Oracle Financial Services Analytical Applications

Configuration for High Availability (HA)

Best Practices Guide

Version 8.0.1.1.0

DOCUMENT CONTROL

Version Number	Revision Date	Changes Done
1.0	Created: June 2016	Captured the Best Practices for OFSAA HA process.
2.0	Modified : Oct 2017	Added configurations for backend servers to enable distribution of tasks on multiple AM nodes.
3.0	Modified: Jan 2018	Added a note in Exclusions/Limitations section as per Bug 27322901.
4.0	Modified: July 2018	Rectified the boken links.
5.0	Modified: August 2021	Updated the Steps for HA Configuration section (Doc 31820287).

TABLE OF CONTENTS

Background

A High Availability architecture is one of the key requirements for any Enterprise Deployment. It refers to the ability of users to access a system without loss of service. Deploying a High Availability system minimizes the time when the system is down or unavailable, and maximizes the time when it is running or available. This section provides an overview of high availability from a problem-solution perspective.

Introduction

High Availability (HA) preparation is an integral part of the contingency planning. This document serves as a reference document for preparation of specific High Availability (HA) architecture. It explains how a standard OFSAA deployment should be architected so as to protect its applications from unplanned down time and minimize planned down time.

Objective

The objective of this document is to establish a process to configure OFSAA instance deployment for High Availability (HA).

NOTE: This document is not applicable for setting up a Disaster Recovery (DR) instance. It should be used to ensure service continuity through maintenance of an additional instance.

Assumptions

This document has been prepared after considering the below assumptions:

1. A Load Balancer (software/ hardware) is identified and installed.
2. An appropriate backup strategy for OFSAA File System (`$FIC_HOME` and `FTPSHARE`) and Oracle Database(s) is already in place.
3. Installation of the OFSAA platform and applications on the primary node is completed and setup is working.
4. A secondary instance (node) for OFSAA has been identified and is configured with appropriate prerequisite software required for OFSAA. No installation of OFSAA products is required at this stage.
5. Hardware configurations (in terms of RAM/ CPU/ CORE) do not vary between the OFSAA primary and secondary nodes.
6. It is also mandatory that the file system references such as the OS mount and folders, Web Application Server Profiles/ Domains/ Deployed Paths and so on are exactly the same between the primary and secondary nodes.

NOTE: The steps in this document consider the OFSAA version as 8.0.1.1.0 release and above. Check with Oracle Support Services if the same documented steps are applicable for any other specific OFSAA release.

Exclusions/ Limitations

1. The OFSAA instance(s) configuration is in ACTIVE-PASSIVE mode. Due to the architectural limitations of the OFSAA platform, the OFSAA components (processing layer) cannot be configured for ACTIVE-ACTIVE mode. However, the web and database tiers can be configured for ACTIVE-ACTIVE mode.

   NOTE: Though OFSAA instance(s) configuration is in ACTIVE-PASSIVE mode, OFSAA allows Distributed Activation Manager (AM) Based Batch Processing from 8.0.5.0.0 onwards, to configure AM engines to run on multiple OFSAA nodes. For more information, see *Distributed Activation Manager (AM) Based Processing* section in OFS Analytical Applications Infrastructure Administration Guide.

