Cloud Service Automation

Software Version: 4.80

For Microsoft Windows and Linux operating systems

Cluster Configuration using a Load Balancer

Document Release Date: January 2017

Software Release Date: January 2017

Hewlett Packard Enterprise

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

The OpenStack® Word Mark and the Square O Design, together or apart, are trademarks or registered trademarks marks of OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

RED HAT READY™ Logo and RED HAT CERTIFIED PARTNER™ Logo are trademarks of Red Hat, Inc.

This product includes an interface of the 'zlib' general purpose compression library, which is Copyright © 1995-2002 Jean-loup Gailly and Mark Adler.

Documentation Updates

To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hpe.com/.

This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

Support

Visit the HPE Software Support site at: https://softwaresupport.hpe.com/.

Most of the support areas require that you register as an HP Passport user and to sign in. Many also require a support contract. To register for an HP Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HP Passport login page.

To find more information about access levels, go to: https://softwaresupport.hpe.com/web/softwaresupport/access-levels.

HPE Software Solutions Now accesses the HPSW Solution and Integration Portal website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is https://softwaresupport.hpe.com/km/KM01702731.

Contents

Cluster configuration using a load balancer overview

HPE Cloud Service Automation (CSA) uses JBoss clustering technology to enable you to configure an active/active (high-availability) cluster. Clustering enables you to run CSA on several parallel servers called *nodes*. Cluster configuration improves performance on systems that handle high transaction volumes or large numbers of concurrent users. In addition to handling higher user loads and providing greater scalability, the cluster configuration supports server failover features.

Web requests to the CSA Controller or Marketplace Portal are load balanced among the nodes in the cluster. Increasing the number of nodes in the cluster will improve web request transaction throughput. Increasing the number of nodes in the cluster will also improve the response time by CSA fulfillment services to a high volume of concurrent deployment requests.

Because clustering distributes the workload across different nodes, if any node fails, CSA remains accessible through other nodes in the cluster. You can continue to improve CSA throughput by simply adding nodes to the cluster. If a node shuts down, activities such as email notifications that are scheduled to run on that node are automatically transferred to another available node. This server failover feature helps ensure that CSA remains operational.

Unsaved changes on a node that shuts down are lost and are not transferred to an available node. Users who log on to CSA after a node shuts down see only changes that were saved on that node.

> **Note:** In this document, path names beginning with the home directory such as `CSA_HOME`, apply to both Windows and Linux path names, even though they appear in Linux format.

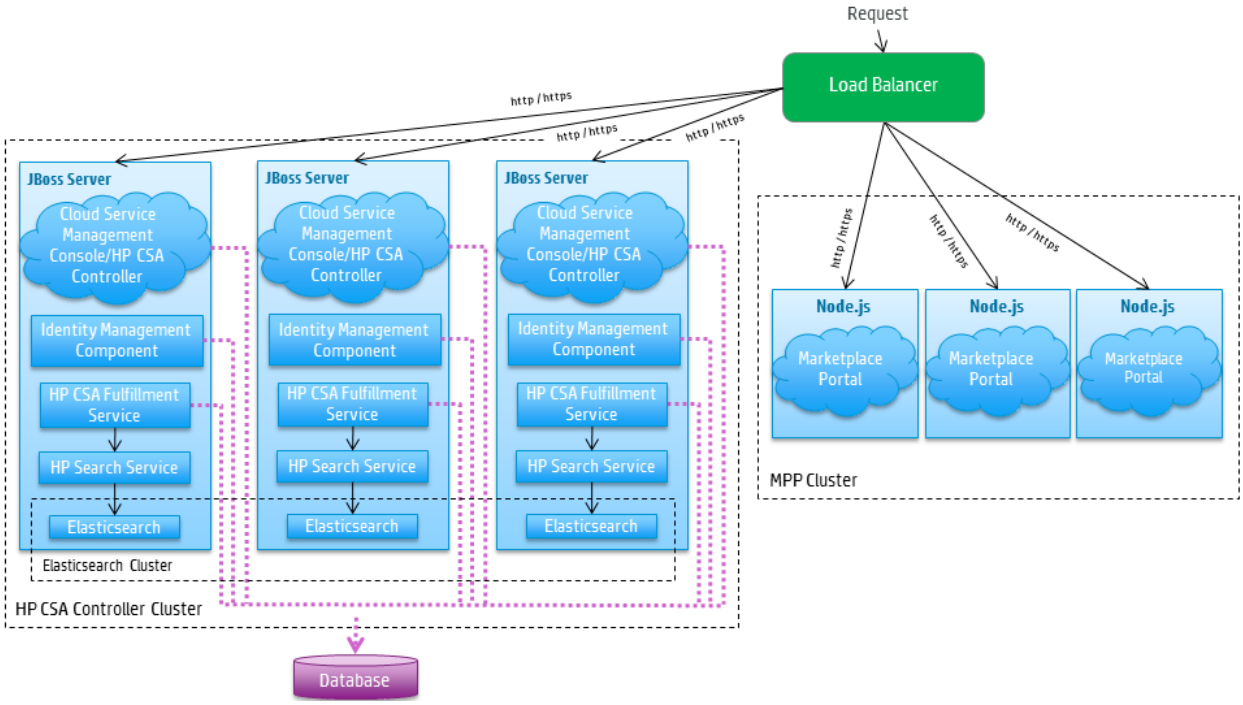
Example cluster configuration

In the following diagram, the example cluster configuration consists of: seven different physical (or virtual) hosts:

- One host is running a load balancer that proxies web requests into the CSA/JBoss cluster or Node.js cluster (for the Marketplace Portal)
- Three hosts are running CSA in standalone mode
- Three hosts are running the Marketplace Portal.

> **Note:** Content on how to use a database cluster or Oracle RAC is beyond the scope of this document. However, configuring CSA to use a Microsoft SQL Server cluster is no different from configuring CSA to use a standalone Microsoft SQL Server. Install and configure the Microsoft SQL Server cluster according to the manufacturer's documentation and follow the instructions to install CSA using a Microsoft SQL Server in the *Cloud Service Automation Installation Guide*.
>
> For information about configuring CSA with Oracle RAC, see the *Configuring CSA to Work with Oracle RAC* whitepaper.

Example cluster configuration diagram

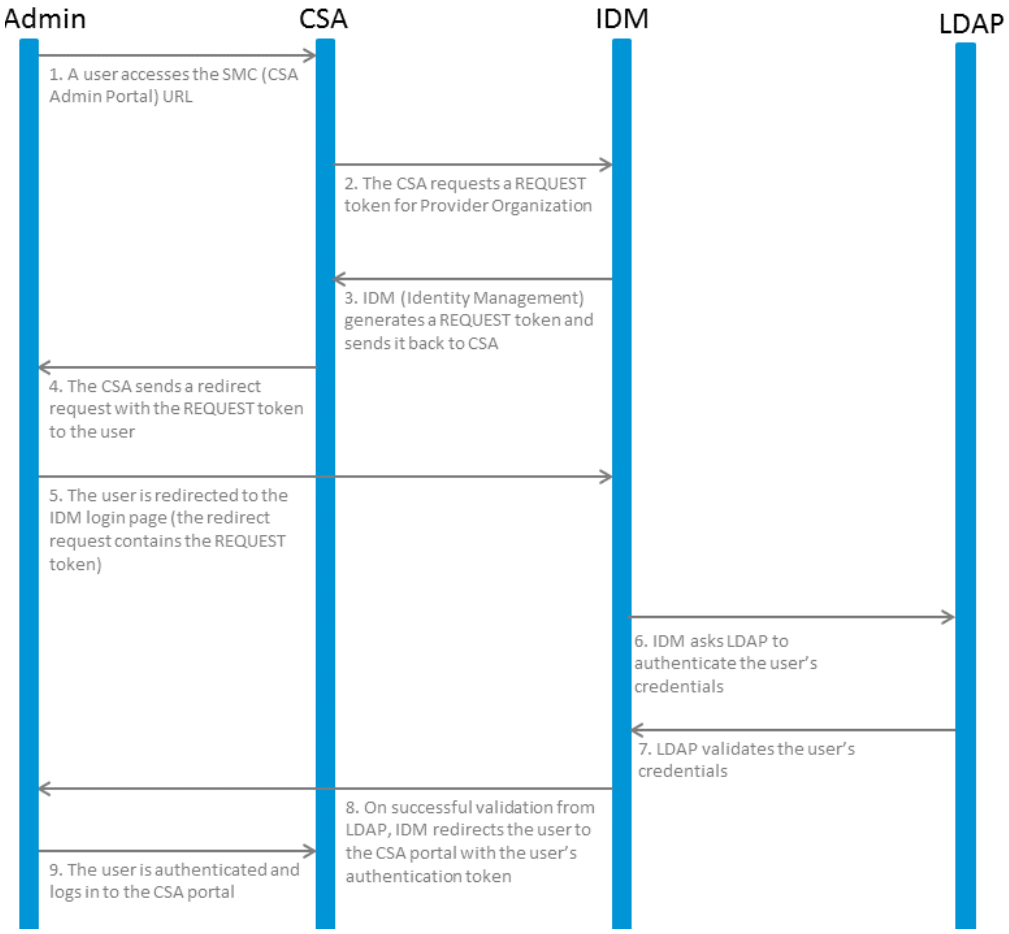


The cluster uses a load balancer to distribute requests among any number of nodes. The load balancer (internal or external) listens for HTTP/S requests from standard interface clients and forwards them to one of the nodes. These nodes are transparent to users and users access only the URL to the load balancer.

