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. “External traffic” refers to communications between the clients and the databases, as well as connections to the management UI in the browser. When only one IP address exists on a machine that serves as an RS node, it is used for both internal and external traffic.

Multiple Active Proxy Support

Redis Enterprise Software (RS) provides high-performance data access through a proxy process that manages and optimizes access to shards within the RS cluster. Each node contains a single proxy process. Each proxy can be active and take incoming traffic or it can be passive and wait for failovers. RS allows multiple databases to be created. Each database gets an endpoint (a unique URL and port on the FQDN). This endpoint receives all the traffic for all operations for that database.

Network Port Configurations

All Redis Enterprise Software deployments span multiple physical/virtual nodes. You’ll need to keep several ports open between these nodes. This document describes the various port ranges and their uses. Note: Whenever you create a new database, you must verify that the ports assigned to the new database’s endpoints are open. The cluster will not perform this verification for you. Ports and port ranges used by Redis Enterprise Software Redis Enterprise Software’s port usage falls into three general categories:

Enable private and public database endpoints

Describes how to enable public and private endpoints for databases on a cluster.

Use the OSS Cluster API

Redis OSS Cluster API reduces access times and latency with near-linear scalability. The Redis OSS Cluster API provides a simple mechanism for Redis clients to know the cluster topology. Clients must first connect to the master node to get the cluster topology, and then they connect directly to the Redis proxy on each node that hosts a master shard. Note: You must use a client that supports the OSS cluster API to connect to a database that has the OSS cluster API enabled.

Updated: June 4, 2020