Internode encryption

As of v6.2.4, Redis Enterprise Software supports internode encryption, which encrypts internal communication between nodes. This improves the security of data as it travels within a cluster.

Internode encryption is enabled for the control plane, which manages the cluster and its databases.

Internode encryption is supported for the data plane, which encrypts communication used to replicate shards between nodes and proxy communication with shards located on different nodes.

The following diagram shows how this works.

Data plane encryption is disabled by default for individual databases in order to optimize for performance. Encryption adds latency and overhead; the impact is measurable and varies according to the database, its field types, and the details of the underlying use case.

You can enable data plane encryption for a database by changing the database configuration settings. This lets you choose when to favor performance and when to encrypt data.

Prerequisites

Internode encryption requires certain prerequisites.

You need to:

- Upgrade all nodes in the cluster to v6.2.4 or later.
- Open port 3342 for the TLS channel used for encrypted communication.

Enable data plane encryption

To enable internode encryption for a database (also called data plane encryption), you need to enable the appropriate setting for each database you wish to encrypt. To do so, you can:
Use the admin console to enable the **Internode encryption** setting from the database configuration screen:

Use the `rladmin` command-line utility to set the `data_internode_encryption` setting for the database:

```bash
grep tune db <database_id> data_internode_encryption enable
```

Use the Redis Enterprise Software REST API to set the `data_internode_encryption` setting for the database:

```bash
put /v1/bdbs/${database_id}
{ "data_internode_encryption" : true }
```

When you change the data internode encryption setting for a database, all active remote client connections are disconnected. This restarts the internal (DMC) proxy and disconnects all client connections.

To enable data plane encryption by default for new databases, use `rladmin` to tune the cluster:

```bash
rladmin tune cluster data_internode_encryption enable
```

**Encryption ciphers and settings**

To encrypt internode communications, Redis Enterprise Software uses TLS 1.2 and the following Cipher suites:

- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-RSA-AES128-GCM-SHA256

No configurable settings are exposed; internode encryption is used internally within a cluster and not exposed to any outside service.

**Certificate authority and rotation**

Starting with v6.2.4, internode communication is managed, in part, by two certificates: one for the control plane and one for the data plane. These certificates are signed by a private certificate authority (CA). The CA is not exposed outside of the cluster, so it cannot be accessed by external processes or services. In addition, each cluster generates a unique CA that is not used anywhere else.

The private CA is generated when a cluster is created or upgraded to 6.2.4.

When nodes join the cluster, the cluster CA is used to generate certificates for the new node, one for each plane. Certificates signed by the private CA are not shared between clusters and they're not exposed outside the cluster.

All certificates signed by the internal CA expire after ninety (90) days and automatically rotate every thirty (30) days. Alerts also monitor certificate expiration and trigger when certificate expiration falls below 45 days. If you receive such an alert, contact support.
You can use the Redis Enterprise Software REST API to rotate certificates manually:

```
put /v1/cluster/certificates/rotate
```

**Updated:** April 26, 2022