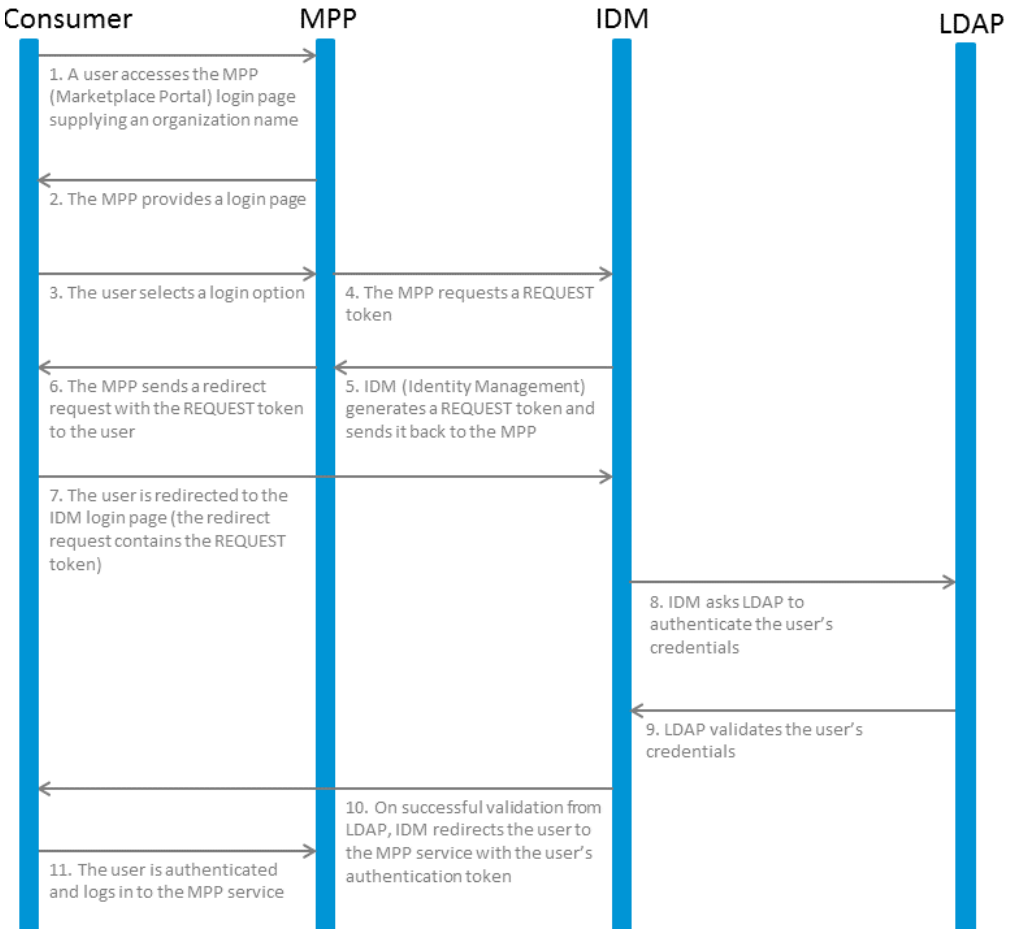
Request flow

The following diagrams show how a request (distributed from the load balancer) is processed in the clustered environment for the Cloud Service Management Console and Marketplace Portal.

Cloud Service Management Console request flow



Marketplace Portal request flow



About the examples

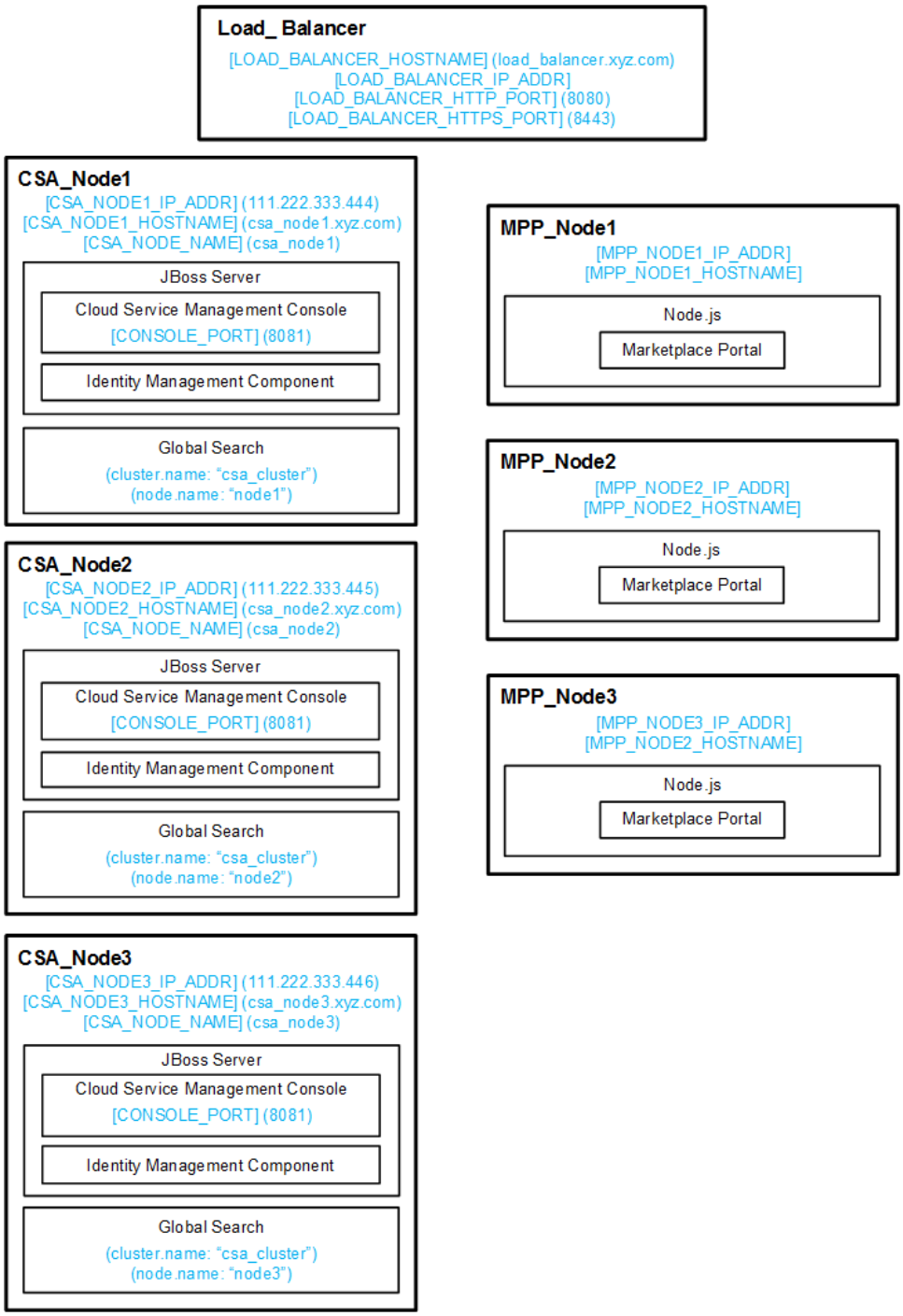
In this document, the following names are used to identify the hosts or nodes in the clustered environment:

- **Load_Balancer node**: the load balancer that distributes requests among the nodes in the clustered environment
- **CSA_Node nodes**: hosts CSA
- **MPP_Node nodes**: hosts the Marketplace Portal

In this document, an item denoted in square brackets is a placeholder for the actual value that has been configured (for example, the hostname of the `CSA_Node1` node is denoted as `[CSA_NODE1_HOSTNAME]`).

In the following diagram, items in parentheses are default or example values used in this document (for example, the default HTTP port used by the Cloud Service Management Console is 8081).

Example values for Example cluster configuration using a load balancer



To set up the nodes, you must have knowledge of or work with someone who has knowledge of CSA, Operations Orchestration, load balancers, JBoss, and resource providers that will be integrated with CSA.

