High Availability Implementation Guide for the Kinetic Edge & Fog Processing Module (EFM)

Last Updated: December 11, 2018

Table of Contents

Introduction

Overview

The objective of high availability (HA) architecture is to minimize Recovery Time Objective (RTO) and Recovery Point Objective (RPO) within the spending envelope of the customer. Recovery Time Objective is the maximum duration of time within which a business process must be restored after a failure or disaster. Recovery Point Objective is the amount of data at risk of being lost during recovery from a failure or disaster.

To implement HA for the Kinetic Edge Fog Module (EFM), you must consider the components of EFM. These three components are brokers, links, and the ParStream historian database.

Brokers and Links

When brokers and links are connected over a local loopback interface**,** a common HA strategy can be used for both. Two types of failures must be examined and managed by two different processes, as shown in the chart below:

Type of Failure	Recovery Process
Process failure of broker or link	Watcher process
Underlying hardware failure	Active-passive failover through VMware

The *watcher* process guarantees that if the broker or any local links fail, they are immediately restarted. This allows recovery from transient errors in a minimal amount of time for any individual component, which helps meet aims for RTO and RPO.

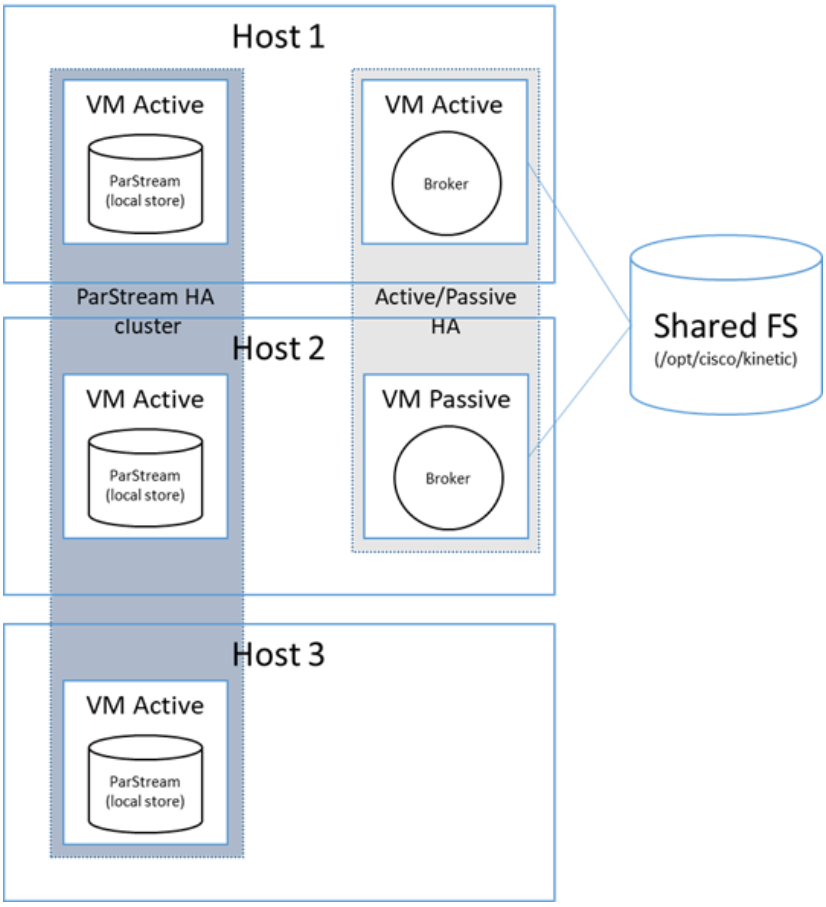
VMware, an implementation of virtualization technology, will be used to set up HA for brokers and links. VMware allows us to create two virtual machines (VMs) on separate hosts that share the same disk subsystem for the EFM installation. With the VMware HA feature, hardware failure on the primary VM is detected and the secondary VM is automatically activated in its place.

