Planning your Cluster Deployment

The information in the section discusses important topics you need to know about when planning your Redis Enterprise Software cluster for a production deployment.

Active-Active Geo-Distributed Redis

In Redis Enterprise, active-active geo-distribution is based on CRDT technology. The Redis Enterprise implementation of CRDT is called an Active-Active database (formerly known as CRDB). With Active-Active databases, applications can read and write to the same data set from different geographical locations seamlessly and with latency less than one millisecond (ms), without changing the way the application connects to the database. Active-Active databases also provide disaster recovery and accelerated data read-access for geographically distributed users.

Active-Passive Geo-Distributed Redis (Replica-Of)

In Redis Enterprise, active-passive geo-distribution provides applications read-only access to replicas of the data set from different geographical locations. The Redis Enterprise implementation of active-passive replication is called Replica Of. In Replica Of, an administrator designates a database as a replica (destination) of one or more databases (sources). After the initial data load from source to destination is completed, all write commands are synchronized from the sources to the destination. Replica Of lets you distribute the read load of your application across multiple databases or synchronize the database, either within Redis Enterprise or external to Redis Enterprise, to another database.

Cluster environment optimization

Redis Enterprise Software employs various algorithms to optimize performance. As part of this process, Redis Enterprise Software examines usage characteristics and load to adjust its run-time configuration accordingly. Depending on your specific usage characteristics and load, it might take Redis Enterprise Software some time to adjust itself to optimal performance. To ensure optimal performance, you must run your workload several times and for a long duration until performance stabilizes. In addition, you can optimize your cluster for two different environments:

Configuring Shard Placement

In Redis Enterprise Software, the location of master and slave shards on the cluster nodes can impact the database and node performance. Master shards and their corresponding slave shards are always placed on separate nodes for data resiliency. The shard placement policy helps to maintain optimal performance and resiliency. In addition to the shard placement policy, considerations that determine shard placement are: Separation of master and replica shards Available persistence and Redis on Flash (RoF) storage Rack awareness Memory available to host the database when fully populated The shard placement policies are:

Hardware requirements

The hardware requirements for Redis Enterprise Software are different for development and production environments. In a development environment, you can test your application with a live database. If you want to test your application under production conditions, use the production environment requirements. In a production environment you must have enough resources to handle the load on the database and recover from failures. Development environment You can build your development environment with non-production hardware, such as a laptop, desktop, or small VM or instance, and with these hardware requirements:

Networking
When architecting a Redis Enterprise Software solution, there are some specific networking features that are worth your time to understand and implement. Multi-IP and IPv6 Redis Enterprise Software (RS) supports server/instances/VMs with multiple IP addresses, as well as IPv6 addresses. RS related traffic can be logically and physically divided into internal traffic and external traffic: “Internal traffic” refers to internal cluster communications, such as communications between the nodes for cluster management purposes.

**Node persistent and ephemeral storage**

For each node in the cluster, you can configure both persistent storage and ephemeral storage paths. Persistent storage is mandatory. It is used by the cluster to store information that needs to persist even if a shard or a node fails, including server logs, configurations, files. For example, if you configure persistence for a database, then the persistence information is stored in this location. The persistent volume must be a SAN (Storage Area Network) using an EXT4 or XFS file system and be connected as an external storage volume.

**Supported connection clients**

You can connect to Redis Enterprise Software databases programmatically using client libraries. Redis client libraries To connect an application to a Redis database hosted by Redis Enterprise Software, use a client library appropriate for your programming language. You can also use the redis-cli utility to connect to a database from the command-line. For examples of each approach, see Get started using Redis Enterprise Software Note: You cannot use client libraries to configure Redis Enterprise Software.

**Synchronizing cluster node clocks**

To avoid problems with internal cluster communications that can impact your data integrity, make sure that the clocks on all of the cluster nodes are synchronized using Chrony and/or NTP. When you install Redis Enterprise Software, the install script checks if Chrony or NTP is running. If they are not, the installation process asks for permission to configure a scheduled Cron job. This should make sure that the node's clock is always synchronized.