General notes about configuring a clustered environment

The following information should be considered when configuring a clustered environment:

- It is recommended that you install and configure the nodes in the order presented in this guide. There are some tasks that are dependent on this order (such as generating certificates and importing them).

  Install and configure the load balancer node first. Follow the manufacturer's recommendations to install and configure the load balancer.

- The system time among all nodes in the cluster must be synchronized. If the time is not synchronized, users may experience problems such as not being able to log in to the Marketplace Portal.
- CSA must be installed in the same directory on all nodes. Some file locations are hardcoded in configuration files and, if these file locations do not match among nodes, CSA fails to start.
- JavaScript files are used to implement dynamic properties in CSA. All JavaScript files are stored in the CSA database. The JavaScript files are managed using the application UI and the changes apply to all nodes in a cluster. Managing the JavaScript directly on the server filesystem is no longer possible. For details about developing and using JavaScripts in CSA, see the *Dynamic Options and Cascading Lists* white paper.

  > **Caution:** Use caution when deleting or modifying a script. In a clustered environment, the script will be removed or modified on all nodes.

- JSP files can also be used to implement dynamic properties in CSA. However, the JSP files have been deprecated due to security and technical issues, so the way these files are used in a clustered environment is different from the way JavaScript files are used. The JSP files are also stored in the CSA database, but they cannot be run from the database. Because of that, CSA copies the JSP files from the database to the particular node filesystem from which they are being executed. It is possible to override a JSP in the database by placing a custom version of the same JSP on the node filesystem. Such a custom JSP is not propagated to the other nodes in a cluster however. For details about developing and using JSP scripts in CSA, see the *Dynamic Options and Cascading Lists* white paper.

Check for updates

Documentation may be updated periodically. To check for recent updates or to verify you are using the most recent edition of a document, click the "go" link to download the guide from the HPE Software Support portal. HPE Passport login is required.

Document	Link
Configuration Guide	go
Cluster Configuration Guide Using a Load Balancer	go
Cluster Configuration Guide Using an Apache Web Server	go
FIPS 140-2 Compliance Configuration Guide	go
FIPS 140-2 Compliance Statement	go

Configure the load balancer node

Install and configure the load balancer on the load balancer node before setting up the CSA cluster configured for high availability.

Install the load balancer

Install the load balancer following the manufacturer's recommendations. Refer to the manufacturer's documentation for more information.

Configure the load balancer

The load balancer should be configured to balance the workload among the nodes in the CSA/JBoss cluster and the nodes in the Marketplace Portal/Node.js cluster.

Configure the load balancer following the manufacturer's recommendations (refer to the manufacturer's documentation for more information) with the following exceptions:

- When configuring the load balancer for the Marketplace Portal`/Node.js` cluster, sticky sessions must be configured. Sticky sessions are required for session persistence between the load balancer and the CSA andMarketplace Portal nodes in a clustered environment. Sessions are not shared across the CSA or Marketplace Portal nodes, thus requiring sticky sessions.
- By default, CSA supports secure connections using TLSv1.2 and to enable support for TLSv1.2, you must configure the load balancer. CSA configuration can be manually changed to support TLSv1.1 or TLSv1.0, to work with older load balancers or other HTTPS clients that do not support TLSv1.2. However it is not recommended to enable TLSv1.1 or TLSv1.0 for security reasons.
- The load balancer `ProxyTimeOut` value should be set to a higher value than the default value. For more details see the *Cloud Service Automation Troubleshooting Guide*.

Generate the certificate

If you will be configuring a secure connection (using a protocol such as TLS) to communicate from the load balancer to the CSA and Marketplace Portal nodes, you will need to generate the load balancer's

certificate (in this document, it will be referred to as `load_balancer.crt`). Copy this certificate to the `CSA_HOME/jboss-as/standalone/configuration` directory on the CSA nodes and to the `CSA_HOME/portal/conf/` directory on the Marketplace Portal nodes.

> **Note:** When configuring CSA, if you intend to refer to the load balancer system by its IP address rather than its fully-qualified domain name, you must generate the certificate with the `Subject Alt Name` attribute set to the IP address of the load balancer system.

Start the load balancer

You can start the load balancer now (following the manufacturer's recommendations) or after configuring the CSA cluster.

Configure the CSA node

This chapter describes how to install, upgrade, and configure a CSA node (for example, csa_node1, csa_node2, or csa_node3) in a CSA cluster configured for high availability.

The CSA node consists of:

- CSA
- Global search
- Identity Management component

To configure the CSA node, do the following:

- Install or Upgrade CSA on each CSA node
- "Configure CSA on each CSA node" on page 17

Install CSA on each CSA node

Install CSA on each CSA node as described in the *Cloud Service Automation Installation Guide* with the following limitations:

- You must install the same version of CSA on each node (including the Marketplace Portal nodes).
- Install CSA in the same location in which you installed or will be installing CSA on all CSA nodes.
- Install CSA database components and create the database schema for one and only one of the CSA nodes. It is recommended that you create the schema when you install CSA on the first CSA node. Then, you do not need to create the schema when you install CSA on the other nodes.

  > **Note:** All CSA nodes must connect to the same database schema. However, you only need to create the database schema once.

- You can only use the installer to install sample content on the node on which database components have been installed and the database schema has been created. On the other nodes in the cluster, use the Cloud Content Capsule Installer to install the sample content after the database schema has been created. See the *Cloud Service Automation Capsule Installer Guide* for more information.
- Use an external (existing) instance of Operations Orchestration.

> **Note:** You cannot configure CSA in a clustered environment using the embedded Operations Orchestration instance.

> **Note:** It is recommended that you install Operations Orchestration in its own cluster configured for high availability. Refer to the Operations Orchestration documentation for more information.

- You must configure a secure protocol connection (such as TLS) between Operations Orchestration and all CSA nodes.
- If you are configuring Operations Orchestration for sequenced designs, set the **Property Value** of the **CSA_REST_URI** System Property to:

  `https://[LOAD_BALANCER_HOSTNAME]:[LOAD_BALANCER_CSA_HTTPS_PORT]/csa/rest`.

  For example:

  `https://load_balancer.xyz.com:8443/csa/rest`

Guides are available on the HPE Software Support web site at: https://softwaresupport.hpe.com (this site requires a Passport ID). Select **Dashboards > Manuals**.

Upgrade CSA on each CSA node

To upgrade CSA, on each CSA node, do the following:

1. If you are using the JRE that is installed with CSA (OpenJRE), back up the truststore (`CSA_JRE_HOME/lib/security/cacerts`) and/or the load balancer certificate outside of `CSA_HOME`.

   Because the JRE will be upgraded, the truststore is also upgraded. Any certificates you manually imported into the truststore will be lost unless you back up the truststore or the certificates. Do not re-use the truststore from the old version of the JRE (in case it contains public Certificate Authority certificates that are no longer trusted). Instead, you must export any root and/or self-signed certificates from the old truststore (certificates that you had manually imported into the old truststore) and import them into the new JRE truststore after running the upgrade installer.

2. Upgrade CSA as described in the *Cloud Service Automation Upgrade Guide* with the following exceptions:

   - When running the upgrade installer, install CSA database components and upgrade the database schema on one and only one of the CSA nodes. It is recommended that you upgrade the schema on the first CSA node that you upgrade. Then, you do not need to install CSA

database components and upgrade the database schema when you upgrade CSA on the other CSA nodes.

> **Note:** All CSA nodes must connect to the same database schema. However, you only need to install CSA database components and upgrade the database schema once.

- When running the upgrade installer, you must continue to use an external (existing) instance of Operations Orchestration. You cannot install or upgrade to use an embedded Operations Orchestration instance.
- When running the upgrade installer, you cannot install the additional sample content. You can deploy the content after the upgrade has completed. See the *Cloud Service Automation Content At a Glance Guide* for information on how to manually deploy this content.
- After running the upgrade installer, any references to the `CSA_HOME/jboss-as/standalone/configuration/standalone.xml` file in the *Cloud Service Automation Upgrade Guide* should be applied to the `CSA_HOME/jboss-as/standalone/configuration/standalone-full-ha.xml` file instead.

  For example, in CSA version 4.50, if you customized the host or ports in the `standalone-full-ha.xml` file, when you follow the instructions in the *Recustomize Host and Ports* section of the *Cloud Service Automation Upgrade Guide*, instead of updating the `standalone.xml` file, you should update the `standalone-full-ha.xml` file. If you do not remember the customizations you made to the file, see the backed up copy, `CSA_HOME/_CSA_4_70_0_installation/Backup/standalone/standalone-full-ha.xml`.
- After upgrade to CSA 4.80, if the `initString` value in the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/hpssoConfig.xml` file is detected to be the default value created during a previous installation of CSA, the `initString` value is regenerated as a security measure.

  If the external Operations Orchestration had not already been configured for SSO, the upgrade process will attempt to update the external Operations Orchestration's SSO configuration with the new `initString` value generated during installation.

