

[Explore GfG Courses](#)[Share Your Experiences](#)

Introduction to Redis

Redis Basics

[Redis and its role in System Design](#)[How does Redis store data?](#)[Complete Tutorial of Configuration in Redis](#)[Redis Cache](#)

Redis Data Structures

[Complete Guide to Redis Lists](#)[Complete Guide on Redis Strings](#)[A Complete Guide to Redis Keys](#)[Complete tutorial on Sets in Redis](#)[Complete tutorial on HyperLogLog in redis](#)[A Complete Guide to Redis Hashes](#)[Complete tutorial on Sorted Sets in Redis](#)

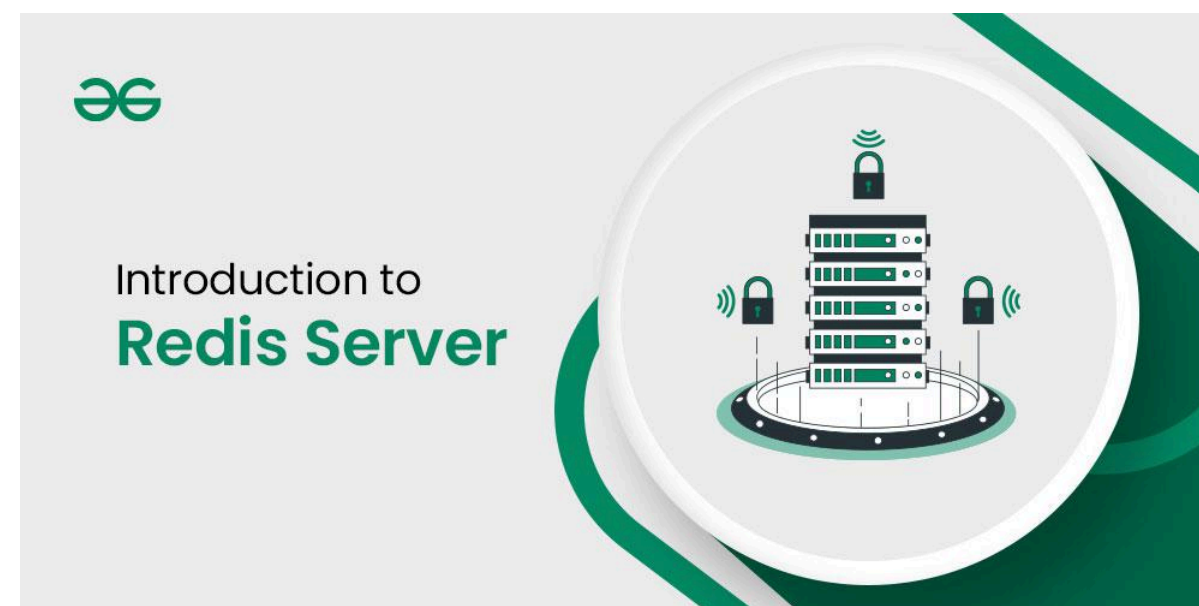
Redis Commands

Introduction to Redis

Last Updated : 12 Jan, 2024

Redis is an in-memory data structure that is used for faster access to data. It is used to store data that needs to be accessed frequently and fast. It is not used for storing large amounts of data. If you want to store and retrieve large amounts of data you need to use a traditional database such as MongoDB or MYSQL. Redis provides a variety of data structures such as sets, strings, hashes, and lists.

- The Redis server is a program that runs and stores data in memory.
- You can just connect to that server and can use it to store and retrieve data faster.
- For that reason, Redis is not used for persistent storing of data as complete data will be lost if the system crashes.
- Redis is scalable as you can run multiple instances of the server.
- It is often used as a cache that stores data temporarily and provides faster access to frequently used data.



Important Topics for Redis Server

- [When to use Redis Server?](#)
- [Advantages of Redis Server](#)
- [Disadvantages of Redis Server](#)



- [How to Start Redis Server?](#)
- [Redis Basics](#)
- [Redis Data Structures](#)
- [Redis Commands](#)
- [Redis Advanced](#)

When to use Redis Server?

Consider you have a MySQL database and you are constantly querying the database which reads the data from the secondary storage, computes the result, and returns the result.

