Overview of Memorystore for Redis

Memorystore for Redis provides a fully-managed service that is powered by the Redis in-memory data store to build application caches that provide sub-millisecond data access.

Memorystore for Redis offers several advantages over self-managed Redis:

- **Deploy what fits your needs.** Memorystore for Redis allows you the flexibility to choose from different service tiers and sizes that fit your performance and operational needs. With a few clicks, you have the option to deploy a Basic Tier standalone Redis instance or a Standard Tier high availability Redis instance up to 300 GB.
- **Easily scale to get blazing speed.** With Memorystore for Redis, you can easily achieve your latency and throughput targets by scaling up your Redis instances with minimal impact to your application's availability. Start with the lowest tier and smallest size, then grow your Redis instance as the needs of your application change. For applications that need scaling of read queries, you can scale the queries across five read replicas (/memorystore/docs/redis/read-replicas) using the read endpoint.
- **Highly available and more secure**. Redis instances are protected from the internet using private IPs and are further secured using Identity and Access Management role-based access control and in-transit encryption. Standard high availability instances provide up to five replicas replicated across zones and provide a 99.9% availability SLA.
- **Focus on your application.** Memorystore for Redis automates the complex operational tasks that are required to deploy and manage Redis. Tasks like provisioning, replication, failover, and monitoring are all automated. Applications connect to a single endpoint, eliminating any node-discovery logic and simplifying management and operations. Additionally, integration with Cloud Monitoring makes it easy to monitor your Redis instances.
- **Redis Protocol Compatible.** Memorystore for Redis is fully Redis protocol compliant. You can move your applications using open source Redis to use Memorystore for Redis without any code changes. There is no need to learn new tools: all existing tools and client libraries just work.

What it's good for

Memorystore for Redis provides a fast, in-memory store for use cases that require fast, real-time processing of data. From simple caching use cases to real time analytics, Memorystore for Redis provides the performance you need.

- **Caching:** Cache is an integral part of modern application architectures. Memorystore for Redis provides low latency access and high throughput for heavily accessed data, compared to accessing the data from a disk based backend store. Session management, frequently accessed queries, scripts, and pages are common examples of caching.
- **Gaming:** Gaming is about capturing and keeping the user's attention. One key aspect that keeps users hooked on a game is the leaderboard. Everyone wants to see how they are progressing and where they stand. Making this experience snappy is critical, and with its in-memory store and data structure like Sorted Set, Memorystore for Redis makes it easy to maintain a sorted list of scores while providing uniqueness of elements. Player Profile is another piece of information that can be accessed frequently. Redis hash makes it fast and easy to store and access profile data.
- **Stream Processing:** Whether processing a Twitter feed or stream of data from IoT devices, Memorystore for Redis is a perfect fit for streaming solutions. Combined with Dataflow, Memorystore for Redis provides a scalable, fast in-memory store for storing intermediate data that thousands of clients can access with very low latency.

Features

The following table describes the features that Memorystore for Redis provides:

- **Fully managed:** Deploying and maintaining a Redis instance can be time consuming. Memorystore for Redis provides patching, 24x7 threat monitoring, failure detection, and automatic failover, allowing you to spend more time on building your applications.
- **Simple deployment:** Easily deploy a single instance or highly available Redis instance by choosing Basic Tier or Standard Tier (/memorystore/docs/redis/redis-tiers) from the Google Cloud console or by automating deployment using Google Cloud CLI or Cloud Client Libraries.
- **High availability:** Memorystore for Redis instances in the Standard Tier are replicated across zones, monitored for health and have fast automatic failover. Standard Tier instances also provide an SLA of 99.9%. For more information, see High Availability (/memorystore/docs/redis/high-availability).

- **Enterprise-grade security:** Redis instances are protected from the internet using private IPs, and access to instances is controlled and limited to applications running on the same authorized Virtual Private Cloud as the Redis instance.

  Instances are further secured using IAM roles (/memorystore/docs/redis/access-control), which enables granular control over who can manage and access the Redis instance. Also, Memorystore for Redis is integrated with Cloud Audit Logging (/logging/docs/audit), enabling you to answer "who did what and when".

- **Scale, as needed:** Memorystore for Redis enables scaling your instances (/memorystore/docs/redis/scaling-instances) up to a maximum of 300 GB and supports up to 16 Gbps of network throughput. With the ability to scale instance size seamlessly, you can start small and increase the size of the instance as needed. Read replicas (/memorystore/docs/redis/read-replicas) allow scaling read queries across five replicas When scaling a Standard Tier instance, your applications experience downtime of less than a minute.