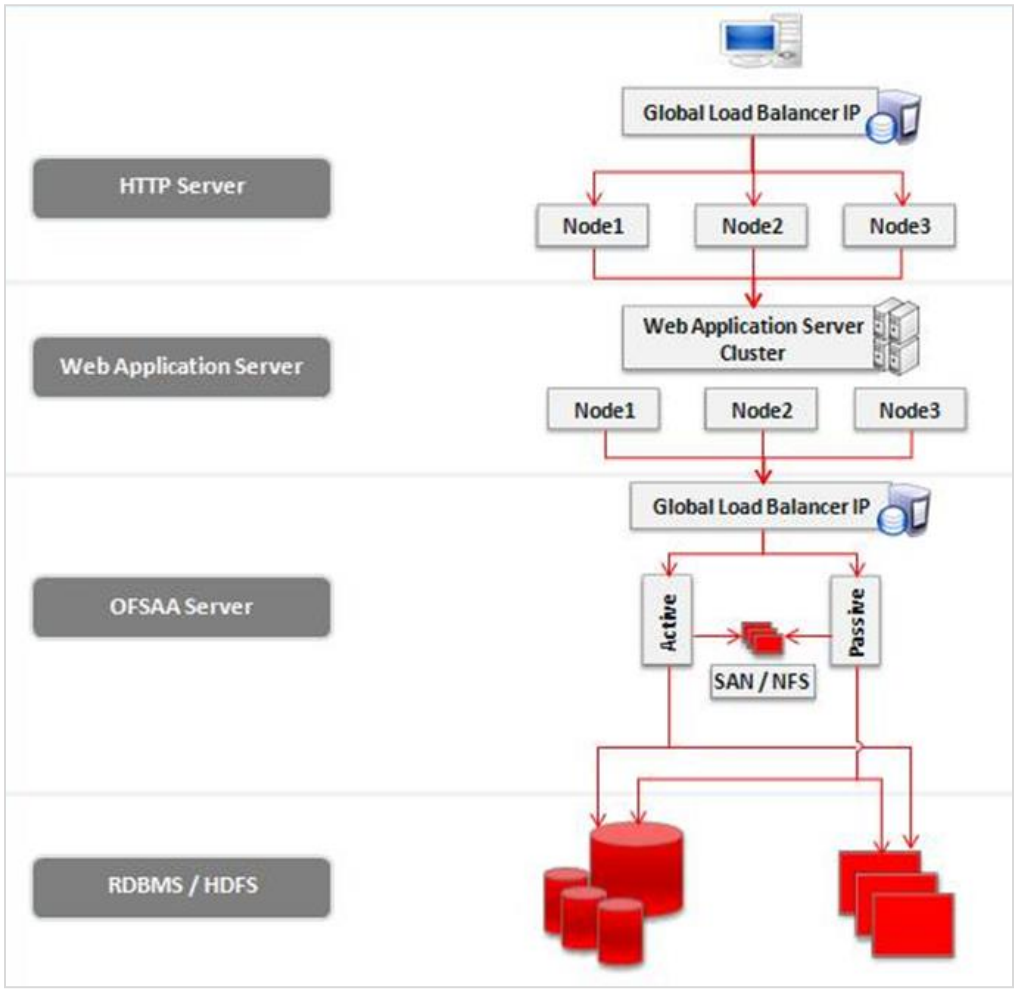
2. This document does not consider any particular OFSAA Application specific configuration. It documents the generic configuration across the platform that is generally applicable for the application stack deployed on top of it.
3. This document does not consider reporting layer HA configuration. For example, OBIEE server.

   This document considers HA configuration only against Oracle WebLogic Server and/ or IBM WebSphere Application Server.

Approach

Though there may be various ways in which the HA architecture can be devised after discussion on the requirements, following is the recommended approach (to be used as reference) to devise any further changes/ modifications to the architecture as per the use cases.

For the purpose of this topic, let us consider the following OFSAA deployment architecture for HA configuration as the end state:



In this diagram, the HA setup is proposed to be ACTIVE-ACTIVE configuration at the HTTP Server, Web Application Server, and Database/ HDFS layers. The OFSAA layer is configured for ACTIVE-PASSIVE configuration.

NOTE: Access to OFSAA applications will be by using the Global Load Balancer IP/ hostname (Virtual IP). In the event of primary node failure, the access to secondary node will be seamless, requiring no changes to the configuration information across all tiers.

NOTE: Session Affinity/ Sticky Session are configured at HTTP Server level.

NOTE: At any time, OFSAA patch installations should be performed only on active node. Promotions of patches to passive node are taken care as part of the sync-up process for File System components.

NOTE: HA configuration for HDFS should be put in place after referring to the HDFS vendor specific documentation. This document does not describe any details about HA configuration for HDFS.

Steps for HA Configuration

Assumptions:

- The Global Load Balancer(s) have been installed and any post installation configuration (hardening) has been completed before beginning with the following steps. If no load balancer has been installed, you may install and configure it on any host at this time.
- The OFSAA primary node installation was *not* performed keeping in mind the HA architecture. That is, multiple HTTP Servers, Web Application Server Cluster nodes, DB RAC cluster nodes, common file storage (FTPSHARE) and so on are not setup.
- The OFSAA primary node installation was performed using the local IP/ Hostname.

To configure the OFSAA instance(s) for HA configuration, perform the following steps:

Step 1 – Ensure the OFSAA primary node is up and running. You are able to access the OFSAA applications by entering the URL in the browser and login is successful.

Step 2 – Configure HA architecture.

Step 2.a – Install at minimum one additional HTTP Server, if only one HTTP Server is installed/ configured currently. If no HTTP Server is installed, you may install at minimum two HTTP Server(s). For information on HTTP Servers, see OFS AAAI Application Pack Installation and Configuration Guide.

Step 2.b – Configure the Global Load Balancer (at OFSAA Server level, that is, processing layer). See the Configuring OFSAA Load Balancer section as an example for configuration of software load balancer at OFSAA Server level.