If the data in the database is not changing much you can just store the results of the query in redis-server and then instead of querying the database which is going to take 100-1000 milliseconds, you can just check whether the result of the query is already available in redis or not and return it result which is going to be much faster as it is already available in the memory.

Note: In a messaging app, Redis can be used to store the last five messages that the user has sent and received using the built-list data structure provided in Redis.

Advantages of Redis Server

1. High Performance:

- Redis excels in terms of performance due to its in-memory nature. It can deliver extremely fast read and write operations, making it suitable for scenarios where low-latency is critical.

2. Simple and Easy-to-Use API:

- Redis has a straightforward API that consists of simple and intuitive commands, making it easy for developers to use and integrate into their applications.

3. Data Structures:

- Redis supports a variety of data structures, including strings, lists, sets, hashes, and more. This versatility allows developers to model their data more effectively, choosing the right data structure for the task at hand.

4. Atomic Operations:

- Redis supports atomic operations on these data structures, making it a great fit for scenarios that require consistency and reliability in multi-step operations.

5. Persistence Options:

- While Redis is an in-memory database, it provides persistence options such as snapshots and append-only files. This allows users to configure the level of durability needed for their specific use case.

6. Replication and High Availability:

- Redis supports master-slave replication, enabling the creation of replicas of the master server. This provides high availability and fault tolerance in case the master node fails.

Disadvantages of Redis Server

- Persistence Mechanism Complexity:

- Redis is an in-memory database, and while it supports persistence, the mechanisms for achieving this (such as snapshots and append-only files) can be complex and may impact performance.

- Limited Query Capability:

- Redis is not a full-fledged relational database and lacks the complex querying capabilities of traditional databases. It primarily operates on key-value pairs and offers basic data structures like strings, lists, sets, and hashes.

- Memory Usage:

available system memory. Large datasets may require significant memory resources, which can be a potential constraint.

- Single-Threaded Nature:

- Redis traditionally uses a single-threaded event-loop architecture. While this design simplifies certain aspects of the system, it may limit performance on multi-core systems. However, recent versions of Redis have introduced multi-threading in some parts to address this limitation.

- No Built-in Security Features:

- Redis initially lacked built-in security features, and it was recommended to be run in trusted environments.
- While newer versions include authentication mechanisms, it's essential to configure and manage these security features properly.

How to Start Redis Server?

To start the Redis server on your machine you need to run the below command in the terminal :

```
redis-server
```

The above command will start the server on port 6379 which is the default port for the Redis server.

You can connect to the redis-server and use it to retrieve and store data using the below command.

```
redis-cli
```

Redis Basics

[Redis and its role in System Design](#)

[How does Redis store data?](#)

[Complete Tutorial of Configuration in Redis](#)

[Redis Cache](#)

Redis Data Structures

[Complete Guide to Redis Lists](#)

[Complete Guide on Redis Strings](#)

[A Complete Guide to Redis Keys](#)

[Complete tutorial on Sets in Redis](#)

[Complete tutorial on HyperLogLog in redis](#)

[A Complete Guide to Redis Hashes](#)

[Complete tutorial on Sorted Sets in Redis](#)

Redis Commands

[Complete Guide to Redis Commands](#)

[Complete Guide of Redis Scripting](#)

[Redis Connections - Syntax, Command Commands & Examples](#)

[Redis - Client Connection](#)

[Complete Guide on Redis Data Types with Commands and Storage](#)

[Redis | Publish Subscribe](#)

[Difference between Redis Pub/sub vs Redis streams publish](#)

[Complete tutorial of Transactions in Redis](#)

Redis Advanced

- [Understanding Redis Partitioning](#)
- [Complete Guide to Redis Pipelining](#)
- [Complete Guide for Redis Benchmark](#)
- [Complete Guide to Redis PHP](#)
- [Complete Guide to Redis Java](#)
- [Complete tutorial on Backup in Redis](#)
- [Complete tutorial on security in Redis](#)

[Comment](#)

[More info](#)

[Advertise with us](#)

Next Article

[Redis and its role in System Design](#)

Similar Reads

- ### Redis and its role in System Design

Redis is an open-source, in-memory data structure store used as a database, cache, and message broker. It is widely used for its fast performance, flexibility, and ease of...

15+ min read
- ### Redis Cache

As we all know caching refers to storing frequently used or dealt-up data in temporary high-speed storage to reduce the latency of a system. Inside a Redis Cluster, there...

