HPE Unified OSS Console

Installation and Configuration Guide

Release 2.3.0
Part number: JP699AAE
Edition: 1.1

Hewlett Packard Enterprise

Notices

Legal notice

Printed in the US

Trademarks

Contents

List of figures

Preface

About this guide

This guide describes how to install the product on the various supported platforms.

Audience

Here are some recommendations based on possible reader profiles:

- Administrators
- Integrators

Software Versions

The term UNIX is used as a generic reference to the operating system, unless otherwise specified.

Product Version	Supported Operating Systems
Unified OSS Console V2.3	Red Hat Enterprise Linux Server release 6.5 Red Hat Enterprise Linux Server release 7.2

Figure 1: Software Versions

Associated Documents

The following documents contain useful reference information:

- HPE Unified OSS Console V2.3 – User Guide
- HPE Unified OSS Console V2.3 – Release Notes
- HPE Unified OSS Console V2.3 – Development Guide
- HPE Unified OSS Console V2.3 – Integration Guide

Support

Please visit our HPE Software Support Online Web site at https://softwaresupport.hp.com/

For contact information, and details about HPE Software products, services, and support.

The Software support area of the Software Web site includes the following:

- Downloadable documentation.
- Troubleshooting information.
- Patches and updates.
- Problem reporting.
- Training information.
- Support program information.

Chapter 1
Code Signing

HPE Unified OSS Console is digitally signed and accompanied by a set of GnuPG keys.

NOTE: HPE strongly recommends using signature verification to ensure the authenticity and the integrity of their products.

For more information about signature verification procedure, please visit:
https://h20392.www2.hpe.com/portal/swdepot/displayProductInfo.do?productNumber=HPLinuxCodeSigning2

1.1 Signature verification

Before proceeding with signature verification process, please make sure that gpg and rpm tools are installed in your system.

1.1.1 Import HPE public key

Perform the following steps to import the public key that is needed for verifying the delivered products. These steps requires to be logged as root.

1. Create a temporary directory where the public keys will be stored:

```
# mkdir -p signcheck
```

2. Download the public keys:

```
# wget -P signcheck/ https://ftp.hp.com/pub/keys/HPE-GPG-Public-Keys.tar.gz
```

3. Import the public key for rpm

```
# rpm --import signcheck/2BAF2262.pub
```

4. Import the public key for gpg:

```
# gpg --import signcheck/2BAF2262.pub
```

5. Configure the level of trust for the imported key:

```
# gpg --edit-key 2BAF2262
```

Then type the command "`trust`", and select "5" for trusting the key ultimately. Confirm and type `quit` to exit.

1.1.2 HPE Unified Console Verification

To verify the integrity of the delivered product, perform the following steps:

1. Untar the delivered product.

```
$ tar xvf UOCV2.3.0-MR.tar
```

2. Go to the created directory uoc2_kit and check the code signing of the RPM using the following command:

```
$ rpm -Kv uoc-2.3.0-MR.x86_64.rpm
```

3. Check the command output. If signature verification completed successfully, the command output will contain the following lines:

```
Header V3 RSA/SHA256 Signature, key ID 2baf2262: OK
Header SHA1 digest: OK (675cd519013a4c34f07376e97d96456d8f30e827)
V3 RSA/SHA256 Signature, key ID 2baf2262: OK
MD5 digest: OK (4e9cbf55ef3337f3c44afc17c1682a2d)
```

4. Check the command outputs. If signature verification completed successfully, the command output will contain the following lines:

```
Header V3 RSA/SHA256 Signature, key ID 2baf2262: OK
Header SHA1 digest: OK (e6293719273b8ada54288216b190a481eae5c213)
V3 RSA/SHA256 Signature, key ID 2baf2262: OK
MD5 digest: OK (b5e22fd6582d04be543433796d632955)
```

5. Check the command outputs. If signature verification completed successfully, the command output will contain the following lines:

```
Header V3 RSA/SHA256 Signature, key ID 2baf2262: OK
Header SHA1 digest: OK (c3455512a59180340e13a644b799ab7db17be953)
V3 RSA/SHA256 Signature, key ID 2baf2262: OK
MD5 digest: OK (7099ce5241a4497eee363639531cd6ff)
```

1.1.3 HPE Unified Console SDK Verification

To verify the integrity of the delivered product, perform the following steps:

1. Take the signature (.sig) file shipped along with the product and use the following command:

```
$ gpg --verify UOCV2.3.0-MR-SDK.zip.sig UOCV2.3.0-MR-SDK.zip
```

2. Check the command output. If signature verification completed successfully, the command output will contain the following lines:

```
gpg: Signature made Fri 13 May 2016 05:56:49 PM CEST using RSA key ID
2BAF2262

gpg: Good signature from "Hewlett Packard Enterprise Company RSA-2048-14
<signhp@hpe.com>"
```

3. Check the command output. If signature verification completed successfully, the command output will contain the following lines:

```
gpg: Signature made Wed 09 Nov 2016 12:18:28 PM CET using RSA key ID
2BAF2262

gpg: Good signature from "Hewlett Packard Enterprise Company RSA-2048-14
<signhp@hpe.com>"
```

Chapter 2 Introduction

It is recommended to read the Release Notes document before proceeding with the installation.

UOC V2 requires an installation on a Linux Server (UOC V2 Server), and web browsers as client to this server. It could be a PC, laptop or mobile device.

NOTE: The UOC V2 Server can run and is supported on virtual machines

The HPE Unified Console software kit is targeted for Red Hat Enterprise Linux V6.5 and V7.2 (x86-64) only.

The software may run properly on many other Linux distributions but no support is provided to them.

Packaging of UOC has been designed to provide a high level of security. All processes have minimal limited privileges and cannot access to information or files from other processes. It will be required to create several administrator groups (uoc, couchdb, redis….) and non login users. These groups can be associated to login users to ease accountability and tracking of commands.

All the administrative commands can be tuned using the Sudo Policy.

IMPORTANT: The customer is responsible for installing all Operating System and pre-required Open Sources security patches to secure the Unified Console deployment.

Unified Console needs pre-requisite before its installation (these are not included in the media for contractual reasons)

- Apache CouchDB V1.6.x used internally by UOC to store its GUI documents and definitions.
- NodeJS V4.6: JavaScript runtime built on Chrome's V8 JavaScript engine
- Redis V3.2.4 (optional, mandatory only if you deployed multiple UOC servers). It is used as notification server to push real-time notifications through all instances of UOC.
- SAML Identity Provider to provide SSO, SAML V2 authentication (e.g. PicketLink, Keycloak, etc.). Note it is possible to use a local authentication system, but it is not dedicated to production (test and demo only)

2.1 Apache CouchDB

Apache CouchDB is a non-relational database ("NoSQL"), open-source, distributed (incremental, bidirectional replication), schema-free. A CouchDB database is a collection of documents; each document is a bunch of string "keys" and corresponding "values" (which can be numbers, strings, lists, dates ...).

CouchDB offers us these features:

- Easy replication of a database across multiple server instances
- Fast indexing and retrieval
- REST-like interface for document insertion, updates, retrieval and deletion
- JSON-based document format

See http://couchdb.apache.org/ for details. The CouchDB server can be accessed remotely through a web-based administration console or a RESTful API.

In UOC, CouchB is used to store all internal documents and definition like Workspaces, Views, Launch Definition, Permission, roles, users (local authentication only), categories, etc.

IMPORTANT:
CouchDB v1.6.0 is available for RHEL6.5 and v1.6.1 for RHEL7.x. Based on your system, check carefully the version you need to install and contact the support team if you have issue finding the right package.

NOTE: CouchDB is a pre-requisites not supported by HP Enterprise.

CouchDB (http://**couchdb**.apache.org)is a pre-requisite not supported by HPE.

Please go to the official web sites for details, documentation, patches and updates. It is strongly recommended to subscribe to Node Security news to monitor the security issues

2.2 NodeJS

(https://nodejs.org)

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries

The UOC core package and NodeJS must be installed on the same system.

NOTE: NodeJS is a pre-requisites not supported by HP Enterprise

Node (http://**nodejs**.org)is a pre-requisites not supported by HPE.

Please go to the official web sites for details, documentation, patches and updates. It is strongly recommended to subscribe to Node Security news to monitor the security issues

NOTE: NodeJS version and security

For security reasons, it is very important to install **NodeJS version 4.x** (recommended v4.6) to make sure all SSL vulnerabilities is fixed in UOC Server. This branch is the LTS version you have to rely on.

2.3 Redis

Redis pub/sub: Redis is an open source (BSD licensed), in-memory data structure store, used as database, cache and message broker. Redis includes a Publish/Subscribe messaging system that has been combined with Socket.io to allow real-time communication across multiple servers.

Only one Redis server is used by all UOC servers. (http://redis.io)

You need first to check how many UOC Servers will be running in your solution.

- It could be an optional component in the solution if there is ONLY 1 UOC Server. Redis configuration should be disabled (active: false)
- It is a mandatory component if there are multiple UOC server instances. In this case, Redis is used to allow communication between clients connected on different servers. Redis pub/sub is used as a common communication channel.

In the case multiple UOC server instances are installed, UOC relies on Redis 3.2.x and above. A Redis server must therefore be installed on the system before installing UOC.

Only one Redis server must be used by all UOC servers.

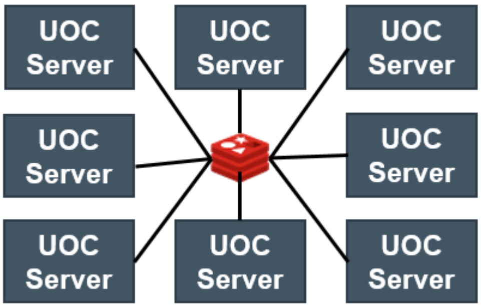


Figure 2 - UOC Servers and Redis

In this example, there are 8 UOC servers linked together to the same notification server (Redis). They are all able to push notification to workspaces, views, us and specific roles across these servers and notify in real-time the one or several end users.

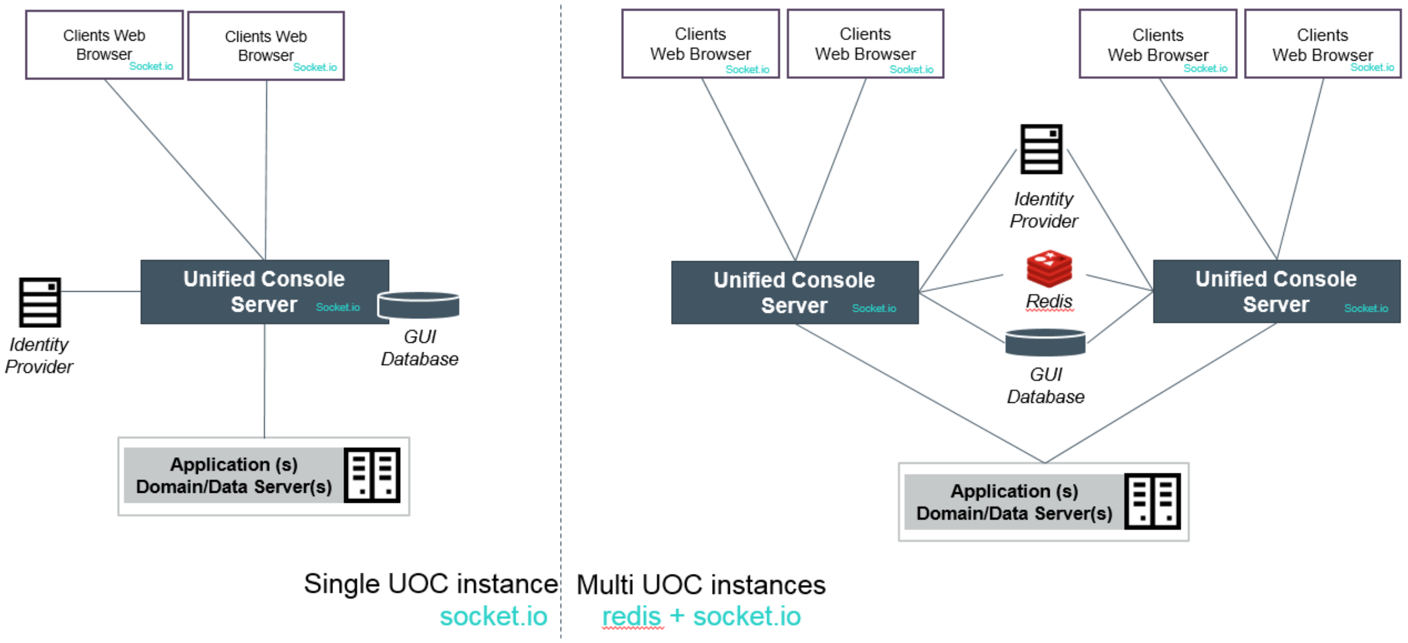


Figure 3 - UOC Servers and Redis: one UOC Servers vs multiple UOC Servers

2.4 Unified Console (UOC)

Unified Console is a presentation server bringing a high level of unification and customization.

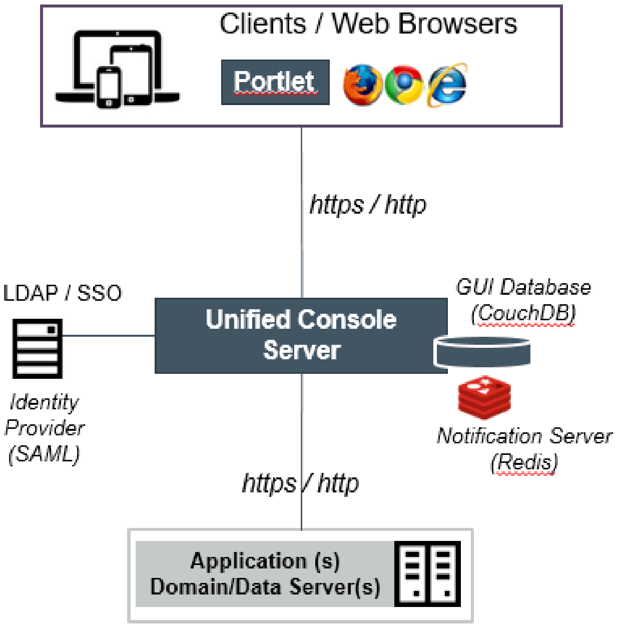


Figure 4: Unified Console Overview

UOC Server leverages NodeJS technology and is extremely scalable and supports multiple ways of deployment in order to adjust the volume of concurrent users and data.

Here are some examples of classic deployments. These ones need to be refined according to projects and use cases.

- Monolithic
- Multiples Instances
- Multiples Servers

NOTE: A chapter is dedicated to load balancing where you can find additional information to support all these deployment use cases

2.4.1 Monolithic

It is the simplest installation using only one machine to install all kits:

- Apache CouchDB
- UOC V2.3
- Plugin(s) (e.g. OSS Analytics Plugin)
- Domain or Data Server(s) (ex: OSS Analytic Server, OSS Assurance Monitoring)

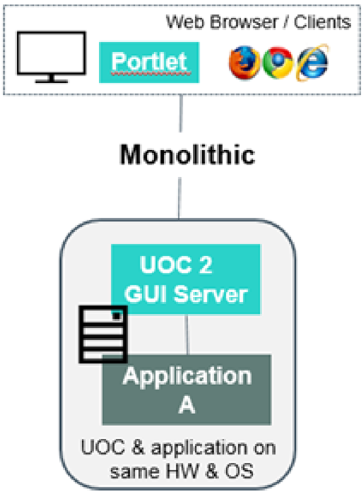


Figure 5: Deployment Examples – Monolithic

NOTE: There is no need of a notification Server and it is not mandatory to install Redis. Client/Server notification will be working directly.

2.4.2 Multiple instances

In this deployment, several instances of the UOC V2 servers are running on one machine so as to use all its cores. The OSSA server uses a separate machine to handle all the requests coming from all UOC V2 users.

NOTE: 1 server Unified Console uses 1 core of the machine

One machine with:

- Apache CouchDB
- N instances of UOC V2.3 (e.g. N core of the mane will be used)
- Plugin(s) (e.g. OSS Analytics Plugin)

One machine with:

- Domain or Data Server(s) (ex: OSS Analytic Server, OSS Assurance Monitoring)

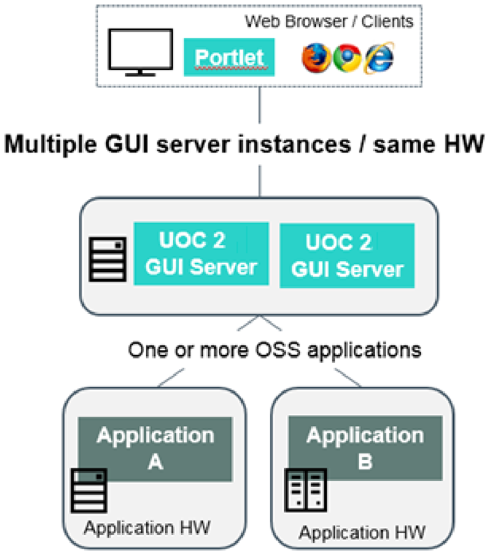


Figure 6: Deployment Examples – Multiples UOC Instances

IMPORTANT: All instances should run the same version of Unified OSS Console. The installation and configuration is done one time, and multiple server can be run on different port. Useful administration tools can be used to ease this deployment (See Chapter 15 Node Process Manager Tools)

2.4.3 Multiple servers

In this deployment, several machines are used to run several instances of the UOC V2 server to support a large volume of data and users. The OSSA server is also installed on several separate servers to handle all the requests coming from all UOC V2 users.

One machine with:

- Apache CouchDB (shared by all UOC instances)

One machine with:

- Redis (Notification Server)

Several machines with:

- N instances of UOC V2.3 (e.g. N core of the mane will be used)
- Plugin(s) (e.g. OSS Analytics Plugin)

One machine with:

- Domain or Data Server(s) (ex: OSS Analytic Server, OSS Assurance Monitoring)

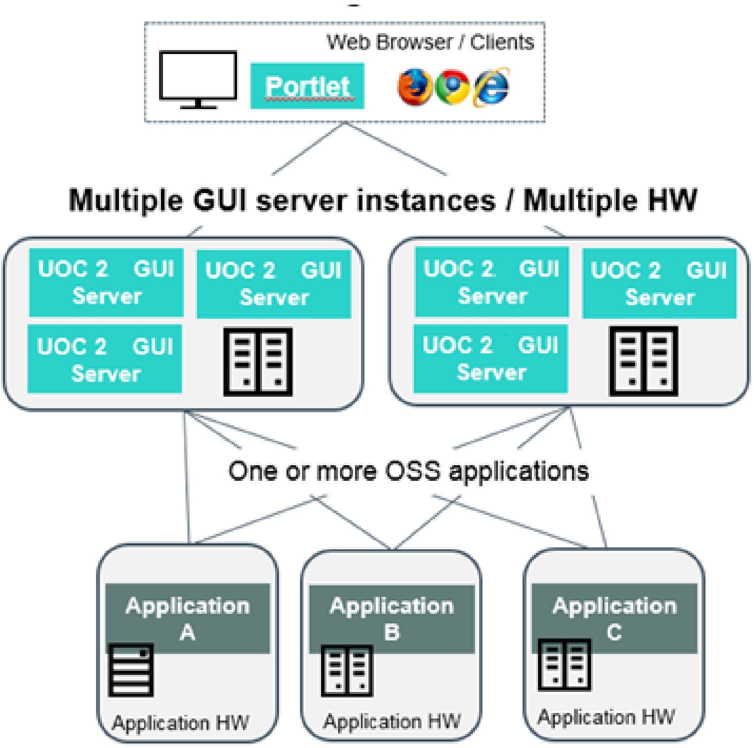


Figure 7: Deployment Examples – Multiples Servers

IMPORTANT: All instances should run the same version of Unified OSS Console. The installation and configuration is done one time, and multiple server can be run on different port. Useful administration tools can be used to ease this deployment (See Chapter 15 Node Process Manager Tools)

Chapter 3
Hardware and Software Requirements

3.1 Hardware Requirements

3.1.1 Unified Console Server

The table below lists the recommended hardware requirements for an UOC server installation. Recommended hardware is: HP ProLiant BL465c or DL360p Gen8

Appropriate sizing is of course subject to real volume of data, throughput and/or number of concurrent users. For an optimum sizing exercise, please contact the product manager.

IMPORTANT: The required Hardware will also be driven by the additional domain servers associated to the UOC Server.

Hardware	Recommended	Optimum
CPU	1x Intel® Xeon® E5-2640 2.5GHz/6-core	Needs sizing
RAM	16 GB	Needs sizing
Hard disk Size	100 GB	Needs sizing
Network	1x 10 Gbps Ethernet Ports on board/Dual Port FC HBA	Needs sizing

Figure 8: Hardware requirements for UOC V2.3 on Linux

3.1.2 Client PC / Laptop Hardware Requirements

UOC is fully compliant with mobile device and provide responsive screens.