Step 2.c – Configure the Global Load Balancer (at HTTP Server level). See Configuring HTTP Load Balancer section as an example for configuration of software load balancer at HTTP Server level. If this is already configured, skip and move to next step.

Step 2.d – If the web application server is already installed as a cluster, skip and proceed with the next steps.
Or
Install/ upgrade the Web Application Server as a cluster of nodes. Create the WebLogic Domain/ WebSphere Profile as appropriate. (Make a note of the paths). Update the HTTP Server configuration to use all web application server nodes. For more details, see Configuration for Apache HTTP Server.

Step 2.e – Archive and restore the existing DB schemas to a DB RAC installation. Ensure to retain the same schema names. (Make a note of the DB RAC URL). If the DB is not installed in RAC mode, you may do it now. Otherwise, skip and proceed with the next steps.

Step 2.f – Create a folder (`FTPSHARE`) on the common file storage (NAS/ NFS) and create a local mount point on the OFSAA server to access this folder.

Step 2.g – Copy the folder contents of the current `FTPSHARE` to the newly created folder as part of **Step 2.f**.

Step 2. h – Perform an FTP/ SFTP login on to the OFSAA server from command prompt and ensure you are able to access this folder contents.

Step 3 – Log in to the OFSAA primary node and stop the OFSAA services. For information on start/ stop of OFSAA services, see OFS AAAI Application Pack Installation and Configuration Guide.

Step 4 – Perform Hostname/ IP address change by following the steps documented in the section "Changing IP/ Hostname, Ports, Deployed paths of the OFSAA Instance" in OFSAAI Administration Guide.

At this time, provide Hostname/ IP address for OFSAA node as OFSAA GLIP in property `OFSAA_Server_IP_Address`. ***Do not*** change the ports. Retain the ports to same as setup during installation.

In the properties - Web Server IP/ Hostname and Port, enter the HTTP Layer GLIP and port configured (HTTP Server level).

Additionally, update the other parameters in the file to reflect change of parameter values for changes made (if any) as part of Steps **2.c, 2.d, 2.e** and **2.f.**

Step 4.a – Edit the `/etc/hosts` file (on the OFSAA primary node) and make an entry by adding the OFSAA GLIP alias as given:

```
192.0.2.1 ofss12345 glip1
```

Step 4.b – Edit the `web.xml` file in the `$FIC_HOME/ficweb/webroot/WEB-INF` directory (on the OFSAA node) and add an entry for **AllowHosts**.

For more information about AllowHosts, see the OFSAA Security Guide 8.0.x.

Step 5 – Navigate to `$FIC_WEB_HOME` and execute the command:

```
./ant.sh.
```

This generates the OFSAA web archive (.ear/ .war) file(s). For information on generating application archives, see OFS AAAI Application Pack Installation and Configuration Guide.

Step 6 – Navigate to `$FIC_HOME/ficapp/common/FICServer/bin/` and start the OFSAA services. For information on start/ stop of OFSAA services, see OFS AAAI Application Pack Installation and Configuration Guide.

Step 7 – Start the Web Server/ Web Application Server services. Access the Admin/ Deployment Console and deploy the archive(s) generated in **Step 5** above. For information on deploying application archives, see OFS AAAI Application Pack Installation and Configuration Guide.

Step 8 – Access the OFSAA application from browser by entering the new URL in the following format:

```
<scheme>://<host>:<port>/<ofsaa-context-name>/login.jsp
```

NOTE: The host and port entered in the URL would be of the Global Load Balancer (at HTTP Server level).

Step 9 – Enter the user name and password and ensure you are able to login and access the applications.

NOTE: At this point the OFSAA primary node is ACTIVE and the secondary node is PASSIVE.

Step 10 – Stop the OFSAA services on primary node. For information on start/ stop of OFSAA services, see OFS AAAI Application Pack Installation and Configuration Guide.

Step 11 - Perform the OFSAA instance cloning on secondary node. For more details, see Cloning OFSAA instance section.

Step 12 - Edit the `/etc/hosts` file (on the OFSAA secondary node) and make an entry by adding the OFSAA GLIP alias as given:

```
192.0.2.2 ofss54321 glip1
```

Step 13 - Start the OFSAA services on secondary node, Web Servers, and Web Application Server services. For information on start/ stop of OFSAA services, see OFS AAAI Application Pack Installation and Configuration Guide.

Step 14 - Configure the Global Load Balancer (at OFSAA Server level) to forward requests to OFSAA secondary node ***only*** if Load Balancer used does not do this automatically.