ParStream Historian Database

Unlike brokers and links, ParStream uses built-in clustering mechanisms, can support an active-active HA configuration and does not require external software. Its internal clustering capabilities must be activated for HA. Note that as ParStream is a distributed system, it has eventual consistency with updates.

Combined HA Configuration

The following diagram depicts the *minimum* number of hosts in a combined HA configuration of brokers, links, and the ParStream database as previously described.



A few comments on specific use cases with regards to HA in ParStream:

- For a use case that requires your data to have HA at all times, we need 3 nodes (redundancy of 2) for ParStream, as depicted.
- For a use case that combines smaller dimensional data with larger fact tables and data, making the smaller, dimensional data available on all nodes and thereby applying full redundancy helps to minimize cluster internal communication.
- For a use case that requires data availability in the case of failure of more than one node, a redundancy larger than 2 should be applied to larger fact tables. However, note that this comes at the cost of more storage allocation across the nodes.

How to Implement VMWare HA in EFM

High availability is implemented in EFM's brokers and links using VMware vSphere Availability 6.5. This external software allows us to create two virtual machines on separate hosts that share the same disk subsystem for EFM installation. With VMware HA, hardware failure on the primary VM is detected and the secondary VM is automatically activated in its place.

The separate hosts are VMware ESXi hosts, which are enterprise-class type-1 hypervisors used in conjunction with VMs.

Before creating VMware HA clusters, the following prerequisites must be met:

- All virtual machines and their configuration files must reside on shared storage.
- The ESXi hosts must be configured to have access to the same virtual machine network.
- Each host in the VMware HA cluster must have a host name assigned to it and a static IP address. The networking routing and reachability must be the same for each host.
- There must be CPU compatibility between the hosts. An ideal cluster is a cluster with exactly the same hardware and memory size.

It is recommended to use redundant Service Console and VMkernel networking configuration.

The following procedure works when (1) all of the links connecting to a broker are on the same VM and (2) the configurations of the passive broker and its links are identical to the active broker and its links. If (1) is not true, then the link that is not on the broker VM either has a separate HA configuration or none. If (2) is not true, then the operation of the system after failover will differ from that before the failover.

Once these prerequisites have been met, proceed to the implementation process on the VMware vSphere Web Client as detailed on the following page.

1. To turn on vSphere High Availability

- Click the cluster name under **Navigator** in the leftmost tab.
- Click **Configure**, then under Services, click **vSphere Availability**.
- Click **Edit** in the upper right corner.
- Check the top box next to **TURN ON vSphere HA** and click **OK**.

2. To select the necessary failure response

- Click **Edit** in the upper right corner as in step (1).
- Click **Failures and Responses**.
- Check the box next to **Enable Host Monitoring**.
- Select **Disabled** for **Response for Host Isolation**, **Datastore with PDL**, **Datastore with APD**, and **VM Monitoring**.
- Select **Restart VMs** for **Host Failure Response**.
- Confirm that these fields have been changed accordingly in **vSphere**

- **Availability**, under **Configure**.

3. To set the highest VM restart priority

- Click **Configure**.
- Click **VM Overrides** under **Configuration**.
- Click **Add** at the top left.
- Click the green plus sign (**+**).
- Select the VM that you want to have the highest priority.
- In the VM Restart Priority field, select **Highest**.
- Click **OK**.

4. To set the other VM restart priorities

- Click **Configure**.
- Click **VM Overrides**.
- Click **Add** at the top left.
- Click the green plus sign (**+**).
- Select all VMs other than the VM with the highest priority.
- In the VM Restart Priority field, select Disabled for each selected VM. An error will be given at the top next to a yellow warning sign.
- Click **OK**.

5. To Confirm

- Click **Configure**.
- Click **VM Overrides**.
- Confirm that the VM priority of the desired VM is set as **Highest** and the VM priority of the other VMs is set as **Disabled**.