Requirements	Minimal	Recommended
CPU	2 cores	4 cores
RAM	1 GB	2 GB
WIFI	*802.11b/g/n*	*802.11ac*
Display Size	*14”*	*24”*

Figure 9: Hardware requirements for client PC

3.1.3 Mobile Device Hardware Requirements

UOC is fully compliant with mobile device and provide responsive screens.

Requirements	Minimal	Recommended
CPU	2 cores	4 cores
RAM	1 GB	2 GB
WIFI	*802.11b/g/n*	*802.11ac*
Display Size	*Any*	*Tablet 10”*

Figure 10: Hardware requirements for mobile devices

3.2 Software Requirements

3.2.1 Supported Operating Systems

The HPE Unified Console software kit is targeted for Red Hat Enterprise Linux V6.5 and V7.2 (x86-64) only.

Operating system	Version
RedHat Enterprise Linux	6.5
RedHat Enterprise Linux	7.2

Figure 11: Supported Operating Systems

3.2.1.1 Package to install for Operating Systems

The different following packages can be installed on Operating System. If not, you can find the package by the following URL in the table.

3.2.1.1.1 Package to install for RHEL 6.5

Package	Version	URL
EPEL	6.8	http://mirror.switch.ch/ftp/mirror/epel/6/x86_64/epel-release-6-8.noarch.rpm
SpiderMonkey	1.8.5	Page search: http://rpm.pbone.net/index.php3?stat=26&dist=74&size=1087180&name=js-1.8.5-2.1.x86_64.rpm RPM Link: ftp://ftp.icm.edu.pl/vol/rzm5/linux-opensuse/repositories/home:/csbuild:/DBA/RedHat_RHEL-6/x86_64/js-1.8.5-2.1.x86_64.rpm

3.2.1.1.2 Package to install for RHEL 7.2

Package	Version	URL
EPEL	7.8	http://download.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-8.noarch.rpm
SpiderMonkey	1.8.5	Page Search: http://rpm.pbone.net/index.php3/stat/4/idpl/31979663/dir/centos_7/com/js-1.8.5-19.el7.x86_64.rpm.html RPM Link: ftp://ftp.icm.edu.pl/vol/rzm5/linux-centos/7.2.1511/os/x86_64/Packages/js-1.8.5-19.el7.x86_64.rpm

3.2.2 Supported Web Browsers

All modern Web Browser supporting HTML 5 / CSS 3 will be supported. The Unified Console have been especially validated on the following browsers:

Web Browser	Version	Web site
Microsoft Internet Explorer	11 or later	http://windows.microsoft.com/en-us/internet-explorer/download-ie
Mozilla Firefox	V32 or later	https://www.mozilla.org/en-US/firefox
Google Chrome	V37 or later	https://www.google.com/chrome
Microsoft Edge	V25 or later	https://www.microsoft.com/en-gb/windows/microsoft-edge

Figure 12: Supported Web browsers

Chapter 4 Pre-requisite Installation on RHEL 6.5

In this chapter, you will find the pre-requisites you need to install on a Red Hat Linux 6.5

You can check the current Linux release information by executing one of the following commands:

```
# lsb_release -id
```

Or

```
# cat /etc/issue
```

Or

```
# cat /etc/redhat-release
```

All the command line will require bash Unix shell. (/bin/bash)

4.1.1 Yum Proxy Configuration

If you are behind a firewall, you will need to define correctly your proxy server to run successfully yum commands.

When you want to install rpm packages using yum on Linux server, you must configure the http proxy in the yum configuration file.

As a root, edit the configuration file **/etc/yum.conf** and add:

```
proxy=http://<my_proxy_host>:<my_proxy_port>
```

Where <my_host_proxy> is the host of your proxy, and <my_port_proxy> is the port of your proxy

4.1.2 Users & Groups

You need to have the **root** credentials for installing the packages. However, installed files will be owned by standard users, and no processes will run under the root account.

For security reasons, no Linux user is created automatically during the installation phase. The required users must therefore be created manually prior to the installation.

4.1.2.1 Create groups

We recommend the creation of the following groups on the system:

- couchdb: group owning the couchdb installation and running the couchdb processes or service.
- uoc: group owning all OSS console files, and running the web server (node.js process).

You can create these groups with the following commands:

```
# groupadd uoc
# groupadd couchdb
```

4.1.2.2 Create non-login users

We recommend the creation of the following non-login users on the system:

- **couchdb**: non-login user belonging to the group **couchdb**.
- **uoc**: non-login user belonging to the group **uoc**.

These two users are the ones expected by default. You can change them during the installation (see below).

If no specific user is given as command line option, and if the default ones (i.e. couchdb and uoc) do not exist on the system, **the installation will abort**.

You can create these users with the following commands:

```
# useradd -m uoc –g uoc
# useradd -m couchdb -g couchdb
```

For security reasons, it is recommended not to fill passwords for these users to keep non-login accounts. There is no check to identify weak passwords.

4.1.3 Node.js

The UOC server relies on Node.js V4.6 or above.

Node.js must therefore be installed on the system before installing UOC.

You can install node.js through npm (requires an Internet connection) or using the standard rpm package (yum):

NOTE: Please refer to the official web site to get detailed information about installation of NodeJS
https://nodejs.org
You also can check the NodeJS repository for latest updates: https://rpm.nodesource.com/pub_4.x/el/6/x86_64/

If you have a doubt about the package to install, please contact our support team that will help you to check the right version of NodeJS.

First, you need to get the rpm package from rpm.nodesource repository

```
# wget https://rpm.nodesource.com/pub_4.x/el/6/x86_64/nodejs-4.6.0-1nodesource.el6.x86_64.rpm
```

You now have the nodejs rpm file in your current directory. Install it with the following rpm command

```
# rpm -ivh nodejs-4.6.0-1nodesource.el6.x86_64.rpm
```

Check the node version you just installed

```
# node --version
V4.6.0
```

Now you have to update npm to the latest version.

```
npm install -g npm@latest
```

You should now have a npm version > 3.x

NOTE: If you cannot download the latest npm package on Linux server by the using following command:

```
npm install -g npm@latest
```

You must set the proxy in the configuration of Linux or in the npm package. To set the proxy you can you refer to the chapter 20.1 Upgrading npm: SSL23_GET_SERVER_HELLO:tlsv1 alert access denied

IMPORTANT: NodeJS disable SSLv2 and SSLv3 by default. Note that SSLv3 is vulnerable and not to be used today (Poodle vulnerability found late 2014). So, the NodeJS v0.10.40 is a vulnerable version in term of security if you want to use SSL v2 or SSL v3. It is strongly recommended to use SSL/TLS instead and monitor regularly the update of NodeJS to migrate on a more recent version able to fix all these vulnerabilities.

You can check the vulnerabilities found at the following link: https://nodejs.org/en/blog/vulnerability/

4.1.4 Apache CouchDB

The UOC server relies on Apache CouchDB V1.6.0 or above.

The Apache CouchDB must therefore be installed on the system before installing UOC.

NOTE: Please refer to the official web site to get detailed information about installation of Apache CouchDB http://couchdb.apache.org/

If you cannot find the package or have a doubt about which package to install,
please contact our support team that will help you to check the right version of Apache CouchDB

ftp://nfsgre.gre.hpecorp.net/pub/UOC_V2/kits/UOC_PREREQUISITES/couchdb-1.6.0-1.el6.x86_64.rpm

Usually Apache CouchDB is installed on this default location:

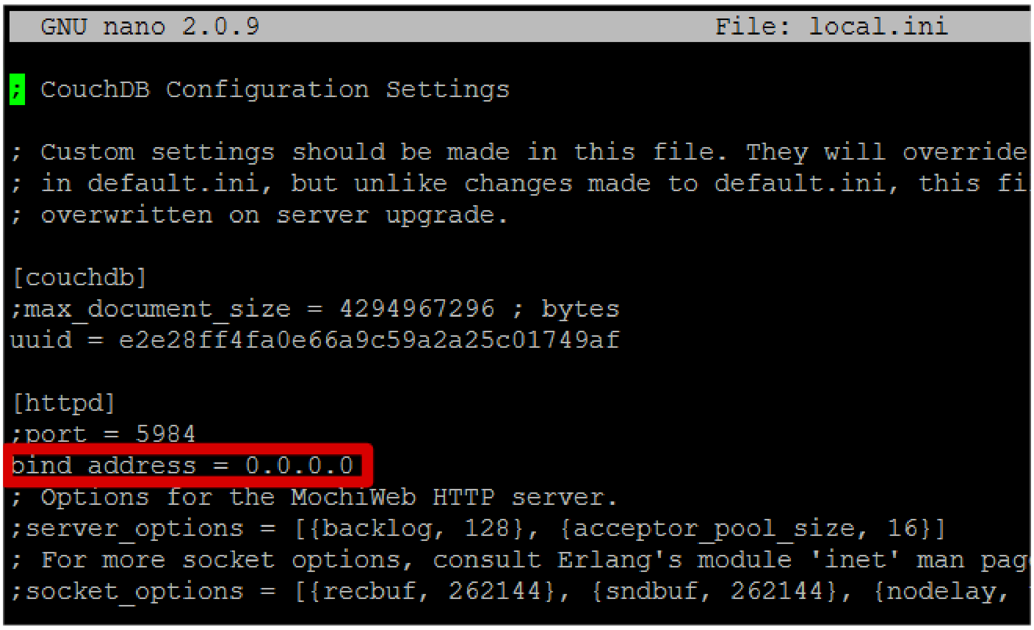
- /opt/couchdb

After installation, you need to authorize request from other server to the couchDB.

IMPORTANT: For the first installation, you must modify the file located to the path "/opt/couchdb/etc/couchdb/local.ini" which is a symbolic link to the file "/opt/couchdb/var/config/couchdb/local.ini".

1. Edit this file and search the line with the property named "bind_address".
2. Uncomment this line and change the value of the property (set to 127.0.0.1) to the value "0.0.0.0".
3. Save the file and restart the service couchdb.

This file is the configuration of the CouchDB on the server and it is kept when the CouchDB component is deleted on the file path "/opt/couchdb/var/config/couchdb/local.ini".



```
  GNU nano 2.0.9                           File: local.ini

; CouchDB Configuration Settings

; Custom settings should be made in this file. They will override
; in default.ini, but unlike changes made to default.ini, this fi
; overwritten on server upgrade.

[couchdb]
;max_document_size = 4294967296 ; bytes
uuid = e2e28ff4fa0e66a9c59a2a25c01749af

[httpd]
;port = 5984
bind_address = 0.0.0.0
; Options for the MochiWeb HTTP server.
;server_options = [{backlog, 128}, {acceptor_pool_size, 16}]
; For more socket options, consult Erlang's module 'inet' man pag
;socket_options = [{recbuf, 262144}, {sndbuf, 262144}, {nodelay,
```

IMPORTANT: Once CouchDB is installed & running, create the CouchDB admin user. To create the user, see the chapter 9.2.2.

After CouchDB installation, you need to open used ports in your firewall if you want to access from an external host or if you want to launch futon from your web browser from another machine. (default port HHTP is 5984, default port for HTTPS is 6984)

Check port open in RHEL 6.x with the following command:

```
sudo iptables -L -n
```

Open port for 5984 (http) and 6984 (https).

```
sudo iptables -I INPUT -p tcp --dport 5984 -j ACCEPT
sudo iptables -I INPUT -p tcp --dport 6984 -j ACCEPT
```

Verify with the following command:

```
sudo iptables -L -n
```

Save the settings so that this change:

```
sudo /sbin/service iptables save
```

You can validate the installation with the following url http://localhost:5984 that will return a JSON answer. CouchDB includes a web-based front end called Futon which can be accessed by your browser via the following address: http://localhost:5984/_utils/

4.1.5 Redis

NOTE: Please refer to the official web site to get detailed information about installation of Redis http://redis.io/

If you have a doubt about the package to install, please contact our support team that will help you to check the right version of Redis.

UOC has been tested with the following RPM: redis-3.2.4-2.el6.remi.x86_64.rpm

As no rpm exists for the version 3.2.4, we recommend installing Redis from source

4.1.5.1 Install from source

Usually Redis is installed on this location:

- /opt/redis

If your server is connected to an enterprise network, you usually need to use a proxy to connect to the internet. Prerequisite: have wget installed and http proxy configured.

To configure wget, simply create a file .wgetrc inside your home directory, with this content.

Replace <my_host_proxy> to the host of your proxy, and <my_port_proxy> with the port of your proxy

```
http_proxy = http://<my_host_proxy>:<my_port_proxy>/
use_proxy = on
wait = 15
```

Get the sources and compile them:

```
# wget http://download.redis.io/releases/redis-3.2.4.tar.gz
# tar xzvf redis-3.2.4.tar.gz
# cd redis-3.2.4
# make
# make install
# make test
```

No need to use root user so far. If you need to start Redis server as a user different from root, use the following commands (not recommended for a production system)

```
# cd src
# ./redis-server
```

Else, use root user and execute the following commands:

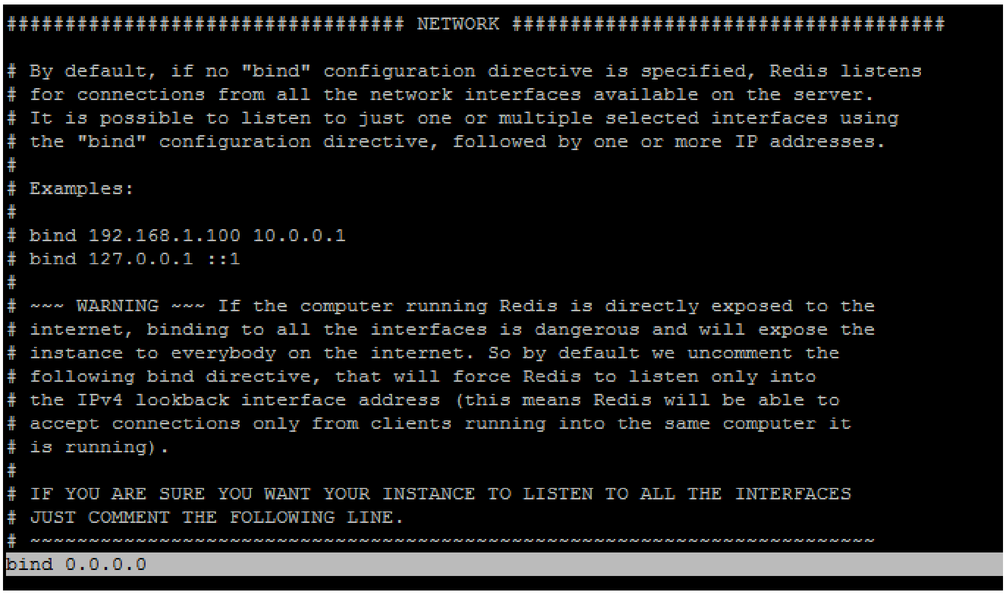
```
# cd utils
# ./install_server.sh
```

Answer questions to configure Redis server.

If you want to access redis from an external machine (i.e. UOC deployed on machine A and machine B, and Redis on machine) you have to configure Redis to be accessible from external machines. After installation, modify the configuration file:

```
nano /etc/redis/6379.conf
```

(Note: if redis is deployed on a different port, the file is <redis_port>.conf)



```
################################## NETWORK #####################################

# By default, if no "bind" configuration directive is specified, Redis listens
# for connections from all the network interfaces available on the server.
# It is possible to listen to just one or multiple selected interfaces using
# the "bind" configuration directive, followed by one or more IP addresses.
#
# Examples:
#
# bind 192.168.1.100 10.0.0.1
# bind 127.0.0.1 ::1
#
# ~~~ WARNING ~~~ If the computer running Redis is directly exposed to the
# internet, binding to all the interfaces is dangerous and will expose the
# instance to everybody on the internet. So by default we uncomment the
# following bind directive, that will force Redis to listen only into
# the IPv4 lookback interface address (this means Redis will be able to
# accept connections only from clients running into the same computer it
# is running).
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
bind 0.0.0.0
```

At the beginning of the file, under NETWORK paragraph, change the line **'bind 127.0.0.1'** to '**bind 0.0.0.0**' (see picture above). Save the file (ctrl+o) and exit document (ctrl+x). Redis is now open to any connection (Consider the warning in the comment). Nevertheless, you can bind your own addresses (see examples in the comment)

If you are behind a firewall, you might need to open your port. Open the port where you deployed redis (default: 6379)

```
 sudo iptables -I INPUT -p tcp --dport 6379 -j ACCEPT
```

Verify with the following command:

```
sudo iptables -L -n
```

Save the settings so that this change:

```
sudo /sbin/service iptables save
```

Then start redis using the command:

```
# service redis_<portnumber> start
```

For instance:

```
# service redis_6379 start
```

Chapter 5 Pre-requisite Installation on RHEL 7.2

In this chapter, you will find the pre-requisites you need to install on a Red Hat Linux 7.2

You can check the current Linux release information by executing one of the following commands:

```
# lsb_release –id
```

Or

```
# cat /etc/issue
```

Or

```
# cat /etc/redhat-release
```

All the command line will require bash Unix shell. (/bin/bash)

5.1.1 Yum Proxy Configuration

If you are behind a firewall, you will need to define correctly your proxy server to run successfully yum commands.

When you want to install rpm packages using yum on Linux server, you must configure the http proxy in the yum configuration file.

As a root, edit the configuration file **/etc/yum.conf** and add:

```
proxy=http://<my_proxy_host>:<my_proxy_port>
```

Where <my_host_proxy> is the host of your proxy, and <my_port_proxy> is the port of your proxy

5.1.2 Users & Groups

You need to have the **root** credentials for installing the packages. However, installed files will be owned by standard users, and no processes will run under the root account.

For security reasons, no Linux user is created automatically during the installation phase. The required users must therefore be created manually prior to the installation.

5.1.2.1 Create groups

We recommend the creation of the following groups on the system:

- couchdb: group owning the couchdb installation and running the couchdb processes or service.
- uoc: group owning all OSS console files, and running the web server (node.js process).

You can create these groups with the following commands:

```
# groupadd uoc
# groupadd couchdb
```

5.1.2.2 Create non-login users

We recommend the creation of the following non-login users on the system:

- couchdb: non-login user belonging to the group couchdb.
- uoc: non-login user belonging to the group uoc.

These two users are the ones expected by default. You can change them during the installation (see below).

If no specific user is given as command line option, and if the default ones (i.e. couchdb and uoc) do not exist on the system, the installation will abort.

You can create these users with the following commands:

```
# useradd -m uoc –g uoc
# useradd -m couchdb -g couchdb
```

For security reasons, it is recommended not to fill passwords for these users to keep non-login accounts. There is no check to identify weak passwords.

5.1.3 Node.js

The UOC server relies on Node.js V4.6 or above.

Node.js must therefore be installed on the system before installing UOC.

You can install node.js through npm (requires an Internet connection) or using the standard rpm package (yum):

NOTE: Please refer to the official web site to get detailed information about installation of NodeJS https://nodejs.org
You also can check the NodeJS repository for latest updates: https://rpm.nodesource.com/pub_4.x/el/7/x86_64/

If you have a doubt about the package to install, please contact our support team that will help you to check the right version of NodeJS.

First, you need to get the rpm package from rpm.nodesource repository and install it with rpm command

Replace <my_host_proxy> to the host of your proxy, and <my_port_proxy> with the port of your proxy

```
# rpm –ivh https://rpm.nodesource.com/pub_4.x/el/7/x86_64/nodejs-4.6.0-1nodesource.el7.centos.x86_64.rpm --httpproxy
http://<my_host_proxy>:<my_port_proxy>
```

Or

```
wget https://rpm.nodesource.com/pub_4.x/el/7/x86_64/nodejs-4.6.0-
1nodesource.el7.centos.x86_64.rpm
rpm -ivh nodejs-4.6.0-1nodesource.el7.centos.x86_64.rpm
```

NOTE: If you cannot download the rpm package on Linux server by the using “wget” command, you must download the package on your computer or set an external proxy in your server. To do so, 2 solutions :
1 – Create a .wgetrc file (configuration file for wget) in /root

```
touch .wgetrc
nano .wgetrc
```

and add the following configuration

```
http_proxy = <your http proxy:port>
https_proxy = <your https proxy:port>
use_proxy = on
wait = 15
```

OR

2 – export https_proxy variable

```
export https_proxy=<your https proxy:port>
```

Don’t forget to unset this variable after you used wget

```
unset https_proxy
```

Then, use the yum installer to install nodejs

```
# yum install nodejs
Loaded plugins: langpacks, product-id, subscription-manager
# node --version
V4.6.0
```

Now you have to update npm to the latest version

```
npm install -g npm@latest
```

You should now have a npm version > 3.x

NOTE: If you cannot download the latest npm package on Linux server by the using following command:

```
npm install -g npm@latest
```

You must set the proxy in the configuration of Linux or in the npm package. To set the proxy you can you refer to the chapter 20.1 Upgrading npm: SSL23_GET_SERVER_HELLO:tlsv1 alert access denied

IMPORTANT: NodeJS disable SSLv2 and SSLv3 by default. Note that SSLv3 is vulnerable and not to be used today (Poodle vulnerability found late 2014). So, the NodeJS v0.10.40 is a vulnerable version in term of security if you want to use SSL v2 or SSL v3. It is strongly recommended to use SSL/TLS instead and monitor regularly the update of NodeJS to migrate on a more recent version able to fix all these vulnerabilities.

