Title: Redis Configuration and Implementation

Objective: To implement different Redis configurations and demonstrate their functionality using the Redis CLI or Redis Telnet CLI.

Scenario: You have been hired as Software architect for a company that provides online gaming services. The company wants to use Redis as one of the databases to store relevant information, game statistics, and other related data. You are tasked with setting up Redis in different configurations to meet the company's requirements.

As this is a study point assignment you are to choose and setup minimum 2 out of the 5 configurations. as well as develop 1 simple application, web site or API that will make use of minimum 1 out of the chosen Redis configurations, minimal requirements for that development is to ensure that you can do all crud operations up against redis.
Whenever the configuration mention more than one computer see it as optional you should be able to use Docker or VM's to simulate this.

Configuration 1: Redis with Retention Policy

Objective: To demonstrate how to implement a Retention Policy in Redis.

Scenario: The company wants to store user data in Redis for a limited period. User data should be automatically deleted after a certain period of time to free up memory. You need to implement a retention policy to automatically delete expired keys from Redis. This is usually useful when caching data or calls between an API and a website.

Tasks:

1. Install Redis on your machine and configure it to run on the default port (6379).
2. Use the Redis CLI or Redis Telnet CLI to set up a retention policy that automatically deletes keys after 24 hours.
3. Use the Redis CLI or Redis Telnet CLI to store user data in Redis.
4. Wait for 24 hours and verify that expired keys are automatically deleted from Redis.

Configuration 2: Redis Master-Slave Replication

Objective: To demonstrate how to set up Master-Slave replication in Redis.

Scenario: The company wants to ensure that user data is always available and can handle high traffic. You need to set up Redis in a Master-Slave configuration to ensure high availability and reliability.

Tasks:

1. Install Redis on two different machines and configure them to run on different ports.
2. Use the Redis CLI or Redis Telnet CLI to set up a Master-Slave replication configuration between the two Redis instances.

3. Use the Redis CLI or Redis Telnet CLI to store user data in the Master Redis instance.
4. Verify that the Slave Redis instance is replicating data from the Master instance.
5. Test the configuration by stopping the Master Redis instance and verifying that the Slave Redis instance can handle requests.

Configuration 3: Redis Cluster

Objective: To demonstrate how to set up Redis in a Cluster configuration.

Scenario: The company wants to scale its Redis database to handle a large amount of data and traffic. You need to set up Redis in a Cluster configuration to ensure scalability and high performance.

Tasks:

1. Install Redis on several machines and configure them to run on different ports.
2. Use the Redis CLI or Redis Telnet CLI to set up a Redis Cluster configuration.
3. Use the Redis CLI or Redis Telnet CLI to store user data in the Redis Cluster.
4. Verify that data is distributed evenly across the Redis Cluster.
5. Test the configuration by stopping one of the Redis instances and verifying that the Redis Cluster can handle requests.

Configuration 4: Redis Security

Objective: To demonstrate how to set up security features in Redis.

Scenario: The company wants to ensure that user data is secure and can only be accessed by authorized users. You need to set up security features in Redis to protect user data from unauthorized access.

Tasks:

1. Install Redis on your machine and configure it to run on the default port (6379).
2. Use the Redis CLI or Redis Telnet CLI to set up a password for Redis.
3. Use the Redis CLI or Redis Telnet CLI to store user data in Redis.
4. Verify that Redis requires a password to access user data.
5. Use the Redis CLI or Redis Telnet CLI to set up access control lists (ACLs) to control access to Redis commands and data.

Configuration 5: Redis Publish-Subscribe Pattern

Objective: To demonstrate how to use the Publish-Subscribe pattern in Redis.

Scenario: The company wants to implement a notification system that sends updates to users when their friends are playing games online. You need to implement the Publish-Subscribe pattern in Redis to enable real-time updates for users.

Tasks:

1. Install Redis on your machine and configure it to run on the default port (6379).

2. Use the Redis CLI or Redis Telnet CLI to set up a Publish-Subscribe pattern in Redis.
3. Use the Redis CLI or Redis Telnet CLI to subscribe to a channel that sends real-time updates for users.
4. Use the Redis CLI or Redis Telnet CLI to publish a message to the channel.
5. Verify that the message is received by all subscribers to the channel.

Configuration 6: Redis Bloom Filters

Objective: To demonstrate how to use Bloom Filters in Redis.

Scenario: The company wants to implement a system that can quickly check whether a user is a registered user or not. You need to implement Bloom Filters in Redis to enable fast membership testing for users.

Tasks:

1. Install Redis on your machine and configure it to run on the default port (6379).
2. Use the Redis CLI or Redis Telnet CLI to set up a Bloom Filter in Redis.
3. Use the Redis CLI or Redis Telnet CLI to add some user IDs to the Bloom Filter.
4. Use the Redis CLI or Redis Telnet CLI to test if a user ID is a member of the Bloom Filter.
5. Verify that the Bloom Filter returns false positives but no false negatives.

Note: Remember to document your steps and findings in a report and include any challenges faced during the implementation process. You may also include any recommendations or improvements for the Redis configurations you have implemented.