FastX Advanced Cluster Managers

For FastX 3.2

This document provides information for advanced installations of the cluster manager. This includes High Availability as well as manual installations of the cluster manager

Software Requirements

NodeJS

Minimum Version 14+
Binary Distributions available at https://github.com/nodesource/distributions

Redis

Recommended Minimum Version 6.0
Note: Versions prior to 6.0 do not ship with TLS support making all communication in cleartext. It is strongly recommended that you do not use any version lower than 6.0

Debian/Ubuntu

https://redis.io/download

RPM Based Systems

The remi repository hosts version 6 of Redis. You can install remi by running the following where $OS_VERSION is the version of RedHat/CentOS you are using (ie 7 or 8)

yum install -y "https://rpms.remirepo.net/enterprise/remi-release-$OS_VERSION.rpm"

MongoDB

Recommended Minimum Version: 4.4+

MongoDB installation Repository
https://docs.mongodb.com/manual/administration/install-community/

FastX

Run the following command to install the repository

```
echo "deb [trusted=yes]
https://www.starnet.com/files/private/starnet-repo-debian ./stable" | sudo
tee /etc/apt/sources.list.d/starnet.list
```

Installing

```
sudo apt update
sudo apt install fastx-cluster fastx-microservices
```

Hardware Considerations

While the FastX Cluster Manager is lightweight, as the number of nodes scale up you may want to look into a more powerful system to handle the number of request across the cluster.

We currently recommend using the minimum system requirements of Redis as the requirements for the FastX cluster manager and scaling up from there depending on need.

Redis

https://docs.redis.com/latest/rs/administering/designing-production/hardware-requirements/

MongoDB

https://docs.mongodb.com/manual/administration/production-notes/#hardware-considerations

High Availability

When Provisioning a Cluster Manager for high availability, we recommend a minimum of 3 nodes for MongoDB and Redis. Ensure that all versions of NodeJS, MongoDB, and Redis are running the same version on all 3 systems.

NodeJS, MongoDB and Redis can be installed on the same system

Setting up MongoDB

MongoDB holds the configuration and the state of the cluster. Without this component, the cluster loses all information. In a High Availability environment, it is important to configure MongoDB to work as a Replica Set. Please refer to the MongoDB documentation for a full discussion. Creating a replica set is well documented by MongoDB and can be found using the following links.

Basic Tutorial on How to Deploy a Replica Set
https://docs.mongodb.com/manual/tutorial/deploy-replica-set/

Documentation index on MongoDB replication
https://docs.mongodb.com/manual/administration/replica-set-deployment/

Url String for a Replica Set
https://docs.mongodb.com/manual/reference/connection-string/

Set up MongoDB Security
https://docs.mongodb.com/manual/security/

Redis

Redis is a high throughput key value store used by the cluster manager to communicate with the FastX services. Redis is the transport mechanism of your FastX Cluster.

Configure Redis as a cluster

Follow the instructions on the redis tutorial to set up a cluster
https://redis.io/topics/cluster-tutorial

An example /etc/redis.conf file will look like this (including TLS enabled)

```
# fastx
bind 0.0.0.0
requirepass MY_SUPER_SECRET_PASSWORD
cluster-enabled yes
cluster-config-file nodes.conf
cluster-node-timeout 5000
appendonly yes


# FastX Cluster additions on Wed Oct 27 10:24:56 EDT 2021:
port 0
tls-port 6379
tls-cert-file /etc/pki/tls/certs/cluster2.dev.crt
tls-key-file /etc/pki/tls/private/cluster2.dev.key
```

```
tls-ca-cert-file /etc/pki/tls/certs/rootCA.crt
tls-auth-clients no
tls-cluster yes
tls-replication yes
```

Notes

The redis cluster will connect via TLS. The other redis nodes need to be able to verify the certificate against a root CA. Make sure to sign all the certificates using the same root CA and distribute the CA bundle accordingly (alternatively you can use tls-ca-cert-dir to make a CA directory). See the redis documentation for more details.

FastX Cluster Service

db.json

You need to modify some options in the */etc/fastx/db.json* to take advantage of the mongo replica set. Please refer to the [MongoDB URL Format Guide] for details. Query options can be set in the options section of the JSON object.

Replica Set configuration example

```
{
    "url":
"mongodb://manager1.example.com:27017,manager2.example.com:27017,manager3.e
xample.com:27017/fastx",
    "options": {
        "useUnifiedTopology": true,
        "replicaSet": "rs0",
        "readPreference": "primaryPreferred"
    }
}
```

broker.json

You need to modify some options in the */etc/fastx/broker.json* to set up the Redis cluster when connecting. Make sure to copy this file to ALL the nodes (cluster manager and compute nodes)

Example broker.json file

```
{
    "transporterType":"redis",
    "namespace": "fastx-cluster",
    "options": {
```

```
        "cluster": {
           "nodes": [
                { "host": "172.16.128.20", "port": 6739 },
                { "host": "172.16.128.21", "port": 6739 },
                { "host": "172.16.128.22", "port": 6379 }
          ],
          "clusterOptions": {
           "redisOptions": {
                  "password": "MY_SUPER_SECRET_PASSWORD",
                  "tls": {
                     "rejectUnauthorized": false
                  }
             }
          }
        }

    }
}
```

StarNet Communications Corp.
4677 Old Ironside Drive, Suite 210, Santa Clara, California 95054-1825 • USA
408.739.0881 • 408.739.0936 • sales@starnet.com • www.starnet.com