You can check the vulnerabilities found at the following link: https://nodejs.org/en/blog/vulnerability/

5.1.4 Apache CouchDB

The UOC server relies on Apache CouchDB V1.6.1 or above.

The Apache CouchDB must therefore be installed on the system before installing UOC.

NOTE: Be sure you installed the prerequisites mentioned in the chapter 3.2.1.1.2. The prerequisites are used by the CouchDB package (erlang & SipderMonkey packages)

NOTE: If you cannot find the package or have a doubt about which package to install, please contact our support team that will help you to check the right version of Apache CouchDB

Usually Apache CouchDB is installed on this default location:

- /opt/couchdb

NOTE: Please refer to the official web site to get detailed information about installation of Apache CouchDB http://couchdb.apache.org/

UOC has been tested with the following RPM: **couchdb-1.6.1-4.el7.centos.x86_64.rpm**

First, you need to install CouchDB dependencies:

CouchdDB RPM dependencies	Versions
Erlang-crypto	R16B
Erlang-ibrowse	R16B
Erlang-snappy	R16B
Erlang-oauth	R16B
Erlang-sd_notify	R16B
Erlang-mochiweb	2.4.2

Replace <my_host_proxy> to the host of your proxy, and <my_port_proxy> with the port of your proxy

In case you do not have access to Erlang RPM, then install Erlang repository (http://packages.erlang-solutions.com/rpm)

```
# rpm -ivh https://packages.erlang-solutions.com/erlang-solutions-1.0-1.noarch.rpm  --httpproxy http://<my_host_proxy<:<my_port_proxy>
```

Install Erlang dependencies

```
# yum install erlang-snappy erlang-ibrowse erlang-oauth erlang-sd_notify erlang-xmerl erlang-os_mon erlang-tools
```

Install Erlang Mochiweb dependency

```
# rpm -ivh http://dev.racf.bnl.gov/yum/snapshots/rhel7/epel7-
x86_64/e/erlang-mochiweb-2.4.2-2.el7.x86_64.rpm --httpproxy
http://<my_host_proxy>:<my_port_proxy>

# yum install erlang-mochiweb
```

NOTE: Erlang distribution can be easily downloaded from Erlang Solutions at https://packages.erlang-solutions.com/erlang/ or http://www.erlang.org/

Then, you can install CouchDB

```
# rpm -ivh https://copr-
be.cloud.fedoraproject.org/results/gorbyo/couchdb/epel-7-
x86_64/00399645-couchdb/couchdb-1.6.1-4.el7.centos.x86_64.rpm  --
httpproxy http://<my_host_proxy>:<my_port_proxy>

# yum install couchdb
```

NOTE: You can find the Couch 1.6.1 RPM to another mirrors:
http://dev.racf.bnl.gov/yum/snapshots/rhel7/epel7-x86_64/c/couchdb-1.6.1-4.el7.x86_64.rpm
https://kojipkgs.fedoraproject.org//packages/couchdb/1.6.1/1.el7/x86_64/couchdb-1.6.1-1.el7.x86_64.rpm
ftp://nfsgre.gre.hpecorp.net/pub/UOC_V2/kits/UOC_PREREQUISITES/couchdb-1.6.1-4.el7.centos.x86_64.rpm

NOTE: If you encounter the following error with dependency:

```
# rpm -ivh http://dev.racf.bnl.gov/yum/snapshots/rhel7/epel7-
x86_64/e/erlang-mochiweb-2.4.2-2.el7.x86_64.rpm --httpproxy
http://<my_host_proxy>:<my_port_proxy>

Retrieving https://copr-
be.cloud.fedoraproject.org/results/gorbyo/couchdb/epel-7-
x86_64/00399645-couchdb/couchdb-1.6.1-4.el7.centos.x86_64.rpm

warning: /var/tmp/rpm-tmp.R74jxu: Header V3 RSA/SHA1 Signature, key
ID 1ff20890: NOKEY

error: Failed dependencies:

        libicudata.so.50()(64bit) is needed by couchdb-1.6.1-
4.el7.centos.x86_64

        libicui18n.so.50()(64bit) is needed by couchdb-1.6.1-
4.el7.centos.x86_64

        libicuuc.so.50()(64bit) is needed by couchdb-1.6.1-
4.el7.centos.x86_64
```

You must to install “libicu” dependency

```
# yum install libicu
```

Once CouchDB is installed, you have to configure it.

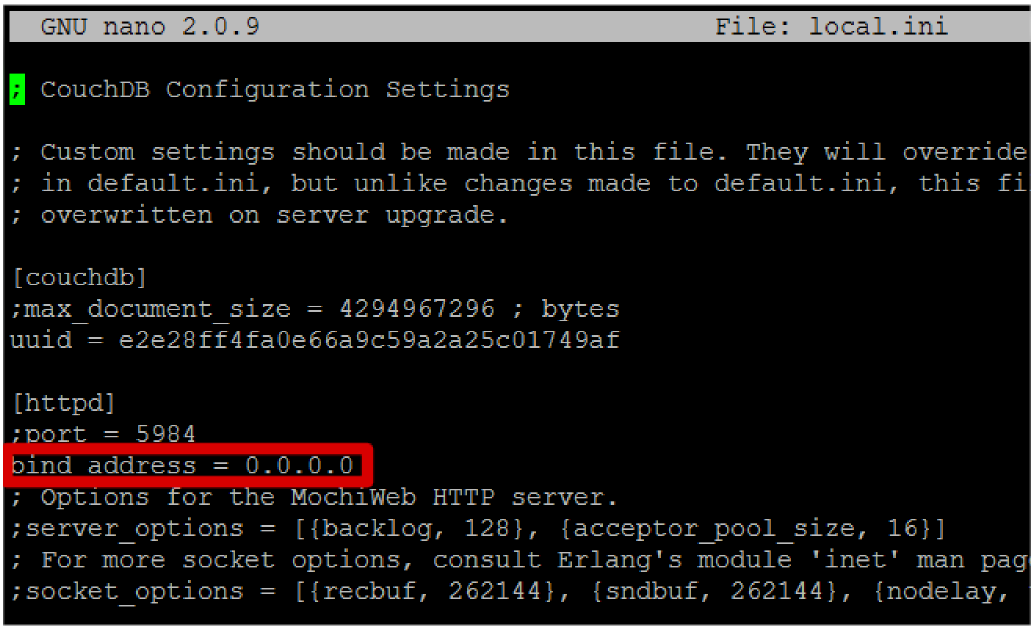
```
# cd /etc/couchdb   (or the couchdb install directory)
```

After installation, you need to authorize request from other server to the CouchDB.

IMPORTANT: For the first installation, you must modify the file located to the path “/etc/couchdb/local.ini”.

1. Edit this file and search the line with the property named “bind_address”.
2. Uncomment this line and change the value of the property (set to 127.0.0.1) to the value “0.0.0.0”.
3. Save the file and restart the service couchdb.

This file is the configuration of the CouchDB on the server and it is kept when the CouchDB component is deleted on the file path “/etc/couchdb/local.ini”.



```
  GNU nano 2.0.9                           File: local.ini

; CouchDB Configuration Settings

; Custom settings should be made in this file. They will override
; in default.ini, but unlike changes made to default.ini, this fi
; overwritten on server upgrade.

[couchdb]
;max_document_size = 4294967296 ; bytes
uuid = e2e28ff4fa0e66a9c59a2a25c01749af

[httpd]
;port = 5984
bind_address = 0.0.0.0
; Options for the MochiWeb HTTP server.
;server_options = [{backlog, 128}, {acceptor_pool_size, 16}]
; For more socket options, consult Erlang's module 'inet' man pag
;socket_options = [{recbuf, 262144}, {sndbuf, 262144}, {nodelay,
```

IMPORTANT: Once CouchDB is installed, be sure you start it with the service command the first time :

```
# service couchdb start
```

If you have issues with starting CouchDB service, please refer to Chapter 20. Mistakes and solutions

NOTE: If you want to start CouchDB on the boot of Linux server, you must enable the systemctl service of CouchDB by the following command:

```
# systemctl enable couchdb.service
```

IMPORTANT: Once CouchDB is installed & running, create the CouchDB admin user. To create the user, see the chapter 9.2.2.

After CouchDB installation, you need to open used ports in your firewall if you want to access from an external host or if you want to launch futon from your web browser from another machine. (default port HTTP is 5984, default port for HTTPS is 6984)

Check port open in RHEL 7.x with the following command:

```
firewall-cmd --list-ports
```

Open port for 5984 (http) and 6984 (https).

```
sudo firewall-cmd --permanent --zone=public --add-port=5984/tcp
sudo firewall-cmd --permanent --zone=public --add-port=6984/tcp
sudo firewall-cmd --reload
```

You can validate the installation with the following url http://localhost:5984 that will return a JSON answer.

CouchDB includes a web-based front end called Futon which can be accessed by your browser via the following address: http://localhost:5984/_utils/

5.1.5 Redis

There are 3 ways of installing Redis:

- Using Rpm
- Using Yum
- Using source

NOTE: Please refer to the official web site to get detailed information about installation of Redis http://redis.io/

If you have a doubt about the package to install, please contact our support team that will help you to check the right version of Redis.

UOC has been tested with the following RPM: redis-3.2.4-1.el7.remi.x86_64.rpm

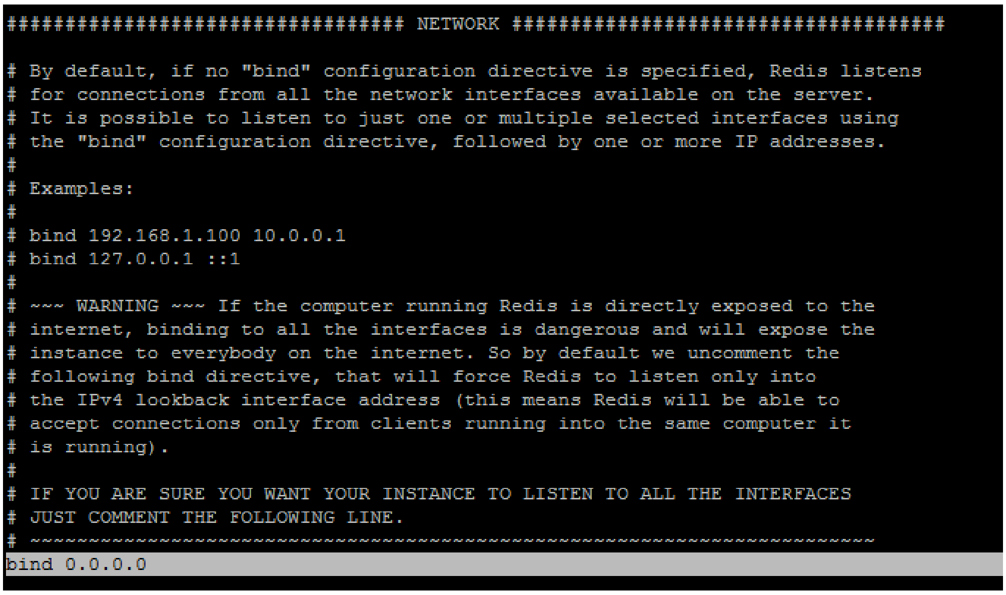
5.1.5.1 Install using rpm

You must be root user to execute one of the following set of commands:

```
# rpm –ivh jemalloc-3.6.0-1.el7.x86_64.rpm
# rpm –ivh redis-3.2.4-1.el7.remi.x86_64.rpm
```

If you want to access redis from an external machine (i.e. UOC deployed on machine A and machine B, and Redis on machine C) you have to configure Redis to be accessible from external machines. After installation, modify the configuration file:

```
nano /etc/redis.conf
```



```
################################## NETWORK #####################################

# By default, if no "bind" configuration directive is specified, Redis listens
# for connections from all the network interfaces available on the server.
# It is possible to listen to just one or multiple selected interfaces using
# the "bind" configuration directive, followed by one or more IP addresses.
#
# Examples:
#
# bind 192.168.1.100 10.0.0.1
# bind 127.0.0.1 ::1
#
# ~~~ WARNING ~~~ If the computer running Redis is directly exposed to the
# internet, binding to all the interfaces is dangerous and will expose the
# instance to everybody on the internet. So by default we uncomment the
# following bind directive, that will force Redis to listen only into
# the IPv4 lookback interface address (this means Redis will be able to
# accept connections only from clients running into the same computer it
# is running).
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
bind 0.0.0.0
```

At the beginning of the file, under NETWORK paragraph, change the line **'bind 127.0.0.1'** to '**bind 0.0.0.0**' (see picture above). Save the file (ctrl+o) and exit document (ctrl+x). Redis is now open to any connection (Consider the warning in the comment). Nevertheless, you can bind your own addresses (see examples in the comment)

If you are behind a firewall, you might need to open your port. Open the port where you deployed redis (default: 6379)

```
 sudo firewall-cmd --permanent --zone=public --add-port=6379/tcp
```

Verify with the following command:

```
sudo iptables -L -n
```

Save the settings so that this change:

```
sudo firewall-cmd --reload
```

Then start, stop, restart, and get status of Redis using service command

```
# service redis start
```

5.1.5.2 Install using yum

If your server is connected to an enterprise network, you usually need to use a proxy to connect to the internet.

Replace <my_host_proxy> and <my_port_proxy> by your proxy host and port. You will need to install yum repo first before installing redis.

Log as **root**.

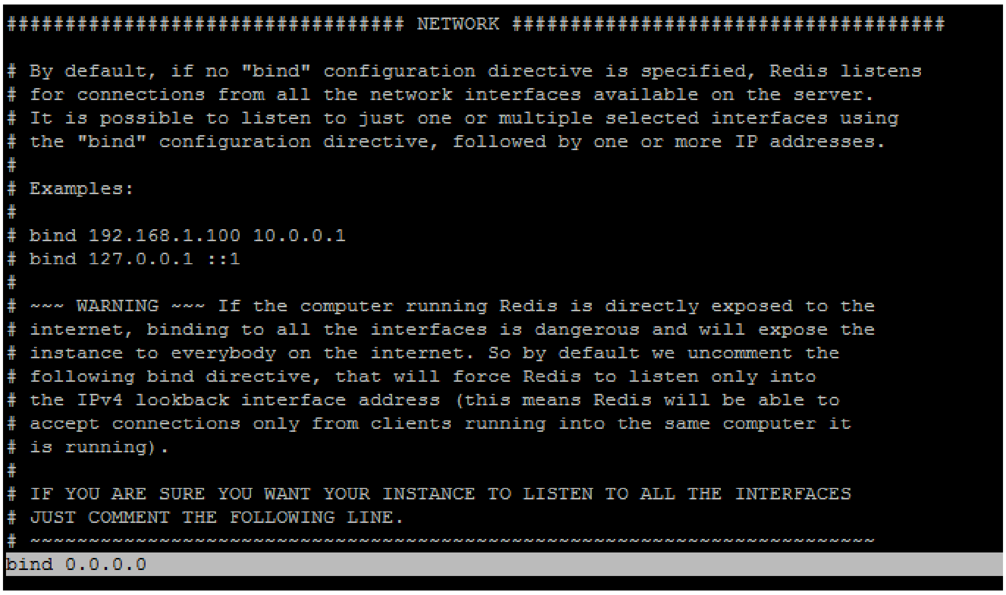
```
# rpm --httpproxy http://host_proxy:port_proxy -Uvh
http://download.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-
8.noarch.rpm


# rpm --httpproxy http://<my_host_proxy>:<my_port_proxy> -Uvh
http://rpms.famillecollet.com/enterprise/remi-release-7.rpm


# yum --enablerepo=remi,remi-test install redis-3.2.4-1.el7.remi.x86_64
```

If you want to access redis from an external machine (i.e. UOC deployed on machine A and machine B, and Redis on machine C) you have to configure Redis to be accessible from external machines. After installation, modify the configuration file:

```
nano /etc/redis.conf
```



```
################################## NETWORK #####################################

# By default, if no "bind" configuration directive is specified, Redis listens
# for connections from all the network interfaces available on the server.
# It is possible to listen to just one or multiple selected interfaces using
# the "bind" configuration directive, followed by one or more IP addresses.
#
# Examples:
#
# bind 192.168.1.100 10.0.0.1
# bind 127.0.0.1 ::1
#
# ~~~ WARNING ~~~ If the computer running Redis is directly exposed to the
# internet, binding to all the interfaces is dangerous and will expose the
# instance to everybody on the internet. So by default we uncomment the
# following bind directive, that will force Redis to listen only into
# the IPv4 lookback interface address (this means Redis will be able to
# accept connections only from clients running into the same computer it
# is running).
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
bind 0.0.0.0
```

At the beginning of the file, under NETWORK paragraph, change the line **'bind 127.0.0.1'** to '**bind 0.0.0.0**' (see picture above). Save the file (ctrl+o) and exit document (ctrl+x). Redis is now open to any connection (Consider the warning in the comment). Nevertheless, you can bind your own addresses (see examples in the comment)

If you are behind a firewall, you might need to open your port. Open the port where you deployed redis (default: 6379)

```
sudo firewall-cmd --permanent --zone=public --add-port=6379/tcp
```

Verify with the following command:

```
sudo iptables -L -n
```

Save the settings so that this change:

```
sudo firewall-cmd --reload
```

Then start, stop, restart, and get status of Redis using service command

```
# service redis start
```

5.1.5.3 Install from source

Usually Redis is installed on this location: **/opt/redis**

If your server is connected to an enterprise network, you usually need to use a proxy to connect to the internet. Prerequisite: have wget installed and http proxy configured.

To configure wget, simply create a file **.wgetrc** inside your home directory, with this content. Replace <my_host_proxy> to the host of your proxy, and <my_port_proxy> with the port of your proxy

```
http_proxy = http://<my_host_proxy>:<my_port_proxy>/
use_proxy = on
wait = 15
```

Get the sources and compile them:

```
# wget http://download.redis.io/releases/redis-3.2.4.tar.gz
# tar xzvf redis-3.2.4.tar.gz
# cd redis-3.2.4
# make
# make install
# make test
```

No need to use root user so far. If you need to start Redis server as a user different from root, use the following commands (**not recommended for a production system**)

```
# cd src
# ./redis-server
```

Else, use root user and execute the following commands:

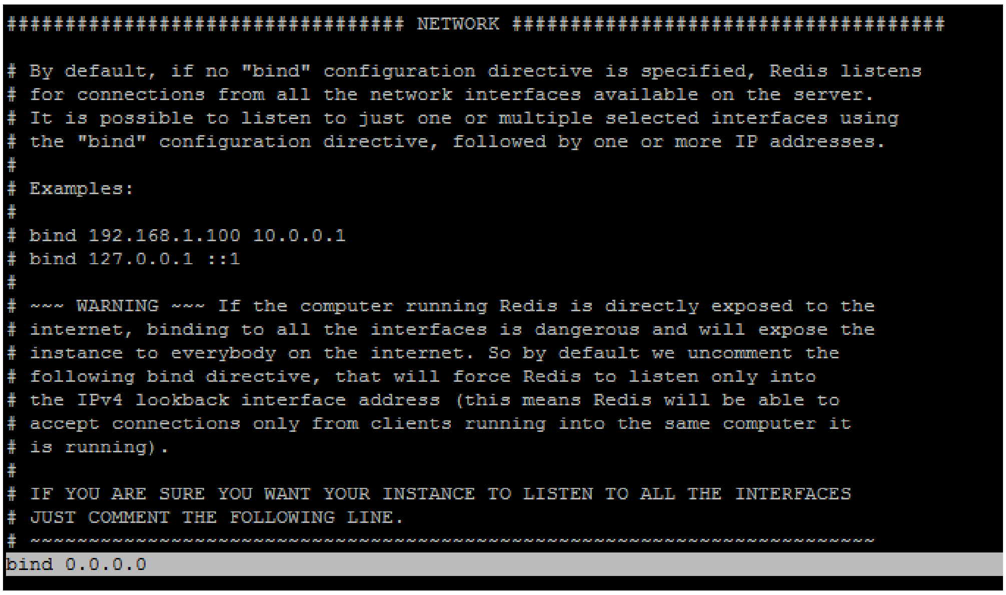
```
# cd utils
# ./install_server.sh
```

Answer questions to configure Redis server.