  Any other products that you had configured for SSO with CSA will need to be updated to share a common `initString` with CSA. After upgrade to CSA 4.80, you should review and update, as needed, the SSO configuration in Operations Orchestration and other integrated products. For more information on configuring SSO between CSA and other products, see the *Cloud Service Automation Configuration Guide*.

3. Follow the instructions in "Configure CSA on each CSA node" on the next page to configure after running the upgrade installer. Do not copy back files from an earlier version of CSA unless you are instructed to do so. Many components of CSA, such as the JRE, JBoss, and Identity

Management component, have been updated and therefore, the configuration files have also been updated. Some files may have retained the information you configured in the previous version. However, you should verify all information in the upgraded files.

Configure CSA on each CSA node

Complete the following tasks to configure CSA on each CSA node:

- "Edit properties" below (required)
- "Enable JNDI" on the next page (required)
- "Request a software license" on page 19 (required)
- "Configure Marketplace Portal redirection" on page 19 (required)
- "Configure JBoss" on page 20 (required)
- "Configure a secure connection" on page 21 (required)
- "Configure the Identity Management component on the CSA node" on page 22 (required)
- "Reconfigure the CSA service" on page 25 (required)
- "Configure global search" on page 27 (required if using global search)
- "Configure Single Sign-On" on page 34 (required if using SSO)

Edit properties

Update property values to route requests to the Cloud Service Management Console through the load balancer node and set the mode in which CSA is running.

Edit properties to route requests to the Cloud Service Management Console through the load balancer node

To edit properties to the Cloud Service Management Console through the load balancer node, complete the following steps:

1. Edit the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/csa.properties` file as follows:

   a. Set the following properties:

   ```
   csa.provider.hostname=[LOAD_BALANCER_HOSTNAME]
   csa.provider.port=[LOAD_BALANCER_CSA_HTTPS_PORT]
   csa.provider.rest.protocol=https
   deploymentMode=clustered
   ```

   For example:

   ```
   csa.provider.hostname=load_balancer.xyz.com
   csa.provider.port=8443
   csa.provider.rest.protocol=https
   deploymentMode=clustered
   ```

   > **Note:** If you set the `csa.provider.hostname` attribute to the IP address of the system on which the load balancer is installed, the `Subject Alt Name` attribute of the load balancer's certificate that has been imported into CSA's keystore must also be set to the IP address of the system on which the load balancer is installed. If the load balancer's certificate does not contain the `Subject Alt Name` attribute or it is not set to the IP address of the system on which the load balancer is installed, you must regenerate and re-import the load balancer's certificate with the `Subject Alt Name` attribute set to the IP address of the system on which the load balancer is installed.

   b. Add and set the following property:

   ```
   csa.provider.ip=[LOAD_BALANCER_IP_ADDR]
   ```

2. Edit the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/classes/swagger.properties` file and set the following property:

   ```
   documentation.services.basePath=https://[LOAD_BALANCER_HOSTNAME]:[LOAD_BALANCER_CSA_HTTPS_PORT]/csa/rest
   ```

   For example:

   ```
   documentation.services.basePath=https://load_balancer.xyz.com:8443/csa/rest
   ```

Enable JNDI

To enable the Java Naming and Directory Interface (JNDI), complete the following steps:

1. Open the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/applicationContext.xml` file in a text editor.
2. Locate the comment `START HA Mode Configuration` and uncomment the following content:

   ```
   <jee:jndi-lookup id="channelGroup"
   jndi-name="java:jboss/clustering/group/server"
   expected-type="org.wildfly.clustering.group.Group"/>
   ```

3. If you modified the channel group, update the value of the `jndi-name` attribute to the new group name.
4. Save and exit the file.

Request a software license

CSA version 4.80 requires a software license. CSA licensing is based on the number of operating system instances (OSIs) being used in current, active subscriptions.

After the initial installation of CSA version 4.80, when you log in to the Cloud Service Management Console, a temporary 90-day trial license is activated. Once the trial license expires, you are limited to 25 OSIs. If you created more than 25 OSIs during the trial period, you cannot create any additional OSIs. You can add more licenses at any time to increase your OSI capacity.

When you request a software license, typically you supply the IP address of the system on which CSA is installed. However, in a clustered environment, use the IP address of the load balancer (`[LOAD_BALANCER_IP_ADDR]`) when requesting a software license. The license should be installed on only one node in the clustered environment.

For more information on managing software licenses, see the *Cloud Service Automation Configuration Guide*. For information on how to view, add, or delete a license, see the *Cloud Service Management Console Help*.

Configure Marketplace Portal redirection

One of the URLs that can be used to launch the Marketplace Portal (for example, `https://load_balancer.xyz.com:8443/mpp`) redirects the request from the JBoss server (the CSA controller) to the `Node.js` server (the Marketplace Portal). By default, the request is redirected to the same system on which CSA is installed. However, in a clustered environment, the request must be redirected to the load balancer.

Update the redirection to the load balancer node

To update the redirection to the load balancer node, do the following:

1. Open the `CSA_HOME/jboss-as/standalone/deployments/mpp.war/index.html` file in a text editor.
2. Locate the following line:

   ```
   <meta http-equiv="refresh" content="0;URL= https://[CSA_NODE_HOSTNAME]:[CSA_NODE_HTTPS_PORT]/"/>
   ```

3. Replace `[CSA_NODE_HOSTNAME]` with `[LOAD_BALANCER_HOSTNAME]` and `[CSA_NODE_HTTPS_PORT]` with `[LOAD_BALANCER_MPP_HTTPS_PORT]`. For example:

   ```
   <meta http-equiv="refresh" content="0;URL= https://[LOAD_BALANCER_HOSTNAME]:[LOAD_BALANCER_MPP_HTTPS_PORT]/"/>
   ```

   or

   ```
   <meta http-equiv="refresh" content="0;URL= https://load_balancer.xyz.com:8089/"/>
   ```

4. Save and exit the file.

Configure JBoss

Configure JBoss for use in a CSA clustered environment:

1. Open the `CSA_HOME/jboss-as/standalone/configuration/standalone-full-ha.xml` file in a text editor.
2. Locate the server property and configure a unique node name for the node. Locate

   ```
   <server xmlns="urn:jboss:domain:2.2" name="CHANGE ME!!">
   ```

   and set the name to *[CSA_NODE_NAME]*.

   For example:

   ```
   <server xmlns="urn:jboss:domain:2.2" name="csa_node1">
   ```

3. Update the messaging subsystem password. Change

```
<cluster-password>${jboss.messaging.cluster.password:CHANGE ME!!}</cluster-
password>
```

to

```
<cluster-password>password</cluster-password>
```

4. Locate the transactions subsystem and configure the node identifier for the `<core-environment>` property (set the node identifier to the unique node name you configured in step 2). Locate

```
<subsystem xmlns="urn:jboss:domain:transactions:2.0">
   <core-environment>
```

and add set the node identifier to *[CSA_NODE_NAME]*. For example:

```
<subsystem xmlns="urn:jboss:domain:transactions:2.0">
   <core-environment node-identifier="csa_node1">
```

5. Add the node's IP address to the public interface. Locate

```
<interface name="public">
```

and add the IP address `[CSA_NODE1_IP_ADDR]`. For example:

```
<interface name="public">
   <inet-address value="${jboss.bind.address:<CSA_Node_ip_Address>}"/>
</interface>
```

Configure a secure connection

Configure a secure connection (using a protocol such as TLS) on the CSA node for communication from the load balancer node and between each CSA node in the cluster.

1. To configure a secure connection between CSA and the load balancer node:

   a. If you have not already done so, copy the certificate from the load balancer node(`load_balancer.crt`) to the `CSA_HOME/jboss-as/standalone/configuration` directory.

   b. Import the certificate into the JVM on the CSA node using the following command:

   Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -file CSA_HOME\jboss-as\
standalone\configuration\load_balancer.crt -alias load_balancer_csa
-keystore "CSA_JRE_HOME\lib\security\cacerts"
```

Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -file CSA_HOME/jboss-as/
standalone/configuration/load_balancer.crt -alias load_balancer_csa
-keystore CSA_JRE_HOME/lib/security/cacerts
```

where `<CSA_JRE_HOME>` is the directory in which the JRE that is used by CSA is installed..

2. Copy and import the certificate of each CSA node to every other CSA node in the cluster:

   a. Copy the certificate of each CSA node to every other CSA node in the cluster. The certificate file on each CSA node is `CSA_HOME/jboss-as/standalone/configuration/jboss.crt`.

      For example, copy the certificates from csa_node2 and csa_node3 to csa_node1 to the directory `C:\CSA-Certificates` on Windows or `/tmp/CSA-Certificates`. Rename the certificate files with unique names, such as `jboss-csa_node2.crt` and `jboss-csa_node3.crt`.

   b. Import each certificate into the JVM of that CSA node.