Step 15 – Access the OFSAA application from browser by entering the URL in the following format:

```
<scheme>://<host>:<port>/<ofsaa-context-name>/login.jsp
```

NOTE: The host and port entered in the URL should be of the Global Load Balancer (at HTTP Server level).

Step 16 – Enter the user name and password and ensure you are able to login and access the applications.

NOTE: At this point the OFSAA primary node is PASSIVE and the secondary node is ACTIVE.

At any point in time, ***only one*** OFSAA node services should be running. If both the node services are running at the same time, routing OFSAA requests will result in incorrect results.

NOTE: If either of the OFSAA instance (primary/ secondary) goes down, ensure the folder contents for `$FIC_HOME` are synced up on the other node using a utility such as Remote Sync (rsync) prior to bringing up the OFSAA services.

NOTE: The sync-up should be a scheduled activity at regular intervals. If you wait for the sync-up until the primary node goes down, you may not be able to sync-up at a later stage.

See the OS specific documentation on configuring Rsync.

Example of rsync command:

```
rsync –uavzP /scratch/ofsaaapp/ofsaa
ofsaauser@drsecondaryserver:/scratch/ofsaaapp/ofsaa
```

-u skip files that are newer on the receiver

-a archive mode; equals -rlptgoD (no -H,-A,-X)

included with "-a"

-r recurse into directories

-l copy symlinks as symlinks

-p preserve permissions

-t preserve modification times

-g preserve group

-o preserve owner (super-user only)

-D same as --devices --specials

-v increase verbosity

-z compress file data during the transfer

-P show progress during transfer

Configuring HTTP Load Balancer

Configure the Global Load Balancer to forward requests to the HTTP Servers using any preferred routing algorithm such as round robin. Refer the following configuration done using HAProxy tool.

NOTE: The following configuration was performed on HAProxy version 1.6.4.

Configure the following setting in `haproxy.cfg` file:

```
frontend ft_web
  bind <hostname>:80
  default_backend bk_web

backend bk_web
  balance roundrobin
  cookie JSESSIONID prefix nocache
  server s1 <server1>:80 check cookie s1
  server s2 <server2>:80 check cookie s2
```

Configuration for Oracle HTTP Server

No need to enable sticky sessions (insert cookie) on the load balancer when Oracle HTTP Server is front-ending Oracle WebLogic Server. You need sticky sessions if you are going directly from the load balancer to Oracle WebLogic Server, which is not the case in the topology described in this document.

For details, see documentation available at http://docs.oracle.com/cd/E23943_01/core.1111/e12037/web_tier_config.htm#WCEDG577.

Configuration for Apache HTTP Server

Configure the following setting in `httpd.conf` file:

```
ProxyPass /test balancer://mycluster stickysession=JSESSIONID
<Proxy balancer://mycluster>
BalancerMember http://<server1>:80 route=1
```

```
BalancerMember http://<server2>:80 route=2
</Proxy>
```

Alternatively, it can be set within balancer configuration using `ProxySet stickysession=JSESSIONID`:

```
<Proxy balancer://mycluster>
BalancerMember http://<server1>:80 route=1
BalancerMember http://<server2>:80 route=2
ProxySet stickysession=JSESSIONID
</Proxy>
```

For details, see documentation available at http://httpd.apache.org/docs/2.2/mod/mod_proxy.html.

Configuration for IBM HTTP Server

Configure the following setting in `plugin-cfg.xml` file:

Set `IgnoreAffinityRequests="false"`

For details, see documentation available at http://www-01.ibm.com/support/knowledgecenter/SSAW57_6.1.0/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/rwsv_plugincfg.html.