If you want to access redis from an external machine (i.e. UOC deployed on machine A and machine B, and Redis on machine) you have to configure Redis to be accessible from external machines. After installation, modify the configuration file:

```
nano /etc/redis/6379.conf
```

(Note: if redis is deployed on a different port, the file is <redis_port>.conf)



```
################################## NETWORK #####################################

# By default, if no "bind" configuration directive is specified, Redis listens
# for connections from all the network interfaces available on the server.
# It is possible to listen to just one or multiple selected interfaces using
# the "bind" configuration directive, followed by one or more IP addresses.
#
# Examples:
#
# bind 192.168.1.100 10.0.0.1
# bind 127.0.0.1 ::1
#
# ~~~ WARNING ~~~ If the computer running Redis is directly exposed to the
# internet, binding to all the interfaces is dangerous and will expose the
# instance to everybody on the internet. So by default we uncomment the
# following bind directive, that will force Redis to listen only into
# the IPv4 lookback interface address (this means Redis will be able to
# accept connections only from clients running into the same computer it
# is running).
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
bind 0.0.0.0
```

At the beginning of the file, under NETWORK paragraph, change the line **'bind 127.0.0.1'** to '**bind 0.0.0.0**' (see picture above). Save the file (ctrl+o) and exit document (ctrl+x). Redis is now open to any connection (Consider the warning in the comment). Nevertheless, you can bind your own addresses (see examples in the comment)

If you are behind a firewall, you might need to open your port. Open the port where you deployed redis (default: 6379)

```
 sudo firewall-cmd --permanent --zone=public --add-port=6379/tcp
```

Verify with the following command:

```
sudo iptables -L -n
```

Save the settings so that this change:

```
sudo firewall-cmd --reload
```

Then start it using the command:

```
# service redis_<portnumber> start
```

For instance:

```
# service redis_6379 start
```

Chapter 6 Installation

6.1 Installation locations

UOC RPM packages have default installation locations:

- /opt/uoc2
- /var/opt/uoc2

6.2 Installation media

The Unified OSS Console V2.3 comes in a standard tar file:

UOCV2.3.0-MR.tar

Unpack the archive in a temporary directory of your choice:

```
$ tar xvf UOCV2.3.0-MR.tar
uoc2_kit/
uoc2_kit/install.sh
uoc2_kit/uoc-2.3.0-MR.x86_64.rpm
uoc2_kit/README
```

6.3 Interactive script

The Unified Console is installed using an interactive shell script that will prompt for important options, like which packages to install, target locations on disk and users. For all options, a default value is proposed. If you want a standard installation, fully monolithic, with all default values, you can use the -s option (for silent or scratch).

Interactive script:

```
# sudo ./install.sh
```

Or, install everything on the same system, using default values, no questions asked.

```
# sudo ./install.sh -s
```

IMPORTANT:
The installation script is able to detect a new installation for V2.3 or an upgrade to a new maintenance Pack (MP).

```
# ./install.sh
uoc-2.2.x-MP.x86_64 is already installed on this system.
If you continue your installation, uoc will be automatically upgraded.
Install uoc (y/n)? y
```

6.4 User profile

UOC ships an environment file, to be sourced by the uoc user. This will set in particular the UOC2_HOME and UOC2_DATA environment variables to the correct values and update the PATH to locate the uoc2 command.

```
# cat /var/opt/uoc2/.environment.sh >> /home/uoc/.bash_profile
# su - uoc
$ source /home/uoc/.bash_profile
$ which uoc2
/opt/uoc2/bin/uoc2
```

Source operation will override your PATH.

6.5 Setup

At this stage, there are 2 options:

- Perform a new installation or upgrade for V2.3 (new CouchDB initialization, preload default settings…)
- Migrate an existing installation from V2.2 (existing CouchDB to migrate with new schema and keep existing data)

IMPORTANT: To migrate from a UOC V2.1, you will need to use the UOC V2.2 migration script first, then apply the UOC V2.3 migration script. There is not direct migration from UOC V2.1 to V2.3

6.5.1 New installation

After a first installation, you need to create Apache CouchDB databases and initialize the mandatory data. You can do this automatically:

Log first using the **root** user, and execute the following script.

```
$ unset http_proxy
$ /opt/uoc2/scripts/setup.sh
```

You can execute this script as a normal user. However, if you wish to start the local Apache CouchDB server on a newly installed local system, CouchDB credentials will be required.

Regarding the UOC V2 server, you can choose a few important parameters, like the Apache CouchDB server hostname and port.

For this, edit the following file (JSON syntax):

```
$ vi /var/opt/uoc2/server/public/conf/config.json
```

If you use a basic local Apache CouchDB installation, you can keep the default values.

If you have installed a notification server in your solution (Redis), please update manually the configuration to setup the server host and port to access.

6.5.2 Migrate an existing UOC V2.2 to UOC V2.3 installation

You already have an Apache CouchDB installation with objects and customization. You need to run a specific script to migrate all the objects with new changes. The script will update all your objects with the new up to date data format.

IMPORTANT: Migration will migrate existing workspaces. All other data does not require any changes.

Log first using the **uoc** user, and execute the following script.

IMPORTANT:
Migration have to be run with uoc right because it requires to access to the local data file and Apache CouchDB. If you use couchdb user, you will have an access right denied due to the Unix rights violation.

```
$ unset http_proxy
$ /opt/uoc2/scripts/migrate_v2.2_to_v2.3.sh
```

You can execute this script as a normal user. However, if you wish to start the local CouchDB server on an installed local system, CouchDB credentials will be required.

You still can made some changes on your existing parameters editing the configuration file like a new installation.

```
$ vi /var/opt/uoc2/server/public/conf/config.json
```

If you use a basic local CouchDB installation, you can keep the default values.

Note that all the default files (configuration, images, icons…) installed with a new kit are still available as backup in

- Public Data are available in /opt/uoc2/data.kitting
- Server Public Data files are available in /opt/uoc2/server/public.kitting
- Client Public Data files are available in /opt/uoc2/client/public.kitting

In case of changes on default files, it may require a manual check and merge to get the latest changes.

Example: To get the new updated icons for widgets, you will need to manually copy the contents of /opt/uoc2/client/public.kitting directory to /var/opt/uoc2/client/public

6.5.3 Use UOC as a Linux service

The UOC server has the ability to be used as a Linux service for RHEL 6.5 or 7.2

6.5.3.1 UOC as a Linux service on RedHat 6.5

The UOC package provides files which can create the Linux service on RHEL 6.5

The file will be used by the component named “chkconfig”.

For more explanation, you can see the following URL:

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/s2-services-chkconfig.html

http://linux.die.net/man/8/chkconfig

6.5.3.1.1 Deploy service

To deploy the UOC as a linux service in “chkconfig”, the UOC package provides a file: “<UOC_HOME>/bin/uoc2.service.6.5”.

This file is a script and has some description explaining its functionality and its initialization in “chkconfig” during its adding.

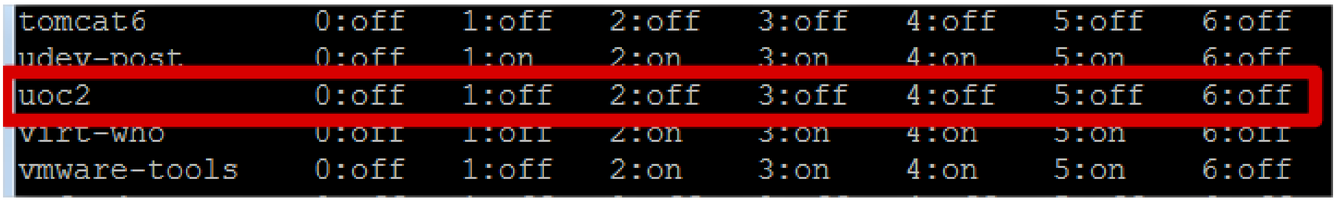
To add this file, you must execute these following commands:

```
$ # copy the file in the “chkconfig” directory
$ cp <UOC_HOME>/bin/uoc2.service.6.5 /etc/init.d/uoc2
$ # add the permissions to execute the script file by chkconfig
$ chmod 755 /etc/init.d/uoc2
$ # add the script in the chkconfig
$ chkconfig --add uoc2
```

The service is available in the “chkconfig” and can be used by the users.

The service is installed and not activated for the server starting:

```
$ # see the configuration of the UOC service in chkconfig
$ chkconfig --list
```



```
tomcat6         0:off   1:off   2:off   3:off   4:off   5:off   6:off
udev-post       0:off   1:on    2:on    3:on    4:on    5:on    6:off
uoc2            0:off   1:off   2:off   3:off   4:off   5:off   6:off
virt-who        0:off   1:off   2:on    3:on    4:on    5:on    6:off
vmware-tools    0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

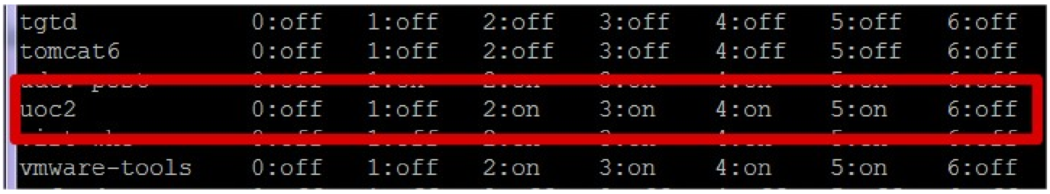
6.5.3.1.2 Auto start UOC service at Linux startup

The UOC service can used all capabilities of “chkconfig” and can be used and launched during the server starting.

To use the service during the server starting, you must execute the following commands:

```
$ # use the UOC service during the server starting
$ chkconfig uoc2 --level 2345 on
```

The service is now available for the startup of the Linux server. The following picture show the configuration of the UOC service in chkconfig:



```
tgtd            0:off   1:off   2:off   3:off   4:off   5:off   6:off
tomcat6         0:off   1:off   2:off   3:off   4:off   5:off   6:off
uoc2            0:off   1:off   2:on    3:on    4:on    5:on    6:off
vmware-tools    0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

6.5.3.1.3 Execute the service

To execute the UOC service you can execute this command:

```
$ service uoc2 [start | stop | status | inventory | usage] [-p
<port_number>]
```

6.5.3.1.4 Delete the UOC service

To delete the uoc as a Linux service, you first have to stop all UOC instance. You can execute this command:

```
$ service uoc2 stop
```

Then, execute these commands:

```
$ # disable all boot profiles
$ chkconfig uoc –level 0123456 off
$ # delete the service in chkconfig
$ chkconfig --del uoc2
$ # delete the file in the /etc/init.d
$ rm /etc/init.d/uoc2
```

6.5.3.2 UOC as a Linux service on RedHat 7.2

The UOC package provides files which can create the Linux service on Redhat 7.2. The file will be used by the component named “systemd” or “systemctl”. For more explanation, you can see the following URL:

https://www.freedesktop.org/software/systemd/man/systemctl.html
https://www.freedesktop.org/software/systemd/man/systemd.html
https://www.freedesktop.org/software/systemd/man/systemd.service.html
https://www.freedesktop.org/software/systemd/man/systemd.unit.html

6.5.3.2.1 Deploy service

To deploy the UOC service in “systemd”, the UOC package provides a file: “<UOC_HOME>/bin/uoc2.service.7.2”.

This file is a description file and has some description explaining its functionality and its initialization in “systemd” during its adding.

To add this file, you must execute these following commands:

```
$ # copy the file in the “systemd” directory
$ cp <UOC_HOME>/bin/uoc2.service.7.2 /etc/systemd/system/uoc2@.service
$ # add the permissions to execute the script file by systemd
$ chmod 444 /etc/systemd/system/uoc2@.service
$ # reload systemd and services
$ systemctl daemon-reload
```

The service is available in the “systemd” and can be used by the users.

6.5.3.2.2 Auto start UOC service at Linux startup

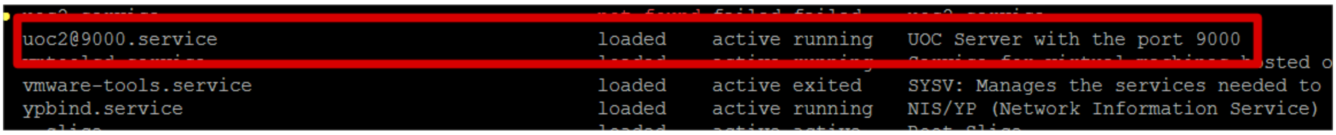
The UOC service can used all capabilities of “systemd” and can be used and launched during the server starting.

To use the service during the server starting, you must execute the following commands:

```
$ # use the UOC service during the server starting on the port 9000
$ systemctl enable uoc2@9000.service
Created symlink from /etc/systemd/system/multi-
user.target.wants/uoc2@9000.service to
/etc/systemd/system/uoc2@.service.
```

The service is now available for the startup of the Linux server.

The following picture show the configuration of the UOC service in systemd:



```
uoc2@9000.service                 loaded    active running   UOC Server with the port 9000
vmware-tools.service              loaded    active exited    SYSV: Manages the services needed to
ypbind.service                    loaded    active running   NIS/YP (Network Information Service)
```

6.5.3.2.3 Execute the service

To execute the UOC service you can execute this command:

```
$ service uoc2@<port number>[.service] [start | stop | status]
$ # for example:
$ service uoc2@3000 start
```

IMPORTANT: The new service must be executed with the port number of UOC instance. The port number is filled between the "@" char and ".service". The service uses the port number because the "systemd" uses a PID file which is filled by the PID of the node instance. This ensures the uniqueness of the process and PID files and allows the capability to start N processes with the same service file description.

See http://serverfault.com/questions/730239/start-n-processes-with-one-systemd-service-file

6.5.3.2.4 Delete the UOC service

To delete the Linux service, you must stop all UOC instance. You can execute this command:

```
$ service uoc2@<port number> stop
```

You can execute a kill command to stop UOC processes or you can execute this command:

```
$ # execute this command with the UOC user
$ su – uoc
$ /opt/uoc2/bin/uoc2 stop
```

Then, execute these commands:

```
$ # disable all UOC instance on the Linux boot
$ systemctl disable uoc2@<port number>
$ # delete the file in the /etc/systemd/system
$ rm /etc/systemd/system/uoc2@.service
```

6.6 Server startup

To start the UOC Server, run the following command with the non-login user uoc.

```
$ su - uoc
$ uoc2 start
```

If you wish to change the TCP port to use, you can give a new port number on the command line. Each UOC server instance can be run with a predefined TCP port number, the port number (useful to support load balancing see:

```
$ uoc2 -p <port_number> start
```

Example:

```
$ uoc2 -p 2222 start
```

Then open the URL http://localhost:3000 (or any host/port combination) to log in the application.

Unified Console Default Port is 3000.

For UOC SDK kit, the port used is 9000. These ports can be fully customizable.

6.7 Create “sudo” policy

You can also start UOC server by using “sudo” command.

You must modified the file /etc/sudoers or execute the following command to edit the file:

```
$ visudo
```

At the end of the file, fill this line to add a new “sudo” command:

```
[USER|%GROUP_NAME] HOST=[(USER_RUN_AS|%GROUP_RUN_AS)] [NOPASSWD:]
COMMANDS
# “USER|%GROUP_NAME”: the user or the group can execute the command (if
you fill “ALL”, all groups/users can execute the command). To fill a
user, enter directly the name of the user (like “uoc”). To fill a group,
enter the name of the group with prefix “%” (like “%uoc”).
# “HOST”: the host which is execute the command
# “USER_RUN_AS|%GROUP_RUN_AS”: the user using when execute the sudo
command.
# “NOPASSWD:”: cannot use password when execute the sudo commands
# “COMMANDS”: the command to execute
```

For example, we can add the UOC sudo command (to start the UOC Server by the launch script) in the file by adding this following line:

```
ALL ALL=(uoc) NOPASSWD: /opt/uoc2/bin/uoc2
```

When another user not belonging to the group “uoc” wants to execute “uoc2 start” command, he must be fill the “runas” parameter:

```
[couchdb@HOST ~]$ sudo -u uoc /opt/uoc2/bin/uoc2 start [-p <port
number>] [-m <memory size>]
```

Other example, we can add the UOC service sudo command (to use the Linux service component with the UOC service) in the file by adding this following line:

```
ALL ALL=NOPASSWD: /sbin/service uoc2*
```

After you can execute the following command:

```
$ sudo service uoc2 start [-p <port number>] [-m <memory size>]
```

For more explanations, you can visit the following URL:

https://www.digitalocean.com/community/tutorials/how-to-edit-the-sudoers-file-on-ubuntu-and-centos.

6.8 Red Hat Linux firewall settings

Netfilter is a host-based firewall for Linux operating systems. It is included as part of the Linux distribution and it is activated by default on RHEL6. This firewall is controlled by the program called iptables. Netfilter filtering takes place at the kernel level, before a program can even process the data from the network packet.

Therefore, when iptables is up and filtering packets, its settings should be modified in order to let the UOC server work properly. In particular, incoming HTTP(s) request on the UOC server port should be allowed.

Please refer to your system admin manual for configuring the firewall if necessary.

```
$ man iptables
```

Usual ports used by UOC:

Component	Description	Port (Default)
Unified OSS Console	Production kit	3000
Unified OSS Console	Development Kit (SDK)	9000
CouchDB	Document database with HTTP	5984
CouchDB	Document database with HTTPS	6984
Identity Provider (IdP)	SAML / SSO server	Check your IdP documentation

Check port open in RHEL 6.x with the following command:

```
sudo iptables -L -n
```

Check port open in RHEL 7.x with the following command:

```
firewall-cmd --list-ports
```

6.9 Licenses management

Official license will be requested from the HP Licensing website: http://enterpriselicense.hpe.com/redirector/home

All licenses keys used by Unified OSS Console product and add-ons can be saved in the following location:

<install_data_dir>/server/public/licenses/licenses.txt

Usually /var/opt/uoc2/server/public/licenses/license.txt

NOTE: These licenses will not be removed or uninstalled, and they will be reused during a new installation or upgrade.

Chapter 7
Validation

If you logged as uoc, you can validate the installation checking the default locations.

The following files should be installed:

```
$ tree -L 1 /opt/uoc2
/opt/uoc2
├── bin
├── client
├── data -> /var/opt/uoc2/data
├── data.kitting
├── install
├── LICENSE.txt
├── json-schema
├── logs -> /var/opt/uoc2/logs
├── node_modules
├── nohup.out
├── scripts
├── server
└── server.js

9 directories, 3 files
$ tree -L 1 /var/opt/uoc2
/var/opt/uoc2
├── client
├── data
├── logs
└── server

4 directories, 0 files
```

- Default Public Data are available in /opt/uoc2/data.kitting
- Default Server Public Data files are available in /opt/uoc2/server/public.kitting
- Default Client Public Data files are available in /opt/uoc2/client/public.kitting

If needed, after an update, you may need to manually apply changes to your own Public Data, Server Public Data or Client Public Data directory. These directories are never deleted or modified to keep your customization.

You can also check the installed packages on your system:

```
$ uoc2 inventory

Packages currently installed:

UOC packages

package                        summary
---------------------------------------------------------------------
uoc-2.3.0-MR                   HPE Unified OSS Console V2.3.0 MR
…
```

UOC inventory commands will list all the UOC kits and their associated UOC Add-ons installed.

This command lists all the UOC kit and maintenance pack kit, add-ons and pre-requisite and their associated version.

Chapter 8
Uninstallation

IMPORTANT:
For an upgrade, it is no more needed to uninstall the previous kit first to replace a new version of UOC. You have to start the installation script that will automatically detect an upgrade and uninstall for you the previous kit before installing the new one.

If you need to uninstall the platform and remove UOC kits, installed packages (and maintenance packs) can be removed interactively with the following command:

```
$ /opt/uoc2/scripts/uninstall.sh
```

It is also possible to uninstall all kits using the yum/rpm command.

NOTE: This uninstallation has not removed the data directory and its possible customization. So, in case of a full uninstallation, you will need to manually clean the data directories.

Chapter 9 Administration

To ease administration of the UOC Server, several commands are available for the platform administrator.

NOTE: These commands are only available for simple deployment (one instance of UOC Server per machine). For advanced deployment with multiples instance of the same machine, multiple UOC Server on several machines… these commands may be not appropriate.

Follow the documentation of advanced tools (node balancer, high availability, …) to put in place such configurations

9.1 UOC

9.1.1 Start UOC Server

9.1.1.1 Command line

IMPORTANT:. You must make sure the CouchDB is running (usually as a service) before starting uoc, else you will not be able to log to uoc and got authentication error messages.