      For example, on csa_node1, run the following commands:

      Windows:

```
"CSA_JRE_HOME\bin\keytool" -importcert -file C:\CSA-Certificates\jboss-csa_
node2.crt -alias csa_node2 -keystore "CSA_JRE_HOME\lib\security\cacerts"

"CSA_JRE_HOME\bin\keytool" -importcert -file C:\CSA-Certificates\jboss-csa_
node3.crt -alias csa_node3 -keystore "CSA_JRE_HOME\lib\security\cacerts"
```

      Linux:

```
CSA_JRE_HOME/bin/keytool -importcert -file /tmp/CSA-Certificates/jboss-csa_
node2.crt -alias csa_node2 -keystore CSA_JRE_HOME/lib/security/cacerts

CSA_JRE_HOME/bin/keytool -importcert -file /tmp/CSA-Certificates/jboss-csa_
node3.crt -alias csa_node3 -keystore CSA_JRE_HOME/lib/security/cacerts
```

Configure the Identity Management component on the CSA node

To configure the Identity Management component on the CSA node, complete the following steps:

1. Edit the following content in the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext.properties` file.
   Update the following values:
   - `idm.csa.hostname` and `idm.csa.audit.hostname` to *[LOAD_BALANCER_HOSTNAME]*
   - `idm.csa.port` and `idm.csa.audit.port` to *[LOAD_BALANCER_HTTPS_PORT]*

```
idm.csa.hostname = [LOAD_BALANCER_HOSTNAME]
idm.csa.port = [LOAD_BALANCER_CSA_HTTPS_PORT]
 .
 .
 .
# Properties for CSA Auditing Server
 .
 .
 .
idm.csa.audit.hostname = [LOAD_BALANCER_HOSTNAME]"/>
idm.csa.audit.port = [LOAD_BALANCER_HTTPS_PORT]"/>
```

   For example:

```
idm.csa.hostname = load_balancer.xyz.com
idm.csa.port = 8443
 .
 .
 .
# Properties for CSA Auditing Server
 .
 .
 .
idm.csa.audit.hostname = load_balancer.xyz.com"
idm.csa.audit.port = 8443"
```

2. Edit the following content in the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/applicationContext-security.xml` file.
   Update the following values:
   - `hostname` to *[LOAD_BALANCER_HOSTNAME]*
   - `port` to*[LOAD_BALANCER_CSA_HTTPS_PORT]*

```
<beans:bean id="idmConfig"
class="com.hp.ccue.identity.rp.IdentityServiceConfig">
   <beans:property name="protocol" value="https"/>
   <beans:property name="hostname" value="[LOAD_BALANCER_HOSTNAME]"/>
   <beans:property name="port" value="[LOAD_BALANCER_CSA_HTTPS_PORT]"/>
   <beans:property name="servicePath" value="idm-service"/> <!-- or hpcloud-
idm-service if you don't change the name of the WAR -->
   <beans:property name="integrationAcctUserName" value="idmTransportUser"/>
   <beans:property name="defaultTenant" value="#{systemEnvironment[CSA_ORG_
NAME_IDENTIFIER] ?:'${csa.orgName.identifier}'}"/>
</beans:bean>
```

For example:

```
<beans:bean id="idmConfig"
class="com.hp.ccue.identity.rp.IdentityServiceConfig">
   <beans:property name="protocol" value="https"/>
   <beans:property name="hostname" value="load_balancer.xyz.com"/>
   <beans:property name="port" value="8443"/>
   <beans:property name="servicePath" value="idm-service"/> <!-- or hpcloud-
idm-service if you don't change the name of the WAR -->
   <beans:property name="integrationAcctUserName" value="idmTransportUser"/>
   <beans:property name="defaultTenant" value="#{systemEnvironment[CSA_ORG_
NAME_IDENTIFIER] ?:'${csa.orgName.identifier}'}"/>
</beans:bean>
```

Configure SAML on CSA nodes in a clustered environment

This section describes how to configure SAML on CSA nodes when generating Identity Management component metadata in a clustered environment.

For more information about SAML configuration, see the *Cloud Service Automation Configuration Guide*.

To configure SAML on CSA nodes, complete the following steps:

> **Note: You must repeat these steps on each CSA node in the cluster.**

1. Edit the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/spring/applicationContext-saml.xml` file on Node 1 in CSA.
2. Locate the following line by searching for `entityBaseUR`, and uncomment the line:

   ```
   <!-- <property name="entityBaseURL" value="https://localhost/idm-service"/> -->
   ```

3. Replace `localhost` with `<LOAD_BALANCER_HOSTNAME>:<port>` for each node.

   This step makes sure that the load balancer host name and port are used when the Identity Management component metadata is generated.

   For example:

   ```
   <!-- <property name="entityBaseURL" value="https://<LOAD_BALANCER_HOSTNAME>:<port>/idm-service"/> -->
   ```

4. Change the URL value to the following:

   `https://lb.csacloud.local:8443/idm-service`

   > **Note:** Be sure the load balancer uses the `https` protocal to distribute the request.

5. Restart the Identity Management component on Node 1 by restarting the CSA service. See Restart the CSA service for instructions.
6. Repeat steps 1 through 5 for all nodes.
7. Download the Identity Management component Service Provider metadata from the load balancer URL:

   `https://lb.csacloud.local:8443/idm-service/saml/metadata`.

8. Upload this Identity Management component Service Provider metadata to the Identity Provider to replace the old Identity Management component Service Provider metadata file.
9. Restart the Identity Provider if required. See the vendor documentation for details.

Reconfigure the CSA service

Reconfigure the CSA service to start, restart, and stop CSA using the `standalone-full-ha.xml` configuration file.

> **Caution:** You must stop the CSA service before reconfiguring it.

To reconfigure the CSA **service on Windows, complete the following steps:**

1. Stop the `CSA` service:
   a. Navigate to the Services screen (**Control Panel > Administrative Tools > Services**).
   b. Right-click on the **CSA** service and select **Stop**.
2. Edit the `CSA_HOME\jboss-as\bin\service.bat` file.
   a. Locate the two occurrences of `standalone.bat`.
   b. Insert the command line option **-c standalone-full-ha.xml** into the `call standalone.bat > .r.lock >> run.log 2>&1` command line.

      For example:

      ```
      call standalone.bat -c standalone-full-ha.xml > .r.lock >> run.log 2>&1
      ```
3. Start the `CSA` service:
   a. Navigate to the Services screen (**Control Panel > Administrative Tools > Services**).
   b. Right-click on the **CSA** service and select **Start**.

To reconfigure the CSA service on Linux, complete the following steps:

> **Note:** If you upgrade from CSA 4.5 to CSA 4.7, copy the CSA 4.7 file under `CSA_HOME/scripts` to the `/etc/init.d` directory, so that you can start Elasticsearch.

1. Open a command prompt.
2. Stop the `CSA` service. Run the following command:

   service csa stop
3. Edit the `CSA_HOME/scripts/csa_env.conf` file:
   a. Locate the `Toggle below two lines to run CSA in HA mode` comment.
   b. Below this comment, comment out the `export CSA_DEPLOY_MODE=standalone.sh # Standalone Mode` line:

      ```
      #export CSA_DEPLOY_MODE=standalone.sh # Standalone Mode
      ```
   c. Uncomment the following line:

      ```
      export CSA_DEPLOY_MODE="standalone.sh -c standalone-full-ha.xml # HA Mode
      ```
4. Start the CSA service. Run the following command:

   service csa start

Configure global search

Global search allows you to find a certain service offering, service instance, or subscription by a meaningful keyword from the Marketplace Portal. For service offerings, global search finds the keyword in the name, description, option sets, options, and properties. For service instances and subscriptions, global search finds the keyword in the name, description, and instance properties (name and value).

Global search is enabled by default. Global search must be enabled to be available on the Marketplace Portal. See the *Cloud Service Automation Configuration Guide* for more information about enabling and disabling global search.

> **Caution:** Be sure to disable global search in a FIPS 140-2 compliant environment.

To configure global search, do the following:

1. Edit the `CSA_HOME/elasticsearch-1.6.1/config/elasticsearch.yml` file:
   a. Uncomment the `cluster.name` property and set it to a unique name that is shared by all the nodes in the cluster. That is, if you have more than one clustered environment on the same network, each clustered environment should have a unique cluster name. All the nodes in the single clustered environment will share the same cluster name.

      For example, `cluster.name: "csa_cluster"`

   b. Set the `node.name` property to a unique name. Each node in the cluster must have a unique node name.

      For example, `node.name: "node1"`

   c. Uncomment the `node.master` property and set it to **true** to make this node a master node. All nodes in the cluster should be a master node.