How to Confirm that VMware HA is working properly

If in a testing or development environment, a good way to confirm that VMware HA is working properly is to simulate a failover. If in a production environment, checking some tabs in VI (Virtual Infrastructure) client can confirm that HA should be working properly.

To simulate a host or VM failure scenario, there are two options: simulation directly from the vSphere Web Client or simulation with the physical hardware in the lab, given access.

Simulation from the vSphere Web Client consists of rebooting a host with one or more powered-on VMs and checking to make sure that the VMs restart on another available host in the cluster. The steps are as follows:

1. Under the cluster in **Navigator**, identify the host to be rebooted and take note of its VMs, which will be restarted on another host within the cluster at the end of the process.
2. Right click the desired host, click **Power**, and the click **Reboot**.
3. When the Reboot Host frame appears, enter "HA Testing" as the reason and click **OK**.
4. Under **Navigator**, click on one of the previously noted VMs and check that its **Host** under **Summary** has been changed to a different one within the cluster. Repeat for each previously noted VM to confirm that they have been transferred to other host(s).
5. Confirm the **State** under **Summary** of the original, rebooted host returns to **Connected**.

Note that, for instance, in a testing environment with two ESXi hosts and one VM, the failover time in Step 5 was about 110 seconds. With EFM installed, the failover time for EFM (including time for EFM to reboot and dataflows to start running) was only about 5-15 more seconds. Depending on the complexity of the dataflows, the failover time will vary.

Simulation with the physical hardware in the lab relies on you to physically disconnect the server rather than rebooting from the vSphere Web Client. The steps are as follows:

1. Install the host operating system (Linux, Windows, etc.) on all VMs within the system, whether or not they are in the cluster. Confirm that the VMs have known static IP addresses.
2. On **vSphere Web Client**, select a VM on a host that is not in the cluster you are testing. In the middle panel, next to the VM's name on the top bar, click the icon of a computer with a green arrow to open the VM in a new console.
3. Enter the password you set for the VM in Step 1. When logged in, search and open **Terminal** on the VM.
4. Type the following, filling in the specific static IP address of one VM within the cluster you are testing:

   ```
   ping staticIP
   ```

   Confirm that messages[1] begin and continue to appear.
5. On vSphere Web Client, click on this VM. Next to **Host** under **Summary**, note which host the VM is currently running on. This is the host that you will physically disconnect to simulate a host failure.
6. Physically disconnect or unplug the server of the ESXi Host that the VM is running on.

[1] message example: 64 bytes from 10.16.192.63: icmp_seq=221 ttl=64 time=0.651 ms

7. On the **Terminal** of the VM, confirm that the messages stop appearing. Within a couple minutes, a new message[2] should appear, and eventually the same messages[3] should appear.

If you are working in a production environment and thus should not simulate a failure, checking a few tabs in VI client can help to confirm that HA is working properly with the following steps:

1. In **VI Client**, click on the first host name under the Cluster in **Navigator**.
2. Click **Summary** and then click **Configuration**.
3. Here, confirm that the vSphere HA State says **Connected** and next to it either **Slave** or **Master**.
4. Repeat the above for all hosts within the cluster.
5. Confirm that within the Cluster, there is exactly one Master and the rest are Slaves.

How to Configure HA in ParStream

High availability is implemented in ParStream using its internal clustering capabilities. Thus, no additional software is required here. To configure ParStream clusters, proceed to the following steps:

Configuration of ParStream clusters is executed in an INI file on a simple text editor, such as vi, Notepad or others. Any chapters, sections, and pages listed as additional information refer to the Cisco ParStream Manual.

Activate Clustering

A cluster is activated by setting the two mandatory global options in the INI file, `clusterId` and `registrationPort`.