To start the UOC Server, run the following command with the non-login user uoc.

```
$ uoc2 start
```

If you wish to change the TCP port in use, you can give a new port number on the command line:

Each UOC server instance can be run with a predefined TCP port number, the port number (useful to support load balancing):

```
$ uoc2 -p <port_number> start
```

Example:

```
$ uoc2 -p 2222 start
```

It is also possible to set the memory limit for the server instance in Mbytes (sets V8 option --max_old_space_size).

```
$ uoc2 -m <memory_size> start
```

Example:

```
$ uoc2 -m 2048 start
```

9.1.1.2 Start UOC Server as service on RHEL 6.5

To start the UOC Server as a service, run the following command:

```
$ # run this command with the user root
$ service uoc2 start
```

You can write a policy in the sudoers file which is allowed the other users to execute the service:

```
# run this command if the sudoers policy is set (use root
permissions for a command)
$ sudo service uoc2 start
```

If you wish to change the TCP port in use, you can give a new port number on the command line:

Each UOC server instance can be run with a predefined TCP port number, the port number (useful to support load balancing):

```
$ service uoc2 -p <port_number> start
```

Example:

```
$ service uoc2 -p 2222 start
```

It is also possible to set the memory limit for the server instance in Mbytes (sets V8 option --max_old_space_size).

```
$ service uoc2 -m <memory_size> start
```

Example:

```
$ service uoc2 -m 2048 start
```

9.1.1.3 Start UOC Server as service on RHEL 7.2

To start the UOC Server as a service, run the following commands with the port number after the “@” char. Each UOC server instance can be run with a predefined TCP port number, the port number (useful to support load balancing):

```
$ # run this command with the user root
$ service uoc2@<port number> start
$ # or
$ systemctl start uoc2@<port number>
```

```
$ # run this command if the sudoers policy is set (use root
permissions for a command)
$ sudo service uoc2@<port number> start
$ # or
$ sudo systemctl start uoc2@<port number>
```

Example:

```
$ service uoc2 -p 2222 start
```

To set the memory limit for the server instance in Mbytes (sets V8 option --max_old_space_size), you must change the service description file or create a new description file and add the memory size in the “ExecStart” command:

```
[Service]
…
ExecStart=/opt/uoc2/bin/uoc2 start -p %i -m <memory_size>
…
```

Example of service description file:

```
[Unit]
Description=UOC Server (run on the port %i)
After=couchdb.service
Requires=couchdb.service

[Service]
User=uoc
Group=uoc
Type=forking
StandardOutput=journal
StandardError=journal
Restart=no
PIDFile=/opt/uoc2/bin/uoc2-%i.pid
ExecStart=/opt/uoc2/bin/uoc2 start -p %i -m 2048
ExecStop=/opt/uoc2/bin/uoc2 stop -p %i

[Install]
WantedBy=multi-user.target
```

9.1.2 Check UOC Server is running

To check the running UOC server instance(s), the following command is used:

```
$ uoc2 show
```

Example:

```
$ uoc2 show
```

```
   UOC server  25216  uoc  1-04:35:51
   UOC server  25226  uoc  1-05:18:50
   UOC server  25235  uoc  1-04:56:43
   UOC server  25254  uoc  1-05:44:40
   UOC server  25263  uoc  3-15:23:04
```

The output is the list of running instances with the process PID, the user name …

The process PID could be used to stop a specific UOC server instance.

9.1.3 Stop UOC Server

9.1.3.1 Command line

To stop the all UOC Server, run the following command:

```
$ uoc2 stop
```

If multiple UOC servers are running and you want to stop a specific UOC Server instance, run the following command:

```
$ uoc2 stop –p <port number> # <port number> is the port of the UOC
instance what you want stop
```

You can stop UOC instance by another command. You need to identify the PID of the instance and run the following command:

```
$ kill <PID_number>
```

9.1.3.2 Stop UOC Server as service on RHEL 6.5

To stop the all UOC Server by Linux service, run the following command with root privileges or by sudo command:

```
$ service stop uoc2
Or
$ sudo service stop uoc2
```

If multiple UOC servers are running and you want to stop a specific UOC Server instance, run the following command:

```
$ service stop uoc2–p <port number> # <port number> is the port of the
UOC instance what you want stop
Or
$ sudo service stop uoc2–p <port number>
```

9.1.3.3 Stop UOC Server as service on RHEL 7.2

To stop the UOC Server, run the following command and specify the port number of the UOC server instance:

```
$ service stop uoc2@<port number> # <port number> is the port of the UOC
instance what you want stop
Or
$ systemctl uoc2@<port number> stop
```

You can stop UOC instance by another command. You need to identify the PID of the instance and run the following command:

```
$ kill <PID_number>
```

9.1.4 Kits Inventory

To list the installed kits on the platform, an administrator can run the following command:

Note: The command also lists Node.js and CouchDB local packages for troubleshooting purposes.

```
$ uoc2 inventory
Packages currently installed:
UOC packages
package                        summary
---------------------------------------------------------------------
uoc-2.3.0-MR                   HPE Unified OSS Console V2.3.0

Node.js package
package                        summary
---------------------------------------------------------------------
nodejs-4.6.0-1nodesource.el7.centos JavaScript runtime

CouchDB package
package                        summary
---------------------------------------------------------------------
couchdb-1.6.1-4.el7.centos     A document database server, accessible
via a REST                    full JSON API

Redis package
package                        summary
---------------------------------------------------------------------
redis-3.2.4-1.el7.remi         A persistent key-value database
```

9.1.5 Purge unused private workspaces

If you turn on the workspace policy to ruse private workspace, you may need at a time to purge unused workspaces, especially when there is a lot of user in and out in a company. UOC cannot detect these changes and a tool is available for administrator to purge unused private workspace.

Each private workspace object stores their last access date/time and it is possible for an administrator to remove workspaces not accessed from a while and ask for automatic deletion from the CouchDB GUI database.

You can define a duration and a unit (days, months, years…)

Example: Purge private workspace not accessed since 6 months.

```
node ./install/database/purge_private_workspaces.js 6 m
23 private workspaces will be permanently deleted from database.
Do you want to continue (y/n)? y
```

9.2 Apache CouchDB Database

An Apache CouchDB server hosts named databases, which store documents. Each document is uniquely named in the database, and Apache CouchDB provides a RESTful HTTP API for reading and updating (add, edit, delete) database documents.

All the information related to Apache CouchDB can be found on the official web site: http://couchdb.apache.org

9.2.1 Built-in Administration

Administration can easily administrate the database using Futon, the built-in administration interface.

http:<host>:<port>/_utils

Example: http://127.0.0.1:5984/_utils

Futon provides full access to all of CouchDB’s features. Futon lets you create and destroy databases; view and edit documents. Futon is also protected by a user/password. It is strongly recommended to use complex password to secure access direct to this database.

NOTE: this GUI database does not contain sensitive data but definitions used by the web application. So, it is recommended to enable the SSL support, do not use default user/password to increase the security level.

9.2.2 Create / Modify admin user

Admin user is usually created during installation steps but you can change or modify it using the following commands. Note: By default, CouchDB is in “admin party” mode and everybody has privileges to do anything.

The admin user is then used by UOC2 to initialize the different databases of the CouchDB server.

To create an admin user, please run the following command:

```
$ curl -X PUT
<couchdb_protocol>://<couchdb_host>:<couchdb_port>/_config/admins/<admin_us
ername> -d '"<admin_password>"'
```

Example:

```
$ curl -X PUT http://ossv072.gre.hpecorp.net:5984/_config/admins/admin -d
'"admin"'
```

To check if the admin user has been created successfully, please run the following command:

```
$ curl –u <admin_username>:<admin_password>
<couchdb_protocol>://<couchdb_host>:<couchdb_port>/_session
```

Example:

```
$ curl -u admin:admin http://ossv072.gre.hpecorp.net:5984/_session
```

You should get a response as follows:

```
{"ok":true,"userCtx":{"name":"admin","roles":["_admin"]},"info":{"authen
tication_db":"_users","authentication_handlers":["oauth","cookie","defau
lt"],"authenticated":"default"}}
```

9.2.3 Start Apache CouchDB Server

To start the Apache CouchDB Server, run the following command with the user root.

```
$ /etc/init.d/couchdb restart
-or-
sudo service couchdb start
```

9.2.4 Stop Apache CouchDB Server

To stop the Apache CouchDB Server, run the following command with the user root.

```
$ /etc/init.d/couchdb stop
-or-
sudo service couchdb stop
```

9.2.5 Check running Apache CouchDB Server

To stop the Apache CouchDB Server, run the following command with the user root.

```
$ /etc/init.d/couchdb status
-or-
sudo service couchdb status
```

9.3 Redis Server

9.3.1 Start Redis Server

To start the Redis Server, run the following command with the user root. Depending on how Redis was installed (in 4.1.5 and 5.1.5)

```
$ /etc/init.d/redis restart
-or-
sudo service redis start
```

If installed from sources:

```
sudo service redis_6379 start
```

9.3.2 Stop Redis Server

To stop the Redis Server, run the following command with the user root. Depending on how Redis was installed (in 4.1.5 and 5.1.5)

```
$ /etc/init.d/redis stop
-or-
sudo service redis stop
```

If installed from sources:

```
sudo service redis_6379 stop
```

9.3.3 Check running Redis Server

To stop the Redis Server, run the following command with the user root. Depending on how Redis was installed (in 4.1.5 and 5.1.5)

```
$ /etc/init.d/redis status
-or-
sudo service redis status
```

If installed from sources:

```
sudo service redis_6379 status
```

9.3.4 Advanced configuration

As is, Redis is rather insecure. It is recommended to go through the following steps to harden your installation in addition to the common procedures (e.g. tune firewall rules, set right UNIX permissions to the files, etc.).

9.3.4.1 Enable authentication

By default, Redis has no authentication at all. Anybody can connect the server and submit commands. We recommend to require users to authenticate beforehand to add an additional layer of security, especially if your Redis server is exposed to the network.

To do so, please edit Redis configuration file (**/etc/redis.conf** or **/etc/redis/<port>.conf** depending on your installation) and search for the directive **requirepass.**

```
# Require clients to issue AUTH <PASSWORD> before processing any other
# commands.  This might be useful in environments in which you do not trust
# others with access to the host running redis-server.
#
# This should stay commented out for backward compatibility and because most
# people do not need auth (e.g. they run their own servers).
#
# Warning: since Redis is pretty fast an outside user can try up to
# 150k passwords per second against a good box. This means that you should
# use a very strong password otherwise it will be very easy to break.
#
#requirepass foobar
```

Uncomment the line and change the default password for very strong one as advised in the comments to prevent a brute force attack. Save and restart the Redis server.

NOTE: UOC server must then be configured to use this password to authenticate to the Redis server. Please see Chapter 9.4 Notification Server (Redis server)

9.3.4.2 Increase default log level

Default Redis log level is notice. We recommend to increase it to verbose to log additional information like accepted connections for audit purposes.

To do so, please edit Redis configuration file (**/etc/redis.conf** or **/etc/redis/<port>.conf** depending on your installation) and search for the directive **loglevel.**

```
# Specify the server verbosity level.
# This can be one of:
# debug (a lot of information, useful for development/testing)
# verbose (many rarely useful info, but not a mess like the debug level)
# notice (moderately verbose, what you want in production probably)
# warning (only very important / critical messages are logged)
loglevel notice
```

Change notice to verbose. Save and restart the Redis server.

Log sample

```
21898:M 13 Sep 10:31:57.501 - Accepted 16.17.100.85:50056
21898:M 13 Sep 10:31:57.503 - Accepted 16.17.100.85:50057
21898:M 13 Sep 10:31:57.509 - Accepted 16.17.100.85:50058
21898:M 13 Sep 10:31:57.519 - Accepted 16.17.100.85:50059
21898:M 13 Sep 10:31:57.529 - Accepted 16.17.100.85:50060
21898:M 13 Sep 10:31:57.529 - Accepted 16.17.100.85:50061
21898:M 13 Sep 10:31:57.610 - Accepted 16.31.78.180:54813
21898:M 13 Sep 10:31:57.610 - Accepted 16.31.78.180:54814
21898:M 13 Sep 10:32:02.167 - 8 clients connected (0 slaves), 921888
bytes in use
```

9.3.4.3 Disable commands

By default, all Redis commands are available for all clients, including for instance CONFIG that allows to reconfigure Redis server on the fly. We recommend to disable such powerful commands in production as well as disable unused commands to reduce your attack surface.

To do so, please edit Redis configuration file (/etc/redis.conf or /etc/redis/<port>.conf depending on your installation) and search for the directive rename-command.

```
# Command renaming.
#
# It is possible to change the name of dangerous commands in a shared
# environment. For instance the CONFIG command may be renamed into
something
# hard to guess so that it will still be available for internal-use
tools
# but not available for general clients.
#
# Example:
```

```
#
# rename-command CONFIG b840fc02d524045429941cc15f59e41cb7be6c52
#
# It is also possible to completely kill a command by renaming it into
# an empty string:
#
# rename-command CONFIG ""
#
# Please note that changing the name of commands that are logged into the
# AOF file or transmitted to slaves may cause problems.
```

Add below **rename-command** directives to restrict available Redis commands to clients. See Redis command reference at http://redis.io/commands.

```
rename-command BGREWRITEAOF ""
rename-command BGSAVE ""
rename-command CLIENT ""
rename-command COMMAND ""
rename-command CONFIG ""
rename-command DBSIZE ""
rename-command DEBUG ""
rename-command FLUSHALL ""
rename-command FLUSHDB ""
rename-command INFO ""
rename-command LASTSAVE ""
rename-command MONITOR ""
rename-command ROLE ""
rename-command SAVE ""
rename-command SLAVEOF ""
rename-command SLOWLOG ""
rename-command TIME ""
rename-command EVAL ""
rename-command EVALSHA ""
rename-command SCRIPT ""
```

Save and restart the Redis server.

9.3.4.4 SSL tunneling

Redis does not support SSL/TLS natively. Though, a SSL tunnel can be set up in front of the Redis server, allowing to communicate with UOC2 servers in an encrypted fashion.

You can use any SSL tunnel tool you want, we present below an example using **stunnel**.

IMPORTANT: The next steps assumes you have a private key and a SSL/TLS certificate for stunnel as well as the certificate of your certification authority.

IMPORTANT: Your Redis server must listen the loopback interface only (bind 127.0.0.1 in the Redis configuration).

1. Install stunnel

```
# yum install stunnel.x86_64
…
# stunnel -version
stunnel 4.29 on x86_64-redhat-linux-gnu with OpenSSL 1.0.1e-fips 11 Feb
2013
```

2. Create a configuration file for stunnel

```
# touch /etc/stunnel/stunnel.conf
```

3. Edit stunnel configuration file

IMPORTANT: The following configuration is a sample. You need to customize it based on your own need.

```
# nano /etc/stunnel/stunnel.conf
;Root directory in which the stunnel process runs
chroot = /var/run/stunnel
;User that the stunnel process runs as, nobody is a restricted system account
setuid = nobody
;Group that the stunnel process runs as, nobody is a restricted system account
setgid = nobody
;File in which stunnel saves its process ID, relative to chroot
pid = /stunnel.pid

[redis]
;Port stunnel listens to
accept = 6380
;Port Redis listens to
connect = 127.0.0.1:6379
;Stunnel SSL certificate
cert = /etc/stunnel/ssl/server.crt
;Stunnel private key
key = /etc/stunnel/ssl/server.key
```

```
;Disable SSLv2
options = NO_SSLv2
;Disable SSLv3
options = NO_SSLv3
;Verify peer certificates
verify = 3
;Certificate authority file
CAfile = /etc/stunnel/ssl/CA.crt
```

4. Run stunnel

```
# stunnel /etc/stunnel/stunnel.conf
```

By default, stunnel logs in /var/log/secure. This path can be changed by configuration.

To stop stunnel, simply kill the process

```
# kill `cat <path_to_stunnel_pid_file>`
```

TIP: According to the certificate you use, you may encounter this error:

```
Could not load DH parameters from …
Diffie-Hellman initialization failed
Error reading certificate file: …
SSL_CTX_use_certificate_chain_file: error:0906D06C:PEM
routines:PEM_read_bio:no start line
```

The following command creates the missing DH parameters that need to be appended at the end of your certificate.

```
dd if=/dev/urandom count=2 | openssl dhparam -rand - 512
```

IMPORTANT: UOC server must be configured to use Redis with SSL/TLS support.
See 17.11.10 UOC ←→ Notification server (Redis server)

Chapter 10
Platform Configuration

It is possible to configure these following setting for the UOC platform.

10.1 UOC Server

The UOC Server has a configuration file where it is possible to customize some parameters. The configuration file is stored in <install_dir>/server/public/conf/config.json

```
Example:
   {…
     "server": {
    "protocol" : "http",
    "port" : "3000"
    },
…}
```

Where

- **Protocol** is the protocol used by the server (http or https). It can be overridden by an environment variable named PROTOCOL.
- **Port** is the port of the server (default is 3000). It can be overridden by an environment variable named PORT.

10.2 Request Timeout

The UOC Server has a configuration file where it is possible to customize the timeout used by http requests. The configuration file is stored in <install_dir>/server/public/conf/config.json

```
Example:
   {…
     "server": {
     "timeout" : 0
    },
…}
```

Where

- **Timeout** is the timeout in seconds allowed for all http/https requests. Default is 0s (unlimited). It can also be overridden by the environment variable TIMEOUT.

10.3 Apache CouchDB Database

Parameters about the CouchDB database can be customized in a configuration file. The configuration file is stored in <install_dir>/server/public/conf/config.json

Here is an example of database configuration (CouchDB Server):

```
{
…
    "database": {
        "protocol": "http",
        "host": "127.0.0.1",
        "port": "5984",
        "username": "user",
        "password": "user"
    }
…
}
```

Where:

- protocol is the protocol used by the CouchDB server (http or https). (Default is http)
  It is strongly recommended to enable the SSL support to access to the GUI database. It is recommended to configure the TLS 1.2 to grant a secure configuration.
- host is the host name or IP address of the CouchDB server.
- port is the port of the CouchDB server (Default is 5984).
- username is the name of CouchDB user used by UOC server (Default is user).
- password is the password of the CouchDB user used by UOC server (Default is user).

10.4 Notification Server (Redis Server)

Parameters about the notification server can be customized in a configuration file. The configuration file is stored in <install_dir>/server/public/conf/config.json

Here is an example of Notification Server configuration (Redis Server)

```
{
…
    "notifications": {
        "publishSubscribeServer": {
            "active": true,
            "host": "ossv086.gre.hpecorp.net",
            "port": 6379,
            "password": "ZWJiYmE1NTM1ZDk3YzM5ZDk3NzM3ZDVl",
        }
    }
…
}
```

Where:

- active is a boolean, true means UOC will use a publish-subscribe server for notifications (Redis Server, mandatory when multiple UOC servers), false means it is disabled (recommended if only one UOC server).
- host is the host name or IP address of the publish-subscribe server for notifications (Redis Server)
- port is the port of the publish-subscribe server for notifications (Redis Server) (Default is 6379).
- password is the password of your Redis server (requirepass property in the redis configuration). Leave empty if you have not set any password.

10.5 Body Parser Size

The UOC Server has some advanced configuration for customization (only if you met some issues, else it is recommended to use the default values)

The configuration file is stored in <install_dir>/server/public/conf/config.json

It is possible to customize size of the body for POST requests. It is possible to controls the maximum request body size.

Example of config.json:

```
  {...
        "bodyParser": {
                "limit" : "500Kb"
  },
...}
```

Where

- **limit** is the maximum request body size in bytes. If this is a number, then the value specifies the number of bytes; if it is a string, the value is converted from string to bytes (ex: ‘1Kb’ = 1024, ‘500Kb’ = 512000, ‘1Mb’=1024000…). Defaults to '100kb'.

10.6 Supported Language (L10N)

UOC allows the platform administration to setup all supported language by the UOC Server and some policy for optimization.

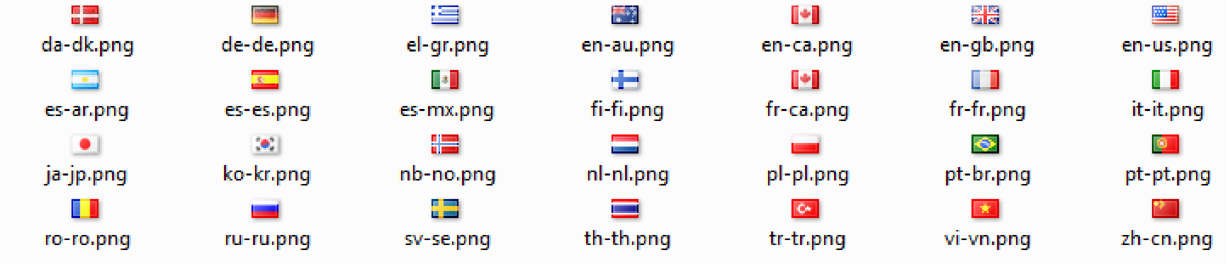
It define the list of available language for the platform (name and language code) and an optional property to keep all localization in a single file. This is interesting to add customized languages not supported by default by UOC.

staticL10nFile is optional and its default value is false. If "staticL10nFile" value is true, L10N file will not be generated at server startup meaning that a complete L10N file must be provided and added in **<install_dir>/client/l10n** with the following naming rule: [languageCode].json where [languageCode] is replaced by the language code. (Example of L10N filename: es-es.json) .

Example of config.json
It defines US English and French and Spanish locale with “staticL10nFile” option enabled for Spanish..

```
{…
  "languages" : [{
    "name": "English",
    "languageCode": "en-us"
  }, {
    "name": "Français",
    "languageCode": "fr-fr"
  }, {
    "name": "Español",
    "languageCode": "es-es",
    "staticL10nFile": true
  }],
…}
```

NOTE: By default, several flags icons are available in <installDir>/client/public/images/languages



da-dk.png de-de.png el-gr.png en-au.png en-ca.png en-gb.png en-us.png
es-ar.png es-es.png es-mx.png fi-fi.png fr-ca.png fr-fr.png it-it.png
ja-jp.png ko-kr.png nb-no.png nl-nl.png pl-pl.png pt-br.png pt-pt.png
ro-ro.png ru-ru.png sv-se.png th-th.png tr-tr.png vi-vn.png zh-cn.png

These icons of the selected language will be displayed in the menu item preference of the user, and he will be able to dynamically switch the language. If a icon is not already present in the list, it is possible to extend these icons.