      For example, `node.master: true`

   d. Optionally, uncomment and set the `node.data` property. Refer to the comments in the file for information of how to combine this and the `node.master` property settings to suit the requirements of the node.

   e. Comment out the `node.local: true` property. When disabled, global search can find and communicate with other nodes on the network. If this property is left enabled, global search will not discover other nodes and will isolate itself from the network.

f. Verify that the following properties are set to these values (and if they are not set to these values, set them to these values):

```
transport.tcp.port: 9300
http.port: 9201
http.enabled: true
discovery.zen.ping.timeout: 5s
```

g. Set the `discovery.zen.ping.unicast.hosts` property to the IP addresses of the master nodes that perform discovery when new master or data nodes are started. Since all nodes in the cluster are master nodes, set this property to all IP addresses of the nodes in the cluster.

   For example, `discovery.zen.ping.unicast.hosts: ["111.222.333.444","111.222.333.445","111.222.333.446"]`

h. Locate the `Transport layer SSL` section and do the following:

   i. Verify that the following properties are set to these values (and if they are not set to these values, set them to these values):

```
searchguard.ssl.transport.node.keystore_type: JKS
searchguard.ssl.transport.node.keystore_password: changeit
searchguard.ssl.transport.node.truststore_type: JKS
searchguard.ssl.transport.node.truststore_password: changeit
```

   ii. Set the `searchguard.ssl.transport.node.keystore_filepath` property to the location of the CSA keystore. For example,

   Windows:

```
searchguard.ssl.transport.node.keystore_filepath: C:\Program
Files\HPE\CSA\jboss-as/standalone/configuration.keystore
```

   Linux:

```
searchguard.ssl.transport.node.keystore_filepath:
/usr/local/hpe/csa/jboss-as/standalone/configuration.keystore
```

   iii. Set the `searchguard.ssl.transport.node.truststore_filepath` property to the location of the CSA truststore. For example,

   Windows:

```
searchguard.ssl.transport.node.truststore_filepath: C:\Program
Files\HPE\CSA/openjre/lib/security/cacerts
```

Linux:

```
searchguard.ssl.transport.node.truststore_filepath:
/usr/local/hpe/csa/openjre/lib/security/cacerts
```

i. Locate the `REST layer SSL` section and do the following:

   i. Verify that the following properties are set to these values (and if they are not set to these values, set them to these values):

   ```
   searchguard.ssl.transport.http.keystore_type: JKS
   searchguard.ssl.transport.http.keystore_password: changeit
   searchguard.ssl.transport.http.truststore_type: JKS
   searchguard.ssl.transport.http.truststore_password: changeit
   ```

   ii. Set the `searchguard.ssl.transport.http.keystore_filepath:` property to the location of CSA's keystore. For example,

   Windows:

   ```
   searchguard.ssl.transport.http.keystore_filepath: C:\Program
   Files\HPE\CSA/jboss-as/standalone/configuration/.keystore
   ```

   Linux:

   ```
   searchguard.ssl.transport.http.keystore_filepath:
   /usr/local/hpe/csa/jboss-as/standalone/configuration/.keystore
   ```

   iii. Set the `searchguard.ssl.transport.http.truststore_filepath` property to the location of the CSA truststore. For example,

   Windows:

   ```
   searchguard.ssl.transport.http.truststore_filepath: C:\Program
   Files\HPE\CSA/openjre/lib/security/cacerts
   ```

   Linux:

   ```
   searchguard.ssl.transport.http.truststore_filepath:
   /usr/local/hpe/csa/openjre/lib/security/cacerts
   ```

j. Save and exit the file.

2. In the `csa.properties` file, set the value of the `csa.provider.msvc.hostname` property with the local node `FQDN`.

3. In the `CSA_HOME/csa-search-service/app.json` file, set the values of the following properties:

   `ccue-basic-server.host` with the local node `FQDN`

`msvc-basic-search.searchidmURL` should point to the load balancer FQDN and load balancer port 8443.

4. If the cluster setup is using default CSA (self-signed) certificates, complete the following step. (This step is not required if the cluster runs valid certificates signed by a common CA).

   In the `csa-search-service\app.json` file, find the following keys and change the values to **false**:

   `msvc-basic-search.strictSSL`

   `rejectUnauthorized`

5. Verify that the following High Availability configurations in the `CSA_HOME/csa-search-service/app.json` file, are maintained after the installation of CSA:

   a. `msvc-basic-search.idmURL:[LOAD_BALANCER_HOSTNAME]:[LOAD_BALANCER_HTTPS_PORT]/idm-service` should point to the load balancer.

      For example:

      `idmURL`: `https://http-loadbalancer.csapcoe.hp.com:8443/idm-service`

      where port 8443 is the load balancer port which was configured manually during installation.

   b. `cert.ca` should point to the load balancer certificate:

      For example:

      `"ca": "C:/Program Files/HPE/CSA/jboss-as/standalone/configuration/loadbalancer_csa.crt".`

      The name of the `crt` file cannot remain as `jboss.crt` which is set as the default.

6. Create the security key on one node and copy it to the other nodes in the cluster. The security key is used to authenticate the communication between the nodes in the cluster when sharing the shards and replicas of the inventory index. The security key must be the same on all nodes in the cluster.

   a. On a `CSA_Node` (for example, `csa_node1`), complete the following steps:

      Windows:

      Stop then start the Elasticsearch 1.6.1 service:

      i. Navigate to the Services screen (**Control Panel > Administrative Tools > Services**).

      ii. Right-click on the **Elasticsearch 1.6.1** service and select **Restart**.

      Linux:

      Stop, then start CSA. Open a command prompt and type:

```
service csa restart
```

b. After the service has restarted on `csa_node1`, copy the `CSA_HOME/elasticsearch-1.6.1/searchguard_node_key.key` file from `csa_node1` to all other nodes in the cluster. Copy the file to the same directory (`CSA_HOME/elasticsearch-1.6.1/`) and use the same file name on the other nodes.

c. On all nodes in the cluster except `csa_node1`, complete the following steps:

Windows:

Restart the the Elasticsearch 1.6.1 service:

i. On all nodes in the cluster except `csa_node1`, navigate to the Services screen (**Control Panel > Administrative Tools > Services**).

ii. Right-click on the **Elasticsearch 1.6.1** service and select **Restart**.

Linux:

Restart CSA on all nodes in the cluster except `csa_node1`. Open a command prompt and type:

```
service csa restart
```

7. Complete the following steps:

Windows:

Stop the HPE Search Service and CSA services and then start them:

a. Navigate to the Services screen (**Control Panel > Administrative Tools > Services**).

b. Right-click on the **HPE Search Service** service and select **Stop**.

c. Right-click on the **HPE Cloud Service Automation** service and select **Stop**.

d. Right-click on the **HPE Search Service** service and select **Start**.

e. Right-click on the **HPE Cloud Service Automation** service and select **Start**.

Linux:

Restart CSA. Open a command prompt and type:

```
service csa restart
```

8. If you changed the `cluster.name`, you must create new indexes. Do the following:

> **Note:** On Windows, the Elasticsearch 1.6.1 service or on Linux, CSA must be running.

a. Open a command prompt and navigate to `CSA_HOME/csa-search-service/bin/`.

b. Run the following command:

Windows:

```
"CSA_HOME\node.js\node.exe" create-index.js
```

Linux:

```
CSA_HOME/node.js/bin/node create-index.js
```

If displayed, ignore the following errors:

```
ERROR: Error connecting to Elasticsearch server. Cannot create index
catalog. Error: DEPTH_ZERO_SELF_SIGNED_CERT
ERROR: Error connecting to Elasticsearch server. Cannot create index
inventory. Error: DEPTH_ZERO_SELF_SIGNED_CERT
```

It may take a few minutes for the first CSA artifact to be indexed.

Configure the TCP communication channel on JGroups

JBoss uses JGroups for communication between nodes in order to establish the cluster and manage membership of nodes in the cluster. By default, the JGroups subsystem on JBoss is configured to communicate through IP multicast messages using UDP. If the environment that you are using to set up the cluster does not support multicast messaging, the JGroups subsystem may alternatively be configured to use multiple TCP unicast messages.

