis 127.0.0.1:6379> SADD tutorials redis
- (integer) 1
- redis 127.0.0.1:6379> SADD tutorials mongodb
- (integer) 1
- redis 127.0.0.1:6379> SADD tutorials mysql
- (integer) 1
- redis 127.0.0.1:6379> SADD tutorials mysql
- (integer) 0
- redis 127.0.0.1:6379> SMEMBERS tutorials
- 1) "mysql"
- 2) "mongodb"
- 3) "redis"

Using Redis with Python:

- https://redislabs.com/lp/python-redis/

Thank you!