IMPORTANT:. **staticL10nFile** property applies only if L10N optimization is enabled (see 10.7 L10N / Languages Optimization)

10.7 L10N / Languages Optimization

The UOC platform is able to enable or disable the language loading optimization at server startup. By default this optimization is enabled but it can be disabled by setting "optimizeL10n" to false.

When L10N optimization is enabled, all language files dynamically found on the platform are concatenated in one single language file in <install_dir>/client/l10n and loaded in one request at the beginning of the application.

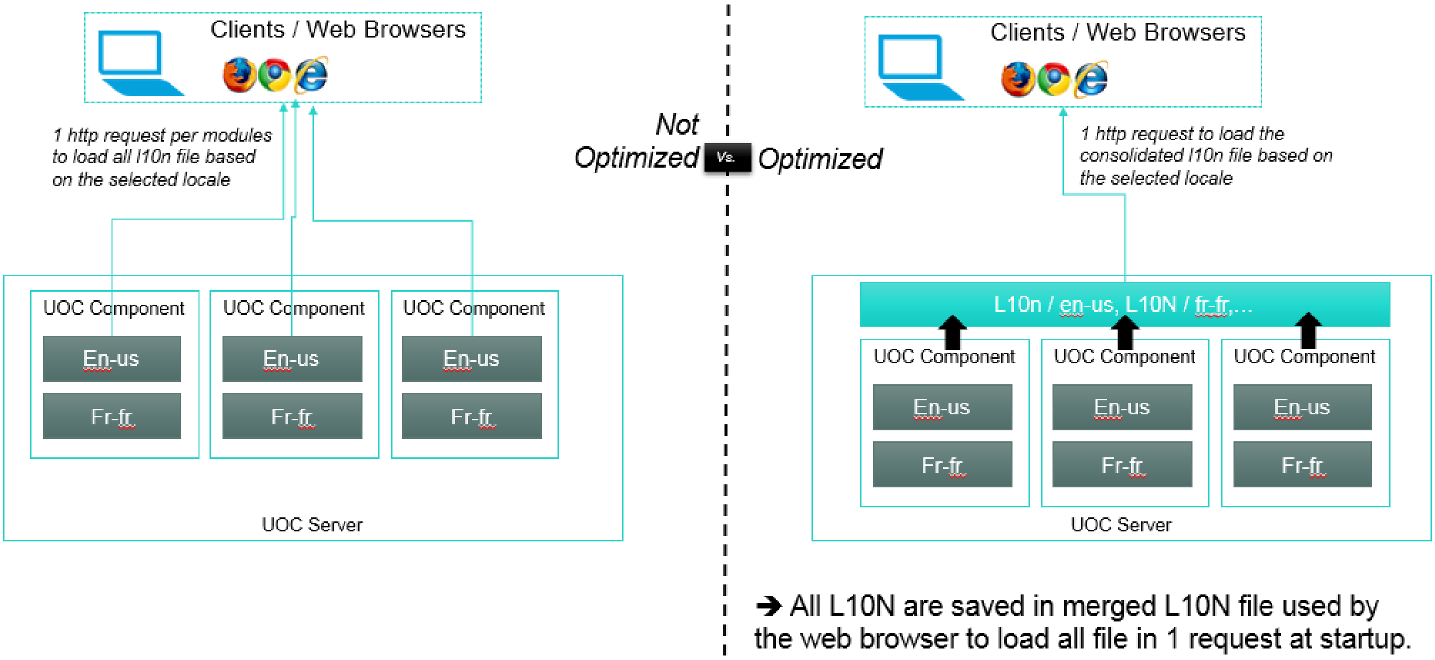

Clients / Web Browsers
1 http request per modules to load all l10n file based on the selected locale
UOC Component
En-us
Fr-fr
UOC Server
Not Optimized
Vs.
Optimized
Clients / Web Browsers
1 http request to load the consolidated l10n file based on the selected locale
L10n / en-us, L10N / fr-fr,…
UOC Component
En-us
Fr-fr
UOC Server
➔ All L10N are saved in merged L10N file used by the web browser to load all file in 1 request at startup.

Figure 13: Language Optimization

L10N file is not generated for this language if staticL10nFile is set to true (see chapter 9.4).

Example of <installdir>/server/public/conf/config.json with Language loading optimization disabled.

```
{…
        "optimizeL10n": false,
…}
```

10.8 Platform / User Preferences

User preferences services is in charge of managing the user / platform preference of the connected user. These preferences allow to select the right look and feel for the user interface. It is a way to completely rebrand the UOC application for integrators.

TIP: There is a default platform preferences, but it is easy to specify some conditions based on tenant identifier and/or roles identifier to apply a different setting. Only the conditional defined setting will override the global properties.

Example: So, it is possible to setup a theme by tenant keeping all other existing settings.

10.8.1 Overview

Platform administrators can configure them in a user preferences configuration file which can contain these options:

Preference	Default	Description
title	Unified OSS Console	Title of the application displayed in the title page, login page and the main menu.
version	2.3	Version associated to the application displayed in the login page. If this value is not defined. The version is hidden (no visible version badge) **Note**: You can use the following keyword to dynamically support the UOC Version (**<UOC:VERSION>**). As this user preference configuration is kept after uninstall/reinstall, it is needed if you want to display at login the last version of UOC.
link	/	URL available if the user clicks on the title. It usually redirects to the home page of the application but it can define another internet address.
language	en-us	Default language to use if the setting of the web browser is not available.
theme	hpe_light	Default theme to apply in the application
showMenuBar	true	Show or hide the menu bar.
menuBar	hpe-menu-bar	Default main menu to use. This main menu can be customized in add-ons
showWorkspaceManager	true	Indicates if after login the main page contains the available workspaces list and allows workspaces operations or if this page needs to stay empty. Useful in case of custom menu “Operations” that expose pre-defined workspace to select.
initialWorkspace	(not defined)	Indicates if a workspace need to be open by default after login or if the user has to be redirected to the main page. Note: this option is not compatible with **lastAccessedWorkspace**
lastAccessedWorkspace	false	This option indicates if the end user will reuse his last access workspace to start his session after a new login. During logout, we keep in the user’s cookie information to the last workspace used. Note: this option is not compatible with initialWorkspace

enablePrivateWorkspace	false	Enabling a private workspace, let a user save a personal and private copy of a public existing workspace. It will be the only one to be able to access to it. It is useful when user want to have personal setting to work with workspaces.
conditions	[ ]	Conditions can be used to set specific platform preference for some specific users based on their tenant identifier and/or roles. You can find an example of condition use in the installation guide.

Table 1: User Preferences

By default, the service returns user preferences defined in the configuration file in <installdir>/server/public/conf/user-preferences.json

```
{
        "title":"Unified OSS Console",
        "version":"<UOC:VERSION>",
        "link":"/",
        "language":"en-us",
        "theme":"hpe_light",
        "menuBar":"hpe-menu-bar",
        "showWorkspaceManager": true,
        "showMenuBar": true,
        “initialWorkspace”: “workspaceId”,
        "enablePrivateWorkspace": false,
        "lastAccessedWorkspace": false
}
```

TIP: <UOC:VERSION> is a keyword that is dynamically resolved by the Unified Console and represents the current version of the product. It will ease the upgrade and detailed version of maintenance pack.

“version”:“Operations Portal <UOC:VERSION” will display “Operations Portal 2.3”

10.8.2 Initial workspace

A Platform administrator can force an initial workspace for all user on this server. Usually, these user does not have access to the workspace manager and login and logout in a very simple way with a pre-defined workspace for executing their job.

To setup an initial workspace, the platform administrator needs to specify the workspace identifier in the user-preference configuration file.

By default, for compatibility reason, this feature is disabled. To enable it, as a platform administrator, you can edit the following user preference file.

```
Example of <installdir>/server/public/conf/user-preferences.json

{...
        " initialWorkspace": “my-demo-wks”
...}
```

10.8.3 Private workspace

A Platform administrator can enable / disable this feature to allow end user to keep private workspace. Default behavior is to have all workspace available to all users and protect by role based access only.

Enabling a private workspace, let a user save a personal and private copy of a public workspace. It will be the only one to be able to access to it. It is useful when user want to have personal setting to work with workspaces.

By default, for compatibility reason, this feature is disabled. To enable it, as a platform administrator, you can edit the following user preference file.

Example of <installdir>/server/public/conf/user-preferences.json

```
{...
        " enablePrivateWorkspace": true
...}
```

10.8.4 Last Accessed Workspace

This option indicates if the end user will reuse his last access workspace to start his session after a new login. During logout, we keep in the user's cookie information to the last workspace used.

IMPORTANT:. Last accessed workspace option is not compatible with initial workspace option. As soon as a platform admin defines an initial workspace, the last access workspace feature is disabled.

By default, for compatibility reason, this feature is disabled. To enable it, as a platform administrator, you can edit the following user preference file.

Example of <installdir>/server/public/conf/user-preferences.json

```
{...
        " lastAccessedWorkspace": true
...}
```

10.8.5 Conditional User Preferences

You can use conditions to use specific user preferences for a tenant identifier and/or a set of roles. Properties defined inside a condition will merge with the default user preference

A condition must specify a set of requiredRoles, or a tenant_id, or a tenant_id and a set of requiredRoles

NOTE: If you have a tenant_id with no requiredRoles, user preferences will be applied for every users with that tenant_id regardless of their roles.

If you specify requiredRoles, the user preference will be applied to users having exactly these roles.

Example of <installdir>/server/public/conf/user-preferences.json

```
{
   "title": "Unified OSS Console",
   "version": "<UOC:VERSION>",
   "link": "/",
   "language": "en-us",
   "theme": "hpe_light",
   "menuBar": "hpe-menu-bar",
   "showWorkspaceManager": true,
   "showMenuBar": true,
   “initialWorkspace”: "workspaceId",
   "enablePrivateWorkspace": false,
   "lastAccessedWorkspace": false,
   "conditions": [{
      "tenant_id": "X",
      "requiredRoles": [],
      "initialWorkspace": "my_wks"
   }, {
      "requiredRoles": ["roleA", "roleB"],
      "language": "fr-fr",
      "theme": "hpe_dark"
   }]
}
```

The first condition will override the default user preference property: **initialWorkspace** for every users belonging to the “X” tenant.

The second condition will override the default user preference properties: **language** and **theme** for every users having exactly **roleA** and **roleB** as roles.

10.9 UOC Data Import

During startup of the UOC server, there are 2 steps:

1. The server browse all available plugins to contact all associated domain server (or data server), and collect all available value pack and their GUI resources (like workspaces and views).

   All graphical resources (Data UI) found are imported into the GUI database for sharing and usage with all users.

NOTE: The reference is the GUI database to access to the graphical resource. Backup this database is strongly recommended to avoid any lost.

See 13.3 GUI Database (Apache CouchDB for detailed options for backup UOC data

All value pack definitions (Metadata UI) are loaded dynamically in the UOC server and kept until the stop of the server.

2. The server browse all data defined locally in a specific local directory **<install_dir>/data**
   This directory defines several definitions used to initialize the GUI database:

- Workspace categories
- Launch categories
- Launches
- Roles
- Permissions
- States
- Users (local authentication mode only)

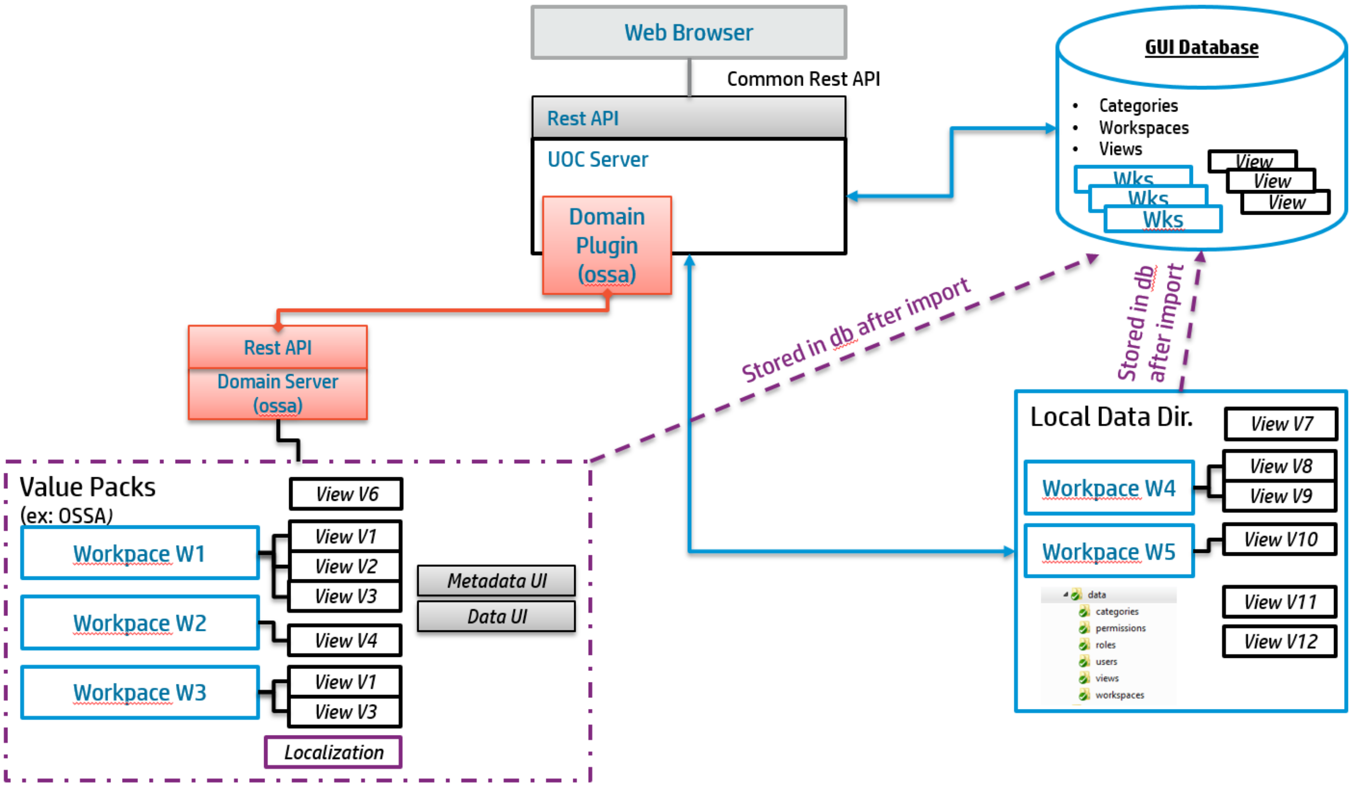


Figure 14: Data Import Overview with OSSA Server and plugin installed

NOTE: Default behavior executes these 2 steps

1. Browse the plugin and their domain server to collect and import resources from servers, and
2. Import ONLY additional data present in the local directory

It is possible to customize the data import policy to ignore one specific step and give priority to data found on the server or the local directory.

An administrator can easily choose to always override in the GUI database data using the last data found on the server during the UOC start and make sure the reference is always on the server and up to date for UOC.

To customize the data import policy, edit the configuration file: <install_dir>/server/public/conf/config.json

```
...
   "startup": {
      "loadLocalUIData": true,
      "overwriteLocalUIData": false,
      "loadRemoteUIData": true,
      "overwriteRemoteUIData": false
   }
...
```

Property	Value	Default	Description
loadLocalUIData	true | false	true	Import local graphical data from <install_dir>/data into the GUI database
overwriteLocalUIData	true | false	false	Override GUI database with local graphical data if the same identifier is found. Reference is the local data directory.
loadRemoteUIData	true | false	true	Import graphical data retrieved from packages in all defined server into the GUI database
overwriteRemoteUIData	true | false	false	Override GUI database with server graphical data if the same identifier is found. Reference is all data servers.

When the platform administrator starts the UOC server, the console displays the current settings.

It is strongly recommended to turn off override options and make sure identifier of objects (workspace, view…) are not duplicated between the 2 sources (local and server).

The GUI database is the reference for all the graphical objects of UOC. Server’s packages usually provide default workspaces and views, but these objects can be customized by integrator, operators and view designer.

10.10 UOC Data Export

UOC has an export service allowing to export (as CVF file) and download data coming from plugins (i.e. metrics based on dimensions or objects) to a file.

IMPORTANT: Configuration needs to be adjusted depending on the needs in terms of concurrent users and volume of data to export. This will have an impact on performance and the HW must be correctly sized.

This export service supports several options. To customize them, please edit the configuration file <install_dir>/server/public/conf/config.json

```
...
  "exportOptions": {
    "maxSimultaneousUsers": 20,
    "maxNumberOfRows": 300000,
    "requestTimeout": “5m”,
    “csv”: {
        "delimiter":",",
        "textQualifier":"\"",
        "lineBreak":"\n",
        "nullValue": "null";
        "ignoreHeadersParameter": false";
        “queryParameters": {
            "show": true,
            "fromToDateFormat":"YYYY-MM-DDTHH:mm:ss",
            "granularity": {
                "durationUnit":"s",
                "durationFormat":"m [min] ss [sec]"
              }
          }
    }
  }
...
```

Property	Value	Default	Description
maxSimultaneousUsers	integer	20	Maximum number of users that can perform export requests at a time
maxNumberOfRows	integer	300000	Maximum number of records per export request
requestTimeout	string	5m	Maximum time for requests to plugins to be completed (a string can be used with ‘h’ (hours), ‘m’ (minutes), ‘d’ (days), ‘s’ (seconds) Ex: 10h, 2h15m, 3d
csv	object		CSV export options

CSV export options are the following ones:

Property	Value	Default	Description
delimiter	string	,	Value delimiter
textQualifier	string	“	Wrapper around values
lineBreak	string	\n	End of line
nullValue	string	null	Value to use for null values
ignoreHeadersParameter	boolean	false	Ignore provided column names for header. If set to true, column id will be used
queryParameters	object		Query parameters in csv
show	boolean	false	Show or hide query parameters
fromToDateFormat	string	YYYY-MM-DDTHH:mm:ss	Specify date format for “From Date” and “To Date” parameters Ex: YYYY-MM-DDTHH:mm:ssZ
granularity	object		Granularity display options
durationUnit	string		Granularity unit Ex: s (seconds), m (minutes)
durationFormat	string		Granularity format Ex: m [min] ss [sec] If no format specified, granularity will be displayed without unit and format
durationsOptions	object		
trim	boolean	false	Leading tokens are not trimmed when they have no value

Chapter 11
Platform Monitoring

The UOC Server has an engine in charge of monitoring all resources of the UOC Platform and log these information. It is also possible to customize the format of the logged line (layout pattern).

This platform monitoring is disabled by default but it can be enabled by a platform administrator and setup to identify which components are monitored.

11.1 Overview

UOC embed an engine in charge of the platform monitoring. If the platform administrator enable this feature, several jobs will be scheduled to check periodically all or a set of the uoc components. Each jobs can be scheduled with different policy and time period. Some dedicated logs are available to store these information to enhance the possible troubleshooting of the uoc platform. These logs have been designed especially to support the standard for message logging (syslog).

The platform monitoring is able to monitor the following resources of UOC:

- Identity provider (IdP)
- UOC Server (Node JS)
- GUI Database (CouchDB)
- Notification Server (Redis) (if installed)
- Active Plugins of the UOC platform

As soon as the monitoring is enabled, it is possible to define by resources a polling period to check the availability of the resource and the response time. A monitoring logger will persist the information in a JSON format to be ready to be processed and integrated into a syslog tools.

