EbManager Service

Contents

[1. Abstract 1](#_Toc424743413)

[2. Design 1](#_Toc424743414)

[3. Zookeeper Considerations 1](#_Toc424743415)

[4. Scenarios 2](#_Toc424743416)

[4.1. New EbServer Node Startup 2](#_Toc424743417)

[4.2. EbServer Node Restart 3](#_Toc424743418)

# Abstract

The EbManager Service runs on a small number of servers and is responsible for maintaining the global and complete state of the EbServer cluster. Upon startup, a participating node will notify its EbManager cluster of its readiness to participate in the cluster. Upon startup, a client node will query the appropriate EbManager cluster of its desire to read from and write to an EbServer cluster.

# Design

* The EbManager Service will run on a small number of EbManager nodes. Let the number of such EbManager nodes be *N*.
* Two SLB endpoints, *SLB(A)* and *SLB(B)*, will be configured to load balance amongst these *N* nodes. If one of these SLB endpoints were to go down or be unavailable for any reason, the other SLB endpoint could be used by EbServer and EbClient nodes.
* EbServer nodes, upon startup, will initialize themselves and notify the EbManager Cluster that they are ready to participate in the cluster.
* EbClient nodes, upon startup, will initialize themselves and request their particular slice of the partition map from the EbManager Cluster.
* EbClient nodes will then communicate directly with the appropriate primary EbServer nodes.



Figure 1 - Design

# Zookeeper Considerations

* The following hierarchical namespaces will be used in Zookeeper:
  + /EbServer
  + /EbServer/status
  + /EbServer/status/<hostname>
* All *zNodes* are persistent. Ephemeral zNodes are not used.

# Scenarios

## New EbServer Node Startup



Figure 2 - New EbServer Node Startup

* On startup, an EbServer Node will first read its local configuration file.
* This configuration file will specify the following:
  + *EbManager SLB(A)* (endpoint);
  + *EbManager SLB(B)* (endpoint);
  + EbManager port number.
* This node will determine that it is starting up for the very first time and send a *JoinRequest* to the EbManager cluster. It will randomly pick one of SLB(A) or SLB(B) to send this request to.
* This *JoinRequest* shall specify the following:
  + Verb (‘*Join*’);
  + Hostname (of request EbServer node);
  + HDD Capacity;
  + SSD Capacity;
  + Memory;
  + CPU Info.
* The SLB that receives this request, will forward it to one of the EbManager nodes[[1]](#footnote-2).
* The EbManager node that receives this request will validate the claimed properties of the requesting node against a configured set of minimum requirements.
* If the requesting node does not meet these criteria, it will be rejected with the following *JoinFailedResponse*:
  + Status (‘JoinFailed’);
  + Description (‘*Specification invalid*’).
* This EbManager node will determine the appropriate Zookeeper endpoint from its own configuration.
* This EbManager node will query Zookeeper for the latest state under the ‘*/EbServer/status/<hostname>*’ namespace.
* This query will return a *Null* response for the hostname specified.
* The EbManager node will determine that this node has never participated in the EbServer cluster.
* It will proceed to initialize the requesting node.
* The EbManager node will update the state for the requesting node in Zookeeper with the following:
  + Status (‘*Reserve’*);
  + HDD Quota;
  + SSD Quota;
  + Memory Quota;
  + CPU Quota;
  + Primary Partition Set;
  + Secondary Partition Set;
  + Current Timestamp.
* The EbManager node will return a *JoinResponse* back to the requesting EbServer node the following specified:
  + Status (‘*Success’*);
  + State (‘Reserve’)’
  + HDD Quota;
  + SSD Quota;
  + Memory Quota;
  + CPU Quota.

## EbServer Node Restart

1. *In the near future, a server-to-server authentication mechanism will be implemented to ensure that nodes participating in the EbServer and EbManager clusters authenticate themselves to each other.* [↑](#footnote-ref-2)