To configure the TCP communication channel on JGroups, do the following:

1. Open the `CSA_HOME/jboss-as/standalone/configuration/standalone-full-ha.xml` file in an editor.
2. Update the jgroups `subsystem` default stack from `udp` to `tcp`. Change:

   ```
   <subsystem xmlns="urn:jboss:domain:jgroups:2.0" default-stack="udp">
   ```

   to

   ```
   <subsystem xmlns="urn:jboss:domain:jgroups:2.0" default-stack="tcp">
   ```

3. Locate the TCP stack and do the following:

a. Replace `<protocol socket-binding="jgroups-mping" type="MPING"/>` with:

```
<protocol type="TCPPING">
   <property name="initial_hosts">[LIST_OF_INITIAL_HOSTS]</property>
      <property name="num_initial_members">[NUMBER_OF_INITIAL_HOSTS]
</property>
      <property name="port_range">1</property>
      <property name="timeout">2000</property>
</protocol>
```

where:

- *[LIST_OF_INITIAL_HOSTS]* is a comma-separated list of nodes (IP address and port) that define the cluster. It is recommended that you list all known nodes in the CSA controller cluster. Other nodes that are not listed may join the cluster and you may remove a node from the list at any time. However, at least one initial host (a node in the list of initial hosts) must be running in order for other nodes (that are not included in this list) to join the cluster. The more initial hosts listed means that there is a greater chance an initial host is running so that an unlisted node may join the cluster (if no initial hosts are running, no unlisted nodes may join the cluster). Once the cluster is running, if you update the list of initial hosts, all nodes in the cluster must be restarted. The following are examples of a list of three initial hosts: *[CSA_NODE1_IP_ADDR]*[7600],*[CSA_NODE2_IP_ADDR]*[7600],*[CSA_NODE3_IP_ADDR]*[7600] or 111.222.333.444[7600],111.222.333.445[7600],111.222.333.446[7600]
- *[NUMBER_OF_INITIAL_HOSTS]* is the number of initial hosts specified in the cluster.

For example:

```
<protocol type="TCPPING">
   <property name="initial_hosts">111.222.333.444[7600],111.222.333.445
[7600],
111.222.333.446[7600]</property>
      <property name="num_initial_members">3</property>
      <property name="port_range">1</property>
      <property name="timeout">2000</property>
</protocol>
```

A TCP-based channel may be less efficient than its UDP counterpart as the size of the cluster increases beyond four to six nodes.

b. Under `<protocol type="pbcast.NAKACK2"/> ---<Stack_name="tcp">` add:

```
<protocol type="pbcast.NAKACK2">
<property name="use_mcast_xmit">false</property>
<property name="use_mcast_xmit_req">false</property>
</protocol>
```

4. Add the node's IP address to the public interface. Locate:

   ```
   <interface name="public">
   ```

   and add the IP address `[CSA_NODE1_IP_ADDR]`. For example:

   ```
   <interface name="public">
      <inet-address value="${jboss.bind.address:<CSA_Node_ip_Address>}"/>
   </interface>
   ```

5. Reconfigure the CSA service as follows:

   Windows:

   In the `\CSA_HOME\bin\service.bat` file:

   a. Locate the two occurrences of `standalone.bat`.

   b. Replace `call standalone.bat > .r.lock >> run.log 2>&1`

      with

      ```
      call standalone.bat -c standalone-full-ha.xml > .r.lock >> run.log 2>&1
      ```

      > **Note:** You do not need to specify the `-u` *`[MULTICAST_ADDRESS]`* option.

   Linux:

   In the `CSA_HOME/scripts/csa_env.conf` file:

   a. Locate the `Toggle below two lines to run CSA in HA mode` comment.

   b. Comment out `export CSA_DEPLOY_MODE=standalone.sh # Standalone Mode`.

   c. Add `export CSA_DEPLOY_MODE="standalone.sh -c standalone-full-ha.xml" # HA Mode`.

      > **Note:** You do not need to specify the `-u` *`[MULTICAST_ADDRESS]`* option.

Configure Single Sign-On

If you have integrated Single Sign-On (SSO) between CSA and another application (such as Operations Orchestration), you must configure SSO on the CSA node.

1. Open the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/hpssoConfig.xml` file in a text editor.
2. Locate the following content:

```
<onFailure>
    .
    .
    .
   <action name="redirectToAP">
      <targetUrl>https://[CSA_NODE_HOSTNAME]:[CSA_NODE_PORT]
/csa/login</targetUrl>
   </action>
```

3. Replace `[CSA_NODE_HOSTNAME]` and `[CSA_NODE_PORT]` as follows:

   Replace `[CSA_NODE_HOSTNAME]` with `LOAD_BALANCER_HOSTNAME`) and the virtual host port for the CSA nodes (`LOAD_BALANCER_CSA_HTTPS_PORT`). For example:

```
<onFailure>
    .
    .
    .
   <action name="redirectToAP">
      <targetUrl>https://load_balancer.xyz.com:8443/csa/login</targetUrl>
   </action>
```

4. Locate the `initString` value in the `crypto` element. The `initString` setting for CSA must be the same value for all nodes in the cluster and any applications (such as Operations Orchestration) that are integrated with Single Sign-On. The `initString` value represents a secret key and should be treated as such in your environment.
5. Copy the `initString` value to the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/hpssoConfig.xml` file on all other nodes in the cluster.
6. Copy the `initString` value to the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/hpssoConfig.xml` file on this and all other nodes in the cluster.
7. Configure this `initString` value in any applications that are integrated with CSA using Single Sign-On, including Operation Orchestration.

Workflow Designer Configuration - SSO

1. Open the `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/hpssoConfig.xml` file on first CSA node in text editor and locate the `initString` value in crypto element.

   Example:

   ```
   <crypto initString="2kDcHB0eOHrHcAGeArIPr7TNfuivOpKqjj29SwKOQIoI"
   cipherType="symmetricBlockCipher" engineName="AES"

   paddingMode="CBC" keySize="256" encodingMode="Base64Url"
   algorithmPaddingName="PKCS7Padding" checkIntegrity="disabled"

   cryptoSource="lw" directKeyEncoded="false" directKeyEncoding="Hex"
   jcePbeAlgorithmName="PBEWithHmacSHA1"

   jcePbeMacAlgorithmName="PBEWithHmacSHA1" macAlgorithmName="SHA1"
   macKeySize="256" macPbeCount="20" macType="hmac"

   pbeCount="20" pbeDigestAlgorithm="SHA1"/>
   ```

   If not already done, copy this `initString` to `CSA_HOME/jboss-as/standalone/deployments/idm-service.war/WEB-INF/hpssoConfig.xml` of all other CSA nodes.

   Create encrypted version of `initString` with encrypt-password script. When prompted for password, provide the `initString` to the script:

   ```
   sh# cd $CSA_HOME/workflow-designer/designer/bin

   sh# ./encrypt_password

   Password (typing will be hidden):

   Confirm password (typing will be hidden):

   {ENCRYPTED}xxts33/07Dtyz0iZ3eOQhzFVuqXvZ7KK6wDNm1A4E5+byAxlDZ+lHzwNRPvLgqXf

   sh#
   ```

2. Edit the file `CSA_HOME/workflow-designer/designer/var/securitysecured.properties` on every CSA node.

3. Add/Edit the `lwsso.initString property` with the encrypted `initString` from encrypt-password sript:

   ```
   #This is for limit the size of single CP upload, default 200MB

   upload.max.fileSize.limit = 209715200
   ```

```
#This is for limit the number of parallel CP creation

max.parallel.cp.creation = 30

#This is for limit the number of parallel CP upload

max.parallel.cp.upload = 50

is.secured.cookie = true

lwsso.initString = {ENCRYPTED}
xxts33/07Dtyz0iZ3eOQhzFVuqXvZ7KK6wDNm1A4E5+byAxlDZ+lHzwNRPvLgqXf
```

4. Edit file `CSA_HOME/workflow-designer/designer/tomcat/conf/server.xml` on every CSA node. Locate Engine element and add `jvmRoute` property with unique node name for every workflow-designer node:

```
Engine element in server.xml file:

<Engine defaultHost="localhost" name="Catalina">


On first node change to:

<Engine defaultHost="localhost" name="Catalina" jvmRoute="ood1">


On second node change to:

<Engine defaultHost="localhost" name="Catalina" jvmRoute="ood2">

...
```

5. Restart workflow-designer on every node using command: `CSA_HOME/workflow-designer/designer/bin/designer restart`

Configure the Marketplace Portal node

This chapter describes how to install, upgrade, and configure a remote Marketplace Portal instance on the Marketplace Portal node (for example, MPP_Node1, MPP_Node2, or MPP_Node3) in a cluster configured for high availability. The Marketplace Portal should be installed as a remote instance on a Marketplace Portal node in its own clustered environment.

To configure the Marketplace Portal, do the following:

- Install or upgrade the remote Marketplace Portal instance
- Configure the remote Marketplace Portal instance

Install the remote Marketplace Portal instance

Install a remote instance of the Marketplace Portal on each Marketplace Portal node, as described in the *Cloud Service Automation Installation Guide* with the following limitations:

- You must install the same version of CSA on each node (including the CSA nodes).
- When selecting a location in which to install the Marketplace Portal, select the same location for all Marketplace Portal nodes.
- When configuring the CSA Host, use the fully-qualified domain name of the following:
  - Load_Balancer node (for example, `load_balancer.xyz.com`) or *[LOAD_BALANCER_HOSTNAME]*
- When configuring the CSA Port, use the following port:
  - Load balancer installed on the Load_Balancer node (for example, 8443 or *[LOAD_BALANCER_CSA_HTTPS_PORT]*).