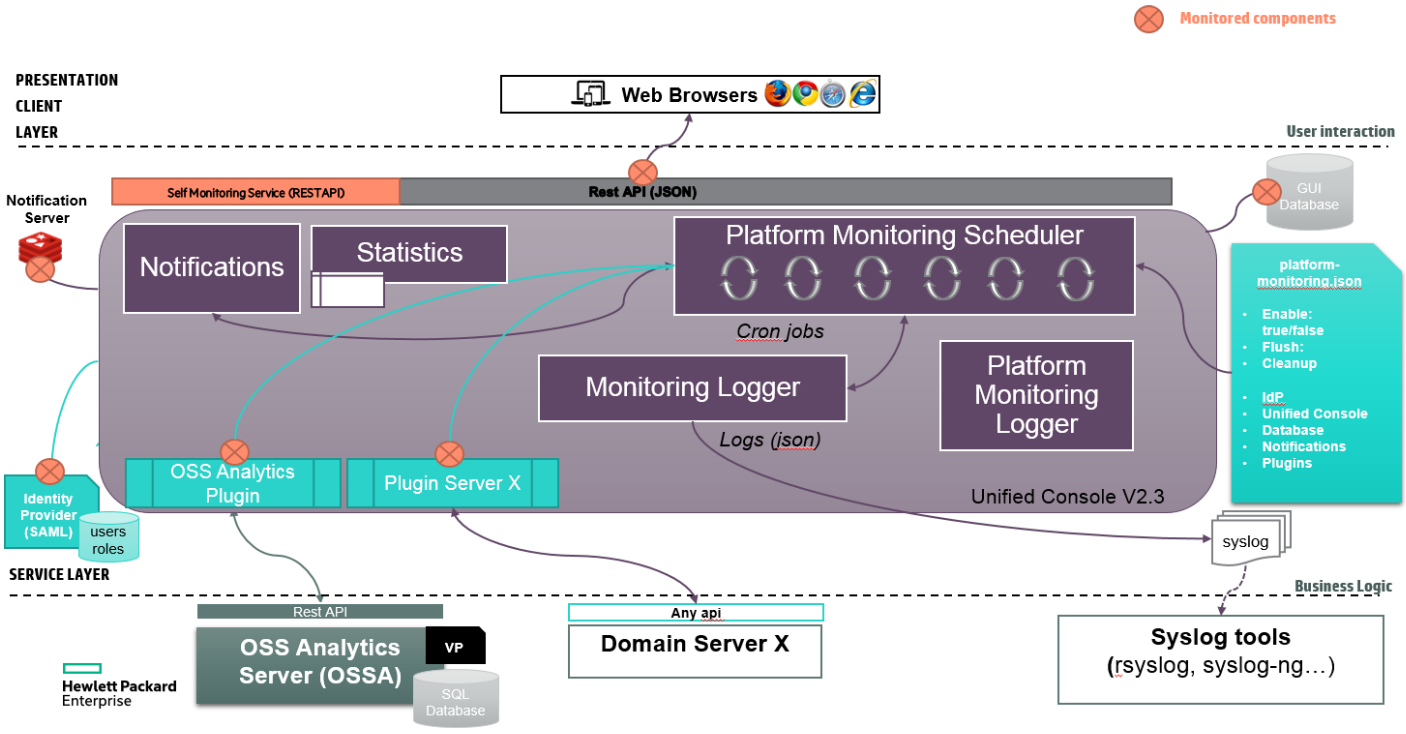


Figure 15: Platform Monitoring Overview

11.2 Configuration

The UOC Server has a configuration file where it is possible to customize the parameters of the monitoring (enable/disable the monitoring features and specify the time zone). The configuration file is stored in <install_dir>/server/public/conf/platform-monitoring.json.

```
{
    "enable": true,
    "cronTimeZone": "GMT",
    "server": {
        "cronTime": "0 */15 * * * *"
    },
    "database": {
        "cronTime": "0 */15 * * * *"
    },
    "idp": {
        "cronTime": "0 */15 * * * *"
    },
    "plugins": [
        {
            "id": "*",
            "cronTime": "0 */15 * * * *"
        },
        {
            "id": "ossa",
            "cronTime": "0 */15 * * * *"
        },
        {
            "id": "training",
            "enable": false,
            "cronTime": "0 */15 * * * *"
        }
    ]
}
```

Where:

- Enable is a Boolean that allow to start the monitoring for all the configured components.
- cronTimeZone is the global time zone you want to use for your CRON tasks. This one is optional, if nothing is specified then the default time zone will be: “GMT”.

The configuration is specific for each component you want to monitor:

- UOC server that we will name “server”
- Notification server,
- Internal database (CouchDB)
- Identity Provider (SAML)
- UOC Plugins.

It is possible to define a CRON time to periodically check the status of each components following this syntax:

Second	Minute	Hour	Day of month	Month	Day of week
(0-59)	(0-59)	(0-23)	(1-31)	(1-12)	(0-6)

As soon as a component is monitored, logs will be automatically generated based on the cron time and indicates useful information to check the status and response time of each components (see 11.3 **Monitoring Logger**)

NOTE: By default, a real-time notification is sent to the platform administrator when a component has been detected not functional. These notifications are customizable (see 11.5 Notifications)

11.2.1 Server

The UOC server can now be monitored if you add in the configuration the information below.

```
{
      "server": {
            "enable" : true,
            "cronTime": "0 */15 * * * *"
    },
…
}
```

Where:

- CronTime is the frequency you want to monitor your components, it is defined with the usual CRON syntax.
- Enable is a Boolean that allow to start the monitoring for this component.

With this configuration a CRON task will be run every 15 minutes to monitor the UOC server. Which means every 15 minutes the server will try to get the login page of UOC. The response time and the status of the request will then be measured and logged.

11.2.2 Internal database (CouchDB)

The internal UOC database can be monitored if you add in the configuration the information below.

```
{
     "database": {
           "enable" : true,
           "cronTime": "0 */15 * * * *"
    },
…
}
```

Where:

- CronTime is the frequency you want to monitor your components, it is defined with the usual CRON syntax.
- Enable is a Boolean that allow to start the monitoring for this component.

With this configuration a CRON task will be run every 15 minutes to monitor the internal UOC database. Which means every 15 minutes the UOC server will try to ping the database. The response time and the status of the request will then be measured and logged.

11.2.3 Notification Server (Redis)

The notification Server (if installed in the solution) can be monitored if you add in the configuration the information below.

```
{
     "notificationServer": {
           "enable" : true,
           "cronTime": "0 */15 * * * *"
    },
…
}
```

Where:

- CronTime is the frequency you want to monitor your components, it is defined with the usual CRON syntax.
- Enable is a Boolean that allow to start the monitoring for this component.

With this configuration a CRON task will be run every 15 minutes to monitor the notification server. Which means every 15 minutes the UOC server will try to ping the notification server. The response time and the status of the request will then be measured and logged.

11.2.4 Identity provider (IdP)

The Identity Provider can be monitored if you add in the configuration the information below.

```
{
     "idp": {
          "enable" : true,
          "cronTime": "0 */15 * * * *",
          "healthCheckUrl": "http://...."
    },
     …
}
```

Where:

- CronTime is the frequency you want to monitor your components, it is defined with the usual CRON syntax.
- Enable is a Boolean that allow to start the monitoring for this component.
- HealthCheckUrl is an URL that will be used to check if the identity provider is up. If the healthCheckUrl property is absent or is empty, the entry point for SAML requests will be used.

With this configuration a CRON task will be run every hour to monitor the identity provider. Which means every hour the server will try to ping this server. The response time and the status of the request will then be measured and logged.

11.2.5 Plugins

The configuration to monitor the plugins of the UOC server is different of the four others. Instead of using an object we are using a list of object, this way you can define a different configuration for every plugins.

```
   { …
     "plugins": [
          {
            "id": "*",
            "cronTime": "0 */15 * * * *"
          },{
            "id": "ossa",
            "enable" : false,
            "cronTime": "0 */15 * * * *"
          }
    ],
…}
```

Where

- Id is the id of the plugin you want to monitor. "*" is the default value if you want to select all the plugins that have been configured on the UOC server. You can specify an official plugin Identifier (ossa, training, ossm…)
- CronTime is the frequency you want to monitor your components, it is defined with the usual CRON syntax.

11.3 Monitoring Logger

The platform monitoring of UOC logs periodic status and information using a customizable logger to ease troubleshooting of the UOC Platform.

IMPORTANT: These logs are only accessible to platform administrator for troubleshooting reason and may content sensitive information related to the privacy.

All monitoring logs can be found under <install_data_dir>/logs/monitoring.log

It is possible to refine the logging policy for this server logs customizing the “server-logger “ appender in the file : <install_data_dir>/server/public/conf/log4js.json

Default level is info and you and customize the format on the line (layout pattern).

Any changes on this file will be dynamically apply without restart the UOC server.

Default logging policy:

```
    {
        "type": "file",
        "filename": "logs/monitoring.log",
        "level": “INFO",
        "maxLogSize": 2048000,
        "backups": 3,
        "category": "monitoring-logger",
        "layout": {
                "type": "pattern",
                "pattern": "%m"
        }
    }
```

11.4 Platform Monitoring Logger

The UOC Server has customizable logger to ease troubleshooting of the UOC Platform Monitoring

IMPORTANT: These logs are only accessible to platform administrator for troubleshooting reason and may content sensitive information related to the privacy, especially if the debug level is enabled.

All logs for the UOC server can be found under <install_data_dir>/logs/ platform-monitoring.log

It is possible to refine the logging policy for this server logs customizing the “platform-monitoring-logger “ appender in the file : <install_data_dir>/server/public/conf/log4js.json

This log file supports multiple level of log (info, warning, error, warning, debug). Default level is warning and you and customize the format on the line (layout pattern).

Any changes on this file will be dynamically apply without restart the UOC server.

Default logging policy:

```
{
    "type": "file",
    "filename": "logs/ platform-monitoring.log",
    "level": "WARN",
    "maxLogSize": 2048000,
    "backups": 3,
    "category": " platform-monitoring-logger"
}
```

11.5 Notifications

The UOC server will automatically send you default notifications if one of the components you decided to monitor crashed. However you can send your own personal notifications if you write them in the configuration file in <install_dir>/server/public/conf/platform-monitoring.json.

```
{
    "enable": true,
    "cronTimeZone": "GMT",
    "server": {
        …
        "notifications": [{
            "origin": "server",
            "type": "alert",
            "id": "platform-monitoring-roles-notification-uoc",
            "roles": ["Platform Administrator"],
            "sendToRoles": ["Platform Administrator"],
            "level": "error",
            "title": "The UOC server is not available",
            "keywords": ["platform-monitoring", "UOC server"],
            "display": {
                "type": "toast"
            }
        }]
    },
    "database": {
        …
        "notifications": [{
            "origin": "server",
            "type": "alert",
            "id": "platform-monitoring-roles-notification-database",
            "roles": ["Platform Administrator"],
            "sendToRoles": ["Platform Administrator"],
            "level": "error",
            "title": "The Internal Database (CouchDB) is not available",
            "keywords": ["platform-monitoring", "database"],
            "display": {
                "type": "toast"
            }
        }]
    },
    "idp": {
        …
        "notifications": [{
            "origin": "server",
            "type": "alert",
            "id": "platform-monitoring-roles-notification-idp",
            "roles": ["Platform Administrator"],
            "sendToRoles": ["Platform Administrator"],
            "level": "error",
            "title": "The Identity Provider is not available",
            "keywords": ["platform-monitoring", "Identity Provider"],
            "display": {
                "type": "toast"
            }
        }]
    },
    "notificationServer": {
        …
        "notifications": [{
            "origin": "server",
            "type": "alert",
            "id": "platform-monitoring-roles-notification-server",
            "roles": ["Platform Administrator"],
            "sendToRoles": ["Platform Administrator"],
            "level": "error",
            "title": "The Notification Server is not available",
            "keywords": ["platform-monitoring", "Notification Server"],
            "display": {
                "type": "toast"
            }
        }]
    },
…
}
```

Where

Notifications is the list of all the notifications that will be send for the given component. One error notification is send to all platform administrators (toaster message) for all resources of UOC except the installed plugins.

Chapter 12
Identity Provider

This chapter explore the installation and usage of an Identity Provider for the Unified OSS Console. It presents key information to start with and recommended configuration and tools for identity Provider we successfully integrated with Unified Console.

IMPORTANT: SAML Authentication is the only supported authentication mode for production. It requires an additional identity provider in charge of managing users and roles.

IMPORTANT: **PicketLink** or **Keycloak** is not part of Unified Console and is an example of possible Identity Provider that have been tested with Unified Console. All the configuration below are given for understanding and example only. Each identity providers are different and provide different level of features.

12.1 PicketLink

PicketLink is an open source project for simplified security and identity management for Java Applications. It provides SSO using SAML v1.1 and v2.0. to build robust SAML enabled applications.

Official web site: http://picketlink.org

12.1.1 Features

- PicketLink provides SSO using SAML v1.1 and v2.0.
- Parsers and Object Model available to build robust SAML enabled applications.
- Build model from various data sources such as Databases, LDAP, File System and mix-n-match!
- Simple API for Users, Roles, Groups and Attributes.
- Application developers have greater control for authentication. You use the IDM as the foundation for your authentication needs.
- PicketLink has a permission model that allows you to have robust access control for your Java Applications.
- Permission implementations include ACL and Drools Rules based implementations.
- API allows custom authorization implementations.
- Easy migration to a fine grained access control model using XACML.
- PicketLink includes login modules for building trusted heterogeneous applications using different application servers including Wildfly Application Server.
- PicketLink allows you to incorporate Social Login into your applications. You can build applications that allow sign in using Facebook, Google and Twitter.
- PicketLink allows JavaEE applications to incorporate robust security. JavaEE constructs are supported.
- PicketLink supports SAML, XACML and WS-Trust.

12.1.2 Prerequisites

These tools are required to be installed and configured properly in order to set up the Identity Provider:

- JDK 7 / OpenJDK 7
- Wildfly 8.1.0Final+
- Apache Ant
- Apache Maven 3.0+

12.1.3 Configure Wildfly

12.1.3.1 Update the PicketLink libraries

The Identity Provider requires PicketLink 2.7.0+. Since Wildfly 8.1.0.Final embed PicketLink 2.5.2 by default, an update is necessary. The simplest way is to use the installer provided by picketlink.org. Stop the Wildfly server if it is running and make sure you have read/write access to your Wildfly server installation.

- Download the installer at: http://downloads.jboss.org/picketlink/2/latest/picketlink-installer-2.7.0.CR1.zip.
- Extract the files from the archive and go into the extracted folder picketlink-installer-2.7.0.CR1. There must be two folders: config and tmp, plus two files: build.xml and installer.properties.
- Open a command-line interface, navigate to this folder and run the command **ant** to launch the installation script.
- Answer **wildfly** to the question 'Which JBoss Application Server are you using?' then enter the path to your Wildfly server installation (WILDFLY_HOME) (ex. */opt/jboss/wildfly-8.1.0.Final*).

Your Wildfly server is now updated with PicketLink 2.7.0.

12.1.3.2 Create the Security Domain

These steps assume you run the server in the standalone mode and use the standalone.xml supplied with the distribution. Before you begin, stop your Wildfly server if it is running and backup your server configuration file in the folder *WILDFL Y_HOME/standalone/configuration/standalone.xml*. You can then replace this file to restore the server to its original configuration if needed.

- Start the Wildfly server.
- Navigate to the root directory of the Identity Provider files with a command-line interface and run the following command, replacing WILDFLY_HOME with the path to your Wildfly server installation and WILDFLY_CONTROLLER with a management endpoint of your server, by default 127.0.0.1:9990.

  Linux:
  WILDFLY_HOME/bin/jboss-cli.sh --connect controller=WILDFLY_CONTROLLER --file=configure-security-domain.cli

  Windows:
  WILDFLY_HOME/bin/jboss-cli.bat --connect controller=WILDFLY_CONTROLLER --file=configure-security-domain.cli

  You should see the following result. The batch executed successfully

```
{
"outcome" => "success",
}
```

- Restart the Wildfly server

12.1.4 Deploy the Identity Provider

- Run a Wildfly server if there is none running yet.
- Navigate to the root directory of the Identity Provider files with a command-line interface.
- Run the command **mvn clean package**. It may take some time.

You should see the following result

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```

- Run the following command, replacing WILDFLY_HOME with the path to your Wildfly server installation and WILDFLY_CONTROLLER with a management endpoint of your server, by default 127.0.0.1:9990.

Linux:

```
WILDFLY_HOME/bin/jboss-cli.sh --connect controller=WILDFLY_CONTROLLER --command=”deploy target/hp-picketlink-federation-saml-idp-basic-wildfly.war --force”
```

Windows:

```
WILDFLY_HOME/bin/jboss-cli.bat --connect controller=WILDFLY_CONTROLLER --command=”deploy target/hp-picketlink-federation-saml-idp-basic-wildfly.war --force”
```

The Identity Provider is now deployed to the Wildfly server and is accessible to the address http://JBOSS.BIND.ADDRESS:JBOSS.HTTP.PORT/idp, by default http://localhost:8080/idp.

12.1.5 Security

These steps describes how to enable optional security features such as signature and encryption of SAML assertions. Basically, it is based on customizations of the Identity Provider configuration file picketlink.xml.

For more information, please visit: https://docs.jboss.org/author/display/PLINK/Identity+Provider+Configuration

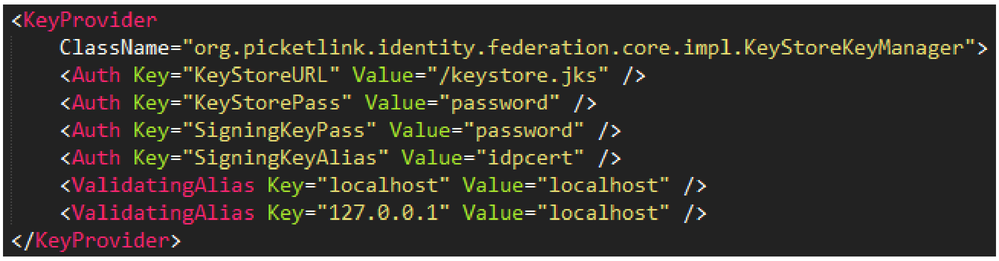
12.1.5.1 Enable Signature Support

- Open the file picketlink.xml in the folder */conf/wildfly/WEB-INF* from the root directory of the Identity Provider files
- In the PicketLinkIDP tag
  - Add the attribute SupportsSignatures="true"
  - Add the attribute CanonicalizationMethod=”http://www.w3.org/2001/10/xml-exc-c14n#”. This attribute is required because the default Canonicalization method (http://www.w3.org/2001/10/xml-exc-c14n#WithComments) is not supported by the UOC V2.0 server.

- Under the Handlers element
  - Add a child handler
    `<Handler class="org.picketlink.identity.federation.web.handlers.saml2. SAML2SignatureGenerationHandler" />`
  - Add a child handler
    `<Handler class="org.picketlink.identity.federation.web.handlers.saml2.SAML2SignatureValidationHandler" />`

Theses handlers enable the processing of signed assertions.

- Under the PicketLinkIDP tag, configure a KeyProvider child element.



```
<KeyProvider
    ClassName="org.picketlink.identity.federation.core.impl.KeyStoreKeyManager">
    <Auth Key="KeyStoreURL" Value="/keystore.jks" />
    <Auth Key="KeyStorePass" Value="password" />
    <Auth Key="SigningKeyPass" Value="password" />
    <Auth Key="SigningKeyAlias" Value="idpcert" />
    <ValidatingAlias Key="localhost" Value="localhost" />
    <ValidatingAlias Key="127.0.0.1" Value="localhost" />
</KeyProvider>
```

Figure 16 - KeyProvider element sample

The KeyProvider element specify some configurations about the Java KeyStore that should be used to sign SAML assertions:

- KeyStoreURL : Location of a Java KeyStore
- KeyStorePass : Password of the KeyStore
- SigningKeyAlias : Alias of the certificate to be used to sign SAML assertions
- SigningKeyPass : Password of the certificate referenced by the SigningKeyAlias
- ValidatingAlias: This element allows to verify the signatures of the SAML assertions. The Key attribute defines the alias of the certificate to should be used.

This name must match one of the trusted domains of the Identity Provider. The Value attribute defines the password of the latter.

12.1.5.2 Enable Encryption Support

- Open the file picketlink.xml in the folder */conf/wildfly/WEB-INF* from the root directory of the Identity Provider files
- In the PicketLinkIDP tag, add the attribute Encrypt="true"
- Under the Handlers element
    - Add a child handler

      <Handler class="org.picketlink.identity.federation.web.handlers.saml2.SAML2EncryptionHandler" />
    - Add a child handler

      <Handler class="org.picketlink.identity.federation.web.handlers.saml2.SAML2SignatureValidationHandler" />

Theses handlers enable the processing of encrypted assertions.

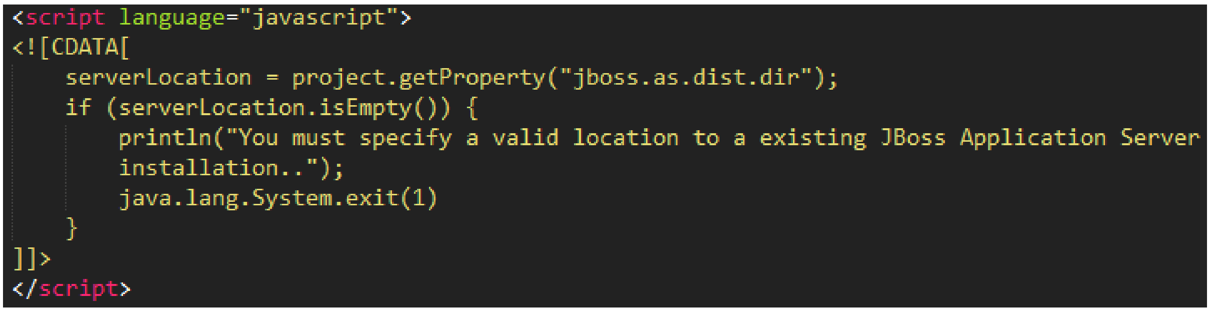
IMPORTANT: Do not use the SAML2EncryptionHandler with the SAML2SignatureGenerationHandler at the same time otherwise SAML messages will be signed several times. In such a case, use only the SAML2EncryptionHandler

- Configure a KeyProvider element.