The option clusterId is a string type that defines the unique ID that is used by all servers and importers to identify the cluster.

```
clusterId = MyTestCluster
```

The option `registrationPort` is an integer type that is the TCP port number used for the registration of cluster nodes and exchanging status information with the cluster leader; all servers and importers must use the same registration port.

```
registrationPort = ####
```

Update Redundancy

For HA, at least 3 nodes are necessary; however, the number of nodes required is specific to each use case of a customer. The main factors that must be considered when determining an appropriate number of nodes are incoming data, outgoing communication, high availability, and load distribution.

When the redundancy is greater than 1 (the default), you must set the global option `minUpdateRedundancy`.

```
minUpdateRedundancy = #
```

[2] message example: From 10.16.192.221 icmp_seq=124 Destination Host Unreachable
[3] message example: 64 bytes from 10.16.192.63: icmp_seq=221 ttl=64 time=0.651 ms

Specify Servers

To specify a server, you must name the server [server.*servername*] and set at least the node's hostname, port number, and rank.

After specifying multiple servers, confirm that each server has a unique rank so the leader election process works.

```
[server.servername]
host = servername
port = ####
rank = #
datadir = ./partitions
```

Specify Importers

To specify an importer, you must name the importer [import.*importername*] and set at least the node's hostname, source directory, target directory, and rank.

Note that importers too need a unique rank even though they cannot become leader.

```
[import.importer]
host = importername
sourcedir = ./import
targetdir = ./import-partitions
rank = #
leaderElectionPort = ####
```

Additional Options

Depending on specific use cases of a customer, you may want to customize clusters further. Additional options are available in the Cisco ParStream Manual Section 13.2 Global Options.

Example INI File

The following diagram depicts a very simple INI file that shows a cluster with 3 servers and 1 importer configured. These options use the same format as previously described.

A few notes about the above example INI file:

- All port numbers are greater than 1024, as lower numbers require administrative privileges.
- The port numbers of all three servers are the same simply for convenience; it is not necessary and these port numbers can be different across the nodes.
- The server names match their corresponding host names simply for convenience.
- The importer's host name matches the first server's host name because in this example, the importer and first server nodes are running on the same hardware.
- After the INI file is complete, it is placed in a directory "conf/", from where it is read by the starting server. Every server node has to have an identical copy of its file.

ExampleINI.ini

```
clusterId = MyExampleCluster
registrationPort = 1234

[server.server1]
host = server1
port = 1100
rank = 1
datadir = /data/partitions

[server.server2]
host = server2
port = 1100
rank = 2
datadir = /data/partitions

[server.server3]
host = server3
port = 1100
rank = 3
datadir = /data/partitions

[import.importer]
host = server1
sourcedir = /data/import
targetdir = /data/import-partitions
rank = 100
leaderElectionPort = 1111
```

How to Confirm that ParStream HA is working properly

To confirm that the clustering capabilities of ParStream are working properly after they have been configured, proceed with the following steps:

1. Open Command Prompt and type in

   ```
   $ pnc -U username -p portnumber
   ```

   where username and portnumber are replaced with the correct corresponding fields, and click **Enter**.

2. Confirm that “Connection Established” appears below the prompt.

3. Type in

   ```
   Cisco ParStream => select * from ps_info_cluster_node;
   ```

   and click **Enter**.

4. Identify the node information to check: in the lines that appear below the prompt entered, the information for leader, follower, active, and online nodes form a block of 1s and 0s.

5. Confirm that there is only one leader and one follower, and that all nodes are active and online by checking that in the block of 1s and 0s:

   a. the first column has exactly one 1 and the rest are 0s.

   b. the second column has exactly one 1 and the rest are 0s.

   c. the third column has all 1s.

   d. the fourth column has all 1s.

For an in-depth example of the procedure described above, refer to 3.6.3 Connect and View Cluster Information in the Cisco ParStream Manual and 6.2.4 State of a Cluster.

Obtaining documentation and submitting a service request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.