The *Cloud Service Automation Installation Guide* can be downloaded from the HPE Software Support Web site at https://softwaresupport.hpe.com/manuals (this site requires a Passport ID).

Upgrade the remote Marketplace Portal instance

Upgrade the remote Marketplace Portal instance on each Marketplace Portal node as described in the *Cloud Service Automation Upgrade Guide* .

Configure the remote Marketplace Portal instance

To configure the remote Marketplace Portal instance on each Marketplace Portal node, complete the following steps:

1. If you have not done so already, copy the certificate of the load balancer from the Load_Balancer node (for example, `load_balancer.crt`) to the `CSA_HOME/portal/conf/` directory on the Marketplace Portal node.
2. Edit the following content in the `CSA_HOME/portal/conf/mpp.json` file:
   - For the provider, update the `url` attribute value to use *`[LOAD_BALANCER_HOSTNAME]`* and *`[LOAD_BALANCER_CSA_HTTPS_PORT]`* and `ca` to use the location of the certificate of the load balancer. For example:

     ```
     "provider": {
        "url": "https://[LOAD_BALANCER_HOSTNAME]:[LOAD_BALANCER_CSA_HTTPS_PORT]"
        "ca": "CSA_HOME/portal/conf/load_balancer.crt"
         .
         .
         .
     },
     ```

     or

     ```
     "provider": {
        "url": "https://load_balancer.xyz.com:8443"
        "ca": "CSA_HOME/portal/conf/load_balancer.crt"
         .
         .
     ```

```
    .
},
```

- For the idmProvider, update the values of the `url` attribute to use *[LOAD_BALANCER_HOSTNAME]* and *[LOAD_BALANCER_CSA_HTTPS_PORT]*, `returnUrl` to use *[LOAD_BALANCER_HOSTNAME]* and *[LOAD_BALANCER_MPP_HTTPS_PORT]*, and `ca` to use the location of the certificate of the load balancer. For example:

```
"idmProvider": {
   "url": "https://[LOAD_BALANCER_HOSTNAME]:[LOAD_BALANCER_CSA_HTTPS_PORT]",
   "returnUrl": "https://[LOAD_BALANCER_HOSTNAME]:[LOAD_BALANCER_MPP_HTTPS_
PORT]",
   "ca": "CSA_HOME/portal/conf/load_balancer.crt"
    .
    .
    .
},
```

or

```
"idmProvider": {
   "url": "https://load_balancer.xyz.com:8443",
   "returnUrl": "https://load_balancer.xyz.com:8089",
   "ca": "CSA_HOME/portal/conf/load_balancer.crt"
    .
    .
    .
},
```

3. Restart the HPE Marketplace Portal service:

   Windows:

   a. Navigate to the Services screen (**Control Panel > Administrative Tools > Services**).

   b. Right-click on the **hpemarketplaceportal** service and select **Restart**.

   Linux:

   Open a command prompt and type:

```
service mpp restart
```

Common tasks

This chapter provides information on how to perform common tasks.

Tasks include:

Start the CSA service

> **Caution:** If you have not already done so, reconfigure the CSA service to start and stop CSA using the `standalone-full-ha.xml` configuration file (you should have completed these steps when you configured the CSA node).

To start CSA **on Windows:**

1. On the server that hosts CSA, navigate to **Control Panel** > **Administrative Tools** > **Services**.
2. Right-click on the **HPE Cloud Service Automation** service and select **Start**.

To start CSA **on Linux:**

On the server that hosts CSA, type the following:

```
service csa start
```

Stop the CSA service

> **Caution:** If you have not already done so, reconfigure the CSA service to start and stop CSA using the `standalone-full-ha.xml` configuration file (you should have completed these steps when you configured the CSA node).

To stop CSA **on Windows:**

1. On the server that hosts Cloud Service Automation, navigate to **Control Panel** > **Administrative Tools** > **Services**.
2. Right-click on the **HPE Cloud Service Automation** service and select **Stop**.

To stop CSA **on Linux:**

On the server that hosts Cloud Service Automation, type the following:

```
service csa stop
```

Start the Marketplace Portal

To start the HPE Marketplace Portal service on Windows:

1. Navigate to the Services screen (**Control Panel > Administrative Tools > Services**).
2. Right-click on the **hpemarketplaceportal** service and select **Start**.

To start Marketplace Portal on Linux:

On the system that hosts CSA, open a command prompt and type:

```
service mpp start
```

Stop the Marketplace Portal

To stop the HPE Marketplace Portal service on Windows:

1. On the server that hosts Marketplace Portal, navigate to **Control Panel** > **Administrative Tools** > **Services**.
2. Right-click on the **hpemarketplaceportal** service and select **Stop**.

To stop Marketplace Portal on Linux:

On the server that hosts Marketplace Portal, type:

```
service mpp stop
```

Launch the Cloud Service Management Console

Launch the Cloud Service Management Console through the load balancer by opening one of the following URLs in a supported web browser:

- `http://[LOAD_BALANCER_HOSTNAME]:[LOAD_BALANCER_HTTP_PORT]/csa`
  For example, `http://load_balancer.xyz.com:8080/csa`
- `https://[LOAD_BALANCER_HOSTNAME]:[LOAD_BALANCER_CSA_HTTPS_PORT]/csa`
  For example, `https://load_balancer.xyz.com:8443/csa`

Launch the Marketplace Portal

To launch the default Marketplace Portal, open the following URL in a supported web browser:

`https://[LOAD_BALANCER_HOSTNAME]:8443/mpp`
For example, `https://load_balancer.xyz.com:8443/mpp`

The organization associated with the default Marketplace Portal is defined in the `CSA_HOME/portal/conf/mpp.json` file. By default, this is the sample organization that is installed with CSA (`CONSUMER`). To modify the organization associated with the default Marketplace Portal, modify the `defaultOrganizationName` property value by setting it to the *<organization_identifier>* of the desired organization, where *<organization_identifier>* is the unique name that CSA assigns to the organization, based on the organization display name (the organization identifier can be found in the General Information section of the **Administration** tile of the Cloud Service Management Console).

To launch an organization's Marketplace Portal, open one of the following URLs in a supported web browser:

`http://[LOAD_BALANCER_HOSTNAME]:[LOAD_BALANCER_HTTP_PORT]/org/<organization_identifier>`
For example, `http://load_balancer.xyz.com:8080/org/ORGANIZATION_A`

where *<organization_identifier>* is the unique name that CSA assigns to the organization, based on the organization display name (the organization identifier can be found in the General Information section of the **Administration** tile of the Cloud Service Management Console)

> **Caution:** Do not launch more than one organization-specific Marketplace Portal from the same browser session. For example, if you launch ORGANIZATION_A's Marketplace Portal in a browser, do not open a tab or another window from that browser and launch ORGANIZATION_B's Marketplace Portal. Otherwise, the user who has logged in to the Marketplace Portal launched for ORGANIZATION_A will start to see data for ORGANIZATION_B.
>
> Instead, start a new browser session to launch another organization's Marketplace Portal.

Cloud Service Management Console or Marketplace Portal access in a CSA clustered environment

When accessing the Cloud Service Management Console or Marketplace Portal in a CSA clustered environment, you must do the following:

1. Open the `CSA_HOME/jboss-as/standalone/deployments/csa.war/WEB-INF/flex/services-config.xml` file.
2. Update the endpoint class value for the `csa-secure-amf` channel to `flex.messaging.endpoints.AMFEndpoint` as follows:

   Change from:

   ```
   <channel-definition id="csa-secure-amf" class="mx.messaging.channels.SecureAMFChannel">
      <endpoint url="https://{server.name}:{server.port}/{context.root}/messagebroker/amfsecure"
   class="flex.messaging.endpoints.SecureAMFEndpoint" />
      <properties>
         <add-no-cache-headers>false</add-no-cache-headers>
      </properties>
   </channel-definition>
   ```

   To:

   ```
   <channel-definition id="csa-secure-amf" class="mx.messaging.channels.SecureAMFChannel">
      <endpoint url="https://{server.name}:{server.port}/{context.root}/messagebroker/amfsecure"
   class="flex.messaging.endpoints.AMFEndpoint" />
         <properties>
            <add-no-cache-headers>false</add-no-cache-headers>
         </properties>
   </channel-definition>
   ```

> **Note:** If you do not follow these steps, when you access the Cloud Service Management Console in a clustered environment, the following error may be generated in the log file. Flex-related errors may also occur if these changes are not made.
>
> `flex.messaging.security.SecurityException: Secure endpoint '/messagebroker/amfsecure' must be contacted via a secure protocol`

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Cluster Configuration using a Load Balancer (Cloud Service Automation 4.80)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to clouddocs@hpe.com.

We appreciate your feedback!