Configuring OFSAA Load Balancer

Configure the Global Load Balancer (for OFSAA server) to forward requests to the OFSAA nodes. Modify the `haproxy.cfg` file using HAProxy tool as shown below:

```
#---------------------------------------------------------------------
# Example configuration for a possible web application.  See the
# full configuration options online.
#
#   http://haproxy.1wt.eu/download/1.4/doc/configuration.txt
#
#---------------------------------------------------------------------


#---------------------------------------------------------------------
# Global settings
#---------------------------------------------------------------------
global
    # to have these messages end up in /var/log/haproxy.log you will
```

```
    # need to:
    #
    # 1) configure syslog to accept network log events.  This is done
    #    by adding the '-r' option to the SYSLOGD_OPTIONS in
    #    /etc/sysconfig/syslog
    #
    # 2) configure local2 events to go to the /var/log/haproxy.log
    #   file. A line like the following can be added to
    #   /etc/sysconfig/syslog
    #
    #    local2.*                       /var/log/haproxy.log
    #
    log         192.0.2.1 local2

    chroot      /var/lib/haproxy
    pidfile     /var/run/haproxy.pid
    maxconn     4000
    user        haproxy
    group       haproxy
    daemon

    # turn on stats unix socket
    stats socket /var/lib/haproxy/stats

#---------------------------------------------------------------------
# common defaults that all the 'listen' and 'backend' sections will
# use if not designated in their block
#---------------------------------------------------------------------
defaults
    mode                    http
    log                     global
    option                  httplog
    option                  dontlognull
    option http-server-close
#    option forwardfor       except 192.0.2.1/8
    option                  redispatch
    retries                 3
```

```
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                 3000

## Start Entries for OFSAA JAVA port and native port. ##

frontend haproxy_in
                                mode tcp
                                option tcplog
                                bind *:9999
                                default_backend haproxy_backend1

backend haproxy_backend1
                                balance roundrobin
                                mode tcp
                                option tcplog
                                server web1 ofsaaserver1:9999 check
                                server web2 ofsaaserver2:9999 check

frontend haproxy_in1
                                mode tcp
                                option tcplog
                                bind *:6666
                                default_backend haproxy_backend2

backend haproxy_backend2
                                balance roundrobin
                                mode tcp
                                option tcplog
                                server web3 ofsaaserver1:6666 check
                                server web4 ofsaaserver2:6666 check
```

```
## End Entries for OFSAA JAVA port and native port. ##

## Start Entries for OFSAA ICC port. ##

frontend haproxy_in2
                                mode tcp
                                option tcplog
                                bind *:6507
                                default_backend haproxy_backend3

backend haproxy_backend3
                                balance roundrobin
                                mode tcp
                                option tcplog
                                server web5 ofsaaserver1:6507 check
                                server web6 ofsaaserver2:6507 check

## End Entries for OFSAA ICC port. ##
```

Configuring Backend Servers to Enable Distribution of Batch Tasks on Multiple AM Nodes

Append the `haproxy.cfg` file using HAProxy tool with the following configuration:

```
## Start Entries for OFSAA Router port. ##
      frontend haproxy_in3
          mode tcp
          option tcplog
          bind *:6500
          default_backend haproxy_backend4


      backend haproxy_backend4
          balance roundrobin
          mode tcp
          option tcplog
```

```
            server web7 <<routerhostname:port>> check
            #server web8 <<routerhostname:port>>  check
## End Entries for OFSAA Router port. ##

## Start Entries for OFSAA AM port. ##
      frontend haproxy_in4
            mode tcp
            option tcplog
            bind *:6505
            default_backend haproxy_backend5

      backend haproxy_backend5
            balance roundrobin
            mode tcp
            option tcplog
            server web9 <<AMhostname:port>> check
            server web10 <<AMhostname:port>> check
## End Entries for OFSAA AM port. ##


## Start Entries for OFSAA MessageServer port. ##
      frontend haproxy_in5
            mode tcp
            option tcplog
            bind *:6507
            default_backend haproxy_backend6

      backend haproxy_backend6
            balance roundrobin
            mode tcp
```

```
        option tcplog
        server web11 <<Messageserverhostname:port>> check
        #server web12 <<Messageserverhostname:port>> check
## End Entries for OFSAA MessageServer port. ##
```

NOTE: Message Server should be running in all the nodes where AM servers are configured.

Cloning the OFSAA instance

Following are the steps to perform a short clone of the OFSAA instance:

1. Log in to the OFSAA primary node as a non-root user.
2. Archive the `$FIC_HOME` folder along with its sub-folders/ files using the following command:

   ```
   tar –zcvf FIC_HOME.tar.gz ./FIC_HOME
   ```

3. Copy the archive in binary mode on to the OFSAA secondary node.
4. Log in to the OFSAA secondary node as a non-root user.
5. Extract the archive at appropriate locations on the OFSAA secondary node using the following command:

   ```
   tar –zxvf FIC_HOME.tar.gz
   ```

6. Grant permission 750 recursively on these folders and their contents.

   ```
   chmod -R 750 <folder name>
   ```

7. Copy the user `.profile` contents (section added by OFSAA installation only) to the user `.profile` on secondary OFSAA instance.
8. Modify the `FIC_HOME`, `PATH`, `LIBPATH` and any other environment variable values as appropriate.
9. Create the `FTPSHARE` directory (using the same path as in primary node).
10. Save and execute the `.profile`.

OFSAA

Configuration for High Availability (HA) Best Practices Guide

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:

Phone: +1.650.506.7000

Fax: +1.650.506.7200

www.oracle.com/us/industries/financial-services/