- **Monitoring:** Redis metrics are available through Monitoring, making it easy to monitor your Redis instances (/memorystore/docs/redis/monitoring-instances). Using Cloud Logging, you can also see the Redis logs for your instance.

- **On-demand billing:** With Memorystore for Redis instances, you are billed by the hour for the capacity (GB) (/memorystore/pricing) that you provision. If your instances are used for a few minutes, you are billed for only those minutes.

- **Redis versions 6.x, 5.0, 4.0, and 3.2:** The versions are always kept up to date with the latest critical patches, ensuring your Redis instances are secure. For more information on the versions and patch levels see the Memorystore for Redis Release Notes (/memorystore/docs/redis/release-notes) and Supported versions (/memorystore/docs/redis/supported-versions).

Connecting to a Memorystore for Redis instance

You can use any standard Redis client on the following environments to connect to your Memorystore for Redis instance (/memorystore/docs/redis/connecting-redis-instance):

- App Engine flexible environment
- App Engine standard environment
- Compute Engine VM instances
- Cloud Functions

- Cloud Run
- Google Kubernetes Engine clusters

Some serverless environments (/memorystore/docs/redis/supported-environments#serverless-environments-that-need-a-serverless-vpc-access-connector) require a Serverless VPC Access connector as a prerequisite for connectivity with Memorystore for Redis. See Serverless VPC Access connector requirement (/memorystore/docs/redis/supported-environments#serverless-vpc-access-connector-requirement) for more details.

In order to connect to your instance, your client must be connected to the same network as your Memorystore for Redis instance. See the Memorystore for Redis Networking (/memorystore/docs/redis/networking) page for more details.

Differences between managed and open source Redis

Memorystore for Redis is Redis protocol compliant. You can move your applications that use open source Redis to Google Cloud without any code changes.

In general, the Redis functionality that Memorystore for Redis instances provide is the same as the functionality that locally-hosted Redis instances provide. Some key differences are as follows:

- Memorystore for Redis does not currently support Redis Clustering.
- Open source Redis provides multiple options for persisting data to disk (https://redis.io/topics/persistence). RDB persistence enables point in time snapshots of the dataset and AOF persistence logs every write operation received by the server. Memorystore for Redis currently supports RDB snapshots (/memorystore/docs/redis/rdb-snapshots) and exporting data (/memorystore/docs/redis/import-export-overview). Memorystore for Redis does not support AOF persistence.
- Most parameters are preconfigured for Memorystore for Redis instances, and you cannot change them. Other parameters you configure when you set up your Memorystore for Redis instance. For more information, see Redis configurations (/memorystore/docs/redis/redis-configs).
- Some commands can interfere with a managed Redis service and are blocked. For more information, see the list of blocked Redis commands

(/memorystore/docs/redis/product-constraints#blocked_redis_commands).

Tier capabilities

The following table describes the differences between the available Memorystore for Redis service tiers:

Specifications	Basic Tier	Standard Tier (read replicas disabled)	Standard Tier (read replicas enabled)
Description	Provides a cache with no replication	Provides redundancy and availability using replication	Provides redundancy and availability using replication to backup data, and multiple Read replicas to increase read throughput
Max Redis node size	300 GB	300 GB	300 GB
Max network bandwidth	16 Gbps	16 Gbps	16 Gbps
I/O threads	Yes[1]	Yes[1]	Yes[1]
Scale node size	Yes	Yes	Yes
Scale number of read replica nodes	No	No	Yes
Cross-zone replication	No	Yes	Yes
Automatic failover	No	Yes	Yes
Read replicas	No	No	Yes
In-transit encryption	Yes	Yes	Yes
Maintenance window	Yes	Yes	Yes
Cloud Monitoring	Yes	Yes	Yes

[1] I/O threads are only available on M3 instances or higher running Redis version 6.x. For more information, see Redis version 6.x (/memorystore/docs/redis/supported-versions#redis_version_6x).

What's next

- For an introduction to Redis, see Introduction to Redis (https://redis.io/topics/introduction) on the Redis website.
- Quickstart using the Google Cloud CLI (/memorystore/docs/redis/create-instance-gcloud)
- Quickstart using the Google Cloud console (/memorystore/docs/redis/create-instance-console)

Last updated 2022-09-28 UTC.