15+ min read
- ### Complete Tutorial of Configuration in Redis

Redis configuration involves setting various options and parameters that govern the behavior of the Redis server. Configuration settings impact aspects such as networking,...

15+ min read
- ### Complete Guide to Redis PHP

Redis, an acronym for "Remote Dictionary Server," is an open-source, in-memory data structure store that can be used as a caching mechanism, message broker, and data store...

15+ min read

Spring Boot - CRUD Operations Using Redis Database

Redis is an in-memory data structure that is used for faster access to data. It is used to store data that needs to be accessed frequently and fast. It is not used for sto...

15+ min read

Redis vs Kafka

Redis and Kafka are two essential components in modern system design, but they serve different purposes. Redis is primarily used for caching and real-time data processing,...

14 min read

Memcached vs. Redis

Memcached and Redis are two of the most popular in-memory data stores used for caching and improving the performance of web applications. While both serve similar purposes...

11 min read

Caching Strategies for API

The article explains how to improve the performance and efficiency of APIs using caching. Caching is a technique where frequently accessed data is stored temporarily to re...

15+ min read

Introduction to Spring Security and its Features


Spring Security is a powerful authentication and authorization framework used to secure Java-based web applications. It easily integrates with Spring Boot and provides adv...

15+ min read

Types of Cache

Cache plays an important role in enhancing system performance by temporarily storing frequently accessed data and reducing latency. Understanding the various types of cach...



15+ min read



GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:
A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)

Registered Address:
K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305



Advertise with us

Company

- About Us
- Legal
- Privacy Policy
- Careers
- In Media
- Contact Us
- GfG Corporate Solution
- Placement Training Program

Python Tutorial

- Python Programming
- Examples
- Django Tutorial
- Python Projects
- Python Tkinter
- Web Scraping
- OpenCV Tutorial
- Python Interview Question

Preparation Corner

- Company-Wise Recruitment Process
- Aptitude Preparation
- Puzzles
- Company-Wise Preparation

Explore

- Job-A-Thon Hiring Challenge
- GfG Weekly Contest
- Offline Classroom Program
- DSA in JAVA/C++
- Master System Design
- Master CP
- GeeksforGeeks Videos

Computer Science

- GATE CS Notes
- Operating Systems
- Computer Network
- Database Management
- System
- Software Engineering
- Digital Logic Design
- Engineering Maths

More Tutorials

- Software Development
- Software Testing
- Product Management
- Project Management
- Linux
- Excel
- All Cheat Sheets

Languages

- Python
- Java
- C++
- PHP
- GoLang
- SQL
- R Language
- Android Tutorial

DevOps

- Git
- AWS
- Docker
- Kubernetes
- Azure
- GCP
- DevOps Roadmap

Machine Learning/Data Science

- Complete Machine Learning & Data Science Program - [LIVE]
- Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
- Data Science Training Program - [LIVE]
- Data Science Course with IBM Certification

DSA

- Data Structures
- Algorithms
- DSA for Beginners
- Basic DSA Problems
- DSA Roadmap
- DSA Interview Questions
- Competitive Programming

System Design

- High Level Design
- Low Level Design
- UML Diagrams
- Interview Guide
- Design Patterns
- OOAD
- System Design Bootcamp
- Interview Questions

Programming Languages

- C Programming with Data Structures
- C++ Programming Course
- Java Programming Course
- Python Full Course

Data Science & ML

- Data Science With Python
- Data Science For Beginner
- Machine Learning
- ML Maths
- Data Visualisation
- Pandas
- NumPy
- NLP
- Deep Learning

School Subjects

- Mathematics
- Physics
- Chemistry
- Biology
- Social Science
- English Grammar

Clouds/Devops

- DevOps Engineering
- AWS Solutions Architect Certification
- Salesforce Certified Administrator Course

Web Technologies

- HTML
- CSS
- JavaScript
- TypeScript
- ReactJS
- NextJS
- NodeJs
- Bootstrap
- Tailwind CSS

Databases

- SQL
- MYSQL
- PostgreSQL
- PL/SQL
- MongoDB

GATE 2026

- GATE CS Rank Booster
- GATE DA Rank Booster
- GATE CS & IT Course - 2026
- GATE DA Course 2026
- GATE Rank Predictor