12.1.5.3 Troubleshooting

Here are some solutions to issues you may encounter during the installation.

1. When I run the PicketLink installer script, I get java.lang.ClassNotFoundException: org.apache.bsf.engines.javascript.JavaScriptEngine.
   ➢ The JavaScript Engine is only included in the JDK6+, you are probably using a JRE or another Java implementation that does include the JavaScript Engine (JDK5 or lower, Oracle JRockit ...). Either install a full JDK 7 to run the installation script properly (recommended) or modify the installation script (build.xml) by removing the following code :



```
<script language="javascript">
<![CDATA[
    serverLocation = project.getProperty("jboss.as.dist.dir");
    if (serverLocation.isEmpty()) {
        println("You must specify a valid location to a existing JBoss Application Server
        installation..");
        java.lang.System.exit(1)
    }
]]>
</script>
```

It suppress the dependency to the JavaScript Engine but also disable the check of the location of your Wildfly server installation during the process. Use at your own risk.

2. When I run command with jboss-cli, I get Error: Could not find or load main class Files\JBoss\wildfly-8.1.0.Final\bin\jboss-cli-logging.properties Press any key to continue.
   ➢ If you use Windows, there is a known issue with jboss-cli.bat usage when Wildfly is installed into a directory with spaces or special characters https://bugzilla.redhat.com/show_bug.cgi?id=1031173.
   This error can be avoided by not installing Wildfly into such a directory or according to https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.2/html/6.2.0_Release_Notes/ar01s07s03.html :
   It can be worked around by editing the jboss-cli.bat file to move line 64 the first " character from the beginning of JAVA_OPTS after the assignment so it looks like the following.

   set JAVA_OPTS="%JAVA_OPTS% -Djboss.modules.system.pkgs=com.sun.java.swing -Dlogging.configuration=file:%JBOSS_HOME%\bin\jboss-cli-logging.properties"

12.2 Keycloak

Integrated SSO and IDM for browser apps and RESTful web services. Built on top of the OAuth 2.0, Open ID Connect, JSON Web Token (JWT) and SAML 2.0 specifications. Keycloak has tight integration with a variety of platforms and has a HTTP security proxy service where we don't have tight integration. Options are to deploy it with an existing app server, as a black-box appliance, or as an Openshift cloud service and/or cartridge.

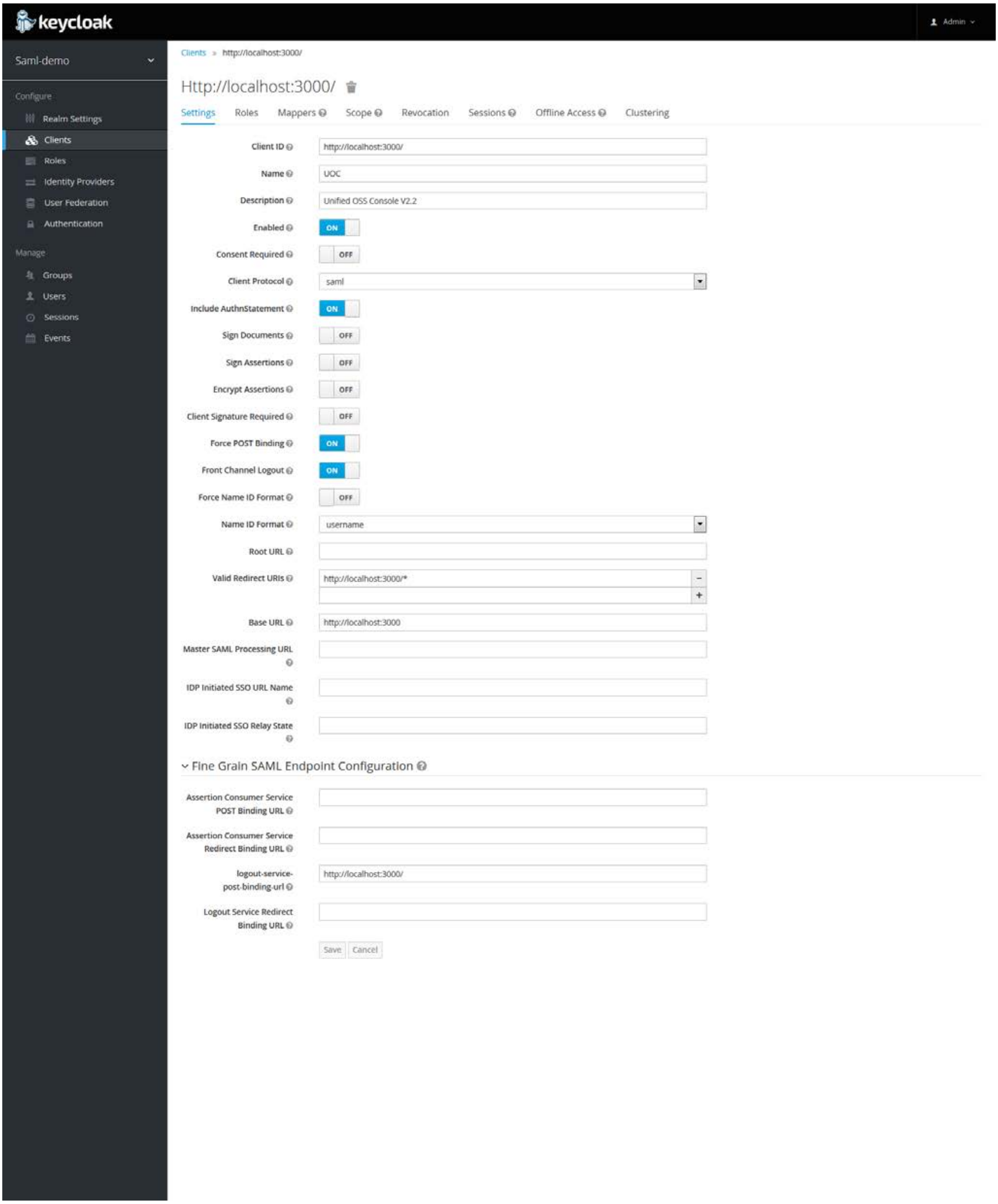
12.2.1 Features

- OpenID Connect and SAML 2.0 SSO and Single Log Out for browser applications
- Social Broker. Enable Google, Facebook, Yahoo, Twitter social login with no code required.
- Identity Broker. Delegate to an external SAML 2.0 or OIDC broker for auth.
- Optional LDAP/Active Directory integration
- Optional User Registration, with optional Recaptcha ability
- Password and TOTP support (via Google Authenticator). Client cert auth coming soon.
- User session management from both admin and user perspective
- Customizable themes for user facing pages: login, grant pages, account management, emails, and admin console all customizable!
- OAuth 2.0 Bearer token auth for REST Services
- Integrated Browser App to REST Service token propagation
- Admin REST API
- CORS Support
- Completely centrally managed user and role mapping metadata. Minimal configuration at the application side
- Admin Console for managing users, roles, role mappings, applications, user sessions, allowed CORS web origins, and OAuth clients.
- Deployable as a WAR, appliance, or an Openshift cloud service (SaaS).
- Supports JBoss AS7, EAP 6.x, Wildfly, Tomcat, Jetty, and Pure Javascript applications. Plans to support Node.js, RAILS, GRAILS, and other non-Java applications.
- HTTP Security Proxy for environments/platforms/languages that don't have a client adapter
- Javascript/HTML 5 adapter for pure Javascript apps
- Session management from admin console
- Claim/assertion mappings. Make your tokens and assertion XML look however you want.
- Revocation policies
- Password policies
- Impersonation. Allow your admins to impersonate a user to debug problems.

Official web site: http://keycloak.jboss.org/

12.2.2 Configuration

Here is an example of configuration screen for keycloak integration



Http://localhost:3000/

Client ID: http://localhost:3000/

Name: UOC

Description: Unified OSS Console V2.2

Enabled: ON

Consent Required: OFF

Client Protocol: saml

Include AuthnStatement: ON

Sign Documents: OFF

Sign Assertions: OFF

Encrypt Assertions: OFF

Client Signature Required: OFF

Force POST Binding: ON

Front Channel Logout: ON

Force Name ID Format: OFF

Name ID Format: username

Root URL:

Valid Redirect URIs: http://localhost:3000/*

Base URL: http://localhost:3000

Master SAML Processing URL:

IDP Initiated SSO URL Name:

IDP Initiated SSO Relay State:

Fine Grain SAML Endpoint Configuration

Assertion Consumer Service POST Binding URL:

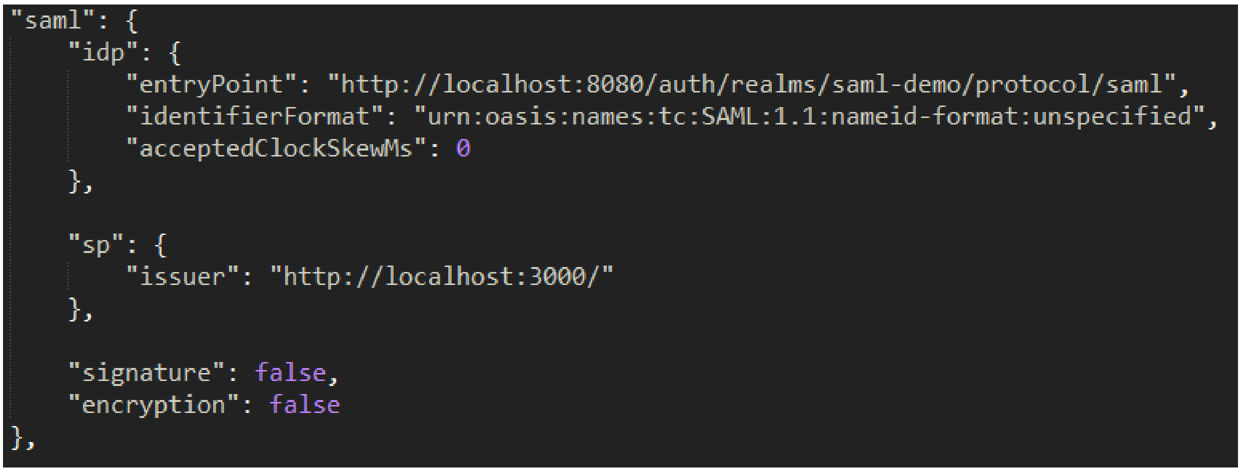
Assertion Consumer Service Redirect Binding URL:

logout-service-post-binding-url: http://localhost:3000/

Logout Service Redirect Binding URL:

Figure 17: Keycloak Configuration Example

The configuration file is stored in <install_dir>/server/public/conf/config.json



```
"saml": {
    "idp": {
        "entryPoint": "http://localhost:8080/auth/realms/saml-demo/protocol/saml",
        "identifierFormat": "urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified",
        "acceptedClockSkewMs": 0
    },

    "sp": {
        "issuer": "http://localhost:3000/"
    },

    "signature": false,
    "encryption": false
},
```

Figure 18: Unfiied Console Configuration Example

Chapter 13
Platform Backup

13.1 Overview

There are 2 types of data to back up on the platform:

- Data stored in the GUI Document Database (Apache CouchDB) like views, workspaces, users, permissions, roles…
- Configuration files updated to defines correctly hostname, port, specific settings (authentication…)

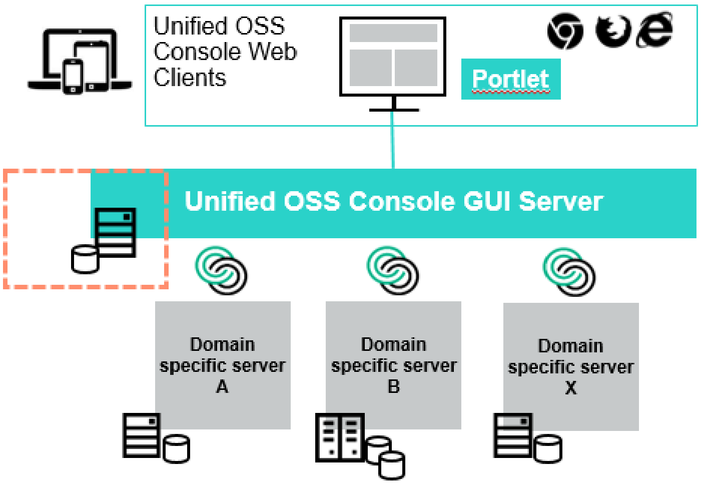


Figure 19: Platform Backup Overview

13.2 UOC Data

All the configuration files and data that the administrator can customize on the UOC platform are saved in the <install_data_dir> (usually /var/opt/uoc2)

After uninstallation, these files are not removed and will not be overridden by a new UOC installation.

It is recommended to integrate these data directories in a backup process.

13.3 GUI Database (Apache CouchDB)

This part of the document assumes that the installation of Apache CouchDB was made in the default directory /opt/couchdb. In case another directory was used, be careful to adapt the following commands.

The following two parts describe how to backup or replicate all UOC CouchDB databases or one or several of these databases. UOC database names are:

- categories
- permissions
- roles
- users (local authentication only)
- views
- workspaces
- launches
- launch-categories

You have two options to save/restore a couchdb database. You can save the data directory or you can replicate data from the database to another instance of couchdb (replication). Replication is the recommended method because it grants you to always have a working couchdb database. You will not need any restore operations.

13.3.1 Backup Apache CouchDB database files

First, stop Apache CouchDB and applications that could use it like UOC.

Then, save the directory /opt/couchdb/var/lib/couchdb (replace DIRECTORY by your backup destination directory)

```
tar –cvf DIRECTORY/couchdb.tar /opt/couchdb/var/lib/couchdb
```

If you want to backup only specific database files, you can save the files named by the database’s name and with the extension .couch.

Restart CouchDB.

13.3.2 Restore Apache CouchDB database files

This part assumes you generated an archive by following the steps described in the previous section (see 13.3.1 Backup Apache CouchDB database files)

If the CouchDB database targeted for the backup recovery is new, be sure to have correctly and totally installed it and initialized it before.

Stop Apache CouchDB and applications that could use it like UOC.

Use your backup archive to restore Apache CouchDB database files (replace DIRECTORY by your backup archive directory)

```
tar –xvf DIRECTORY/couchdb.tar /opt/couchdb/var/lib
```

Restart Apache CouchDB.

13.3.3 Replicate CouchDB

Replication synchronizes two copies of the same database, allowing users to have low latency access data no matter where they are. These database can live on the same server or on two different servers—CouchDB doesn't make a distinction. If you change one copy of the database, replication will send these changes to the other copy.

Please refer to the high availability details to run provided script to help the CouchDB Replication.

13.3.3.1 Simple Replication with the Admin Interface

You can run replication from your web browser using Futon, CouchDB's built-in administration interface.

Prerequisites:

- CouchDB has to be started
- Access to http://SERVER_IP:5984/_utils has to be enabled.

If it is not the case, stop CouchDB and modify this file:

```
nano /opt/couchdb/var/config/couchdb/local.ini
```

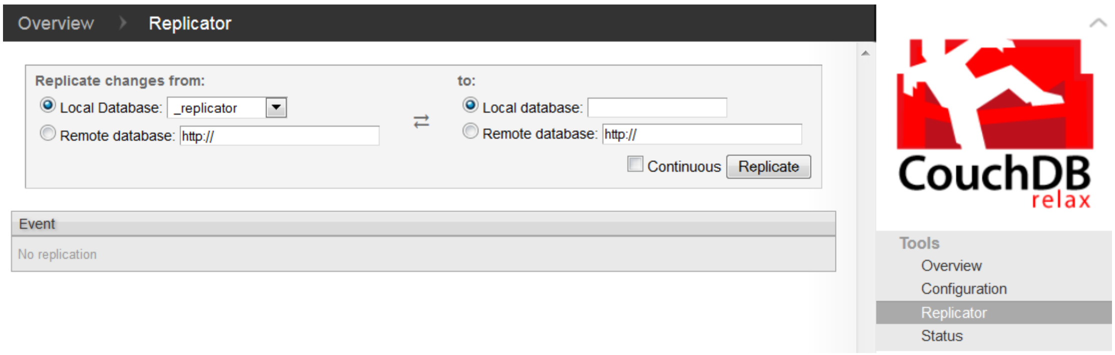
Add the following two lines under [httpd]:

```
[httpd]
port = 5984
bind_address = 0.0.0.0
```

Start CouchDB.

Open your browser to http://SERVER_IP:5984/_utils. On the right-hand side, you will see a list of things to visit in Futon. Click on "Replication."

Futon will show you an interface to start replication. You can specify a source and a target by either picking a database from the list of local databases or filling in the URL of a remote database.



Replicate changes from:
● Local Database: _replicator
○ Remote database: http://

to:
● Local database:
○ Remote database: http://

□ Continuous

Event
No replication

Figure 20: CouchDB Futon Replication Interface

Click on the Replicate button, wait a bit for the replication to proceed, and have a look at the lower half of the screen where CouchDB gives you some statistics about the replication run or, if an error occurred, an explanatory message.

For additional information about CouchDB replication possibilities, please visit http://guide.couchdb.org/editions/1/en/replication.html.

Chapter 14
Load Balancer

This chapter explore the installation and usage of a load balancer for the Unified OSS Console. It presents key information to start with and recommended configuration and tools.

14.1 Introduction

A load balancer is the application that distributes workloads of client requests across a set of running server instances to maximize performance and optimize resources usage.

For the UOC platform, a load balancer is needed for the following reasons:

- Scalability: The main role of a load balancer is to spread the incoming client requests traffic through a set of UOC server instances (across local or remote servers). This allows to easily adapt the infrastructure to support the workload while being cost most effective.
- Health-check: In order to efficiently perform load balancing between the instances of UOC server, the load balancer keeps track current status of the UOC server application. Health checking allows a load balancing solution to determine whether a server/instance is “available” following these three steps:
    - The first check is an ICMP ping: it's used to determine whether the server is available or not.
    - A three-way TCP handshake is the next “step”: this will tell the load balancing solution, whether an application is capable of accepting connections (or not).
    - The highest check level is a simple HTTP GET request: it retrieves actual data and ensure it is valid in order to consider an application “available”.
- Failover: In case of an application crash or a hardware fails, the load balancer is able to redirect the request to an available running instance of the UOC server. Load balancers allow the user to set a default address to display the status page, they also enable him to configure the number successive failed connections needed to flag a UOC server instance, as being "Down" and how much time the load balancer should wait before sending a new status check of the flagged UOC server instance.

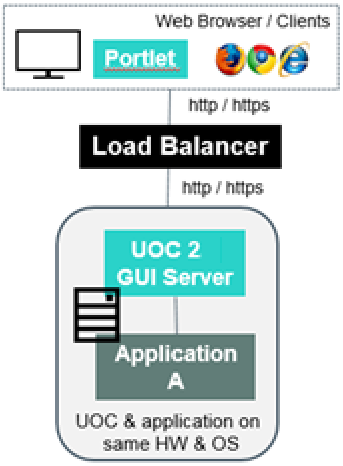


Figure 21: UOC architecture with load balancing overview

14.2 Deployment Examples

There are basically multiple ways to support load balancing.

- It is possible to run multiples instances of UOC server on the same server (medium deployment)
- It is also possible to run multiples instances of UOC on multiple server (large deployment)

14.2.1 Multiple UOC server instances in one machine

In this deployment, several instances of the UOC server are running on one machine so as to use all its cores (each UOC server uses 1 core). The Data server uses a separate machine to handle all the requests coming from all UOC users.

One machine with:

- Load balancer (HAProxy for example)
- UOC Server
- Apache CouchDB (GUI Database)

One or more machines with:

- Domain or Data Server (ex: OSS Analytic Server)

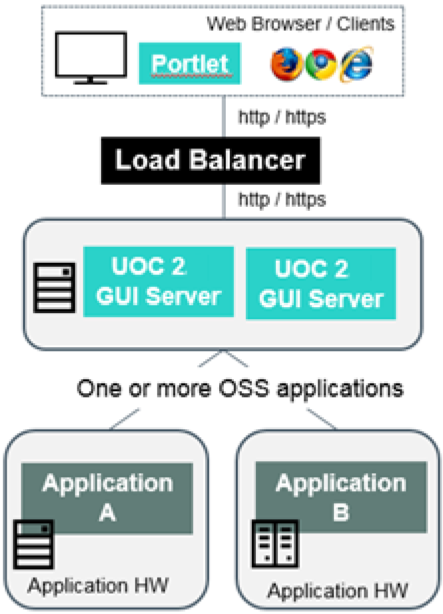


Figure 22: Load Balancer - Multiples UOC Instances running on one server

14.2.2 Multiple UOC server instances on multiple machines

In this deployment, several machines are used to run several instances of the UOC servers to support a large volume of data and users.

One machine with:

- Load balancer (HAProxy for example)

One machine with:

- Apache CouchDB

One or more machines with:

- Data server (Example: OSSA server).

Several machines with UOC Servers. Each machine can run multiple instances of UOC Server (one per core).

Important: The port number for each instance must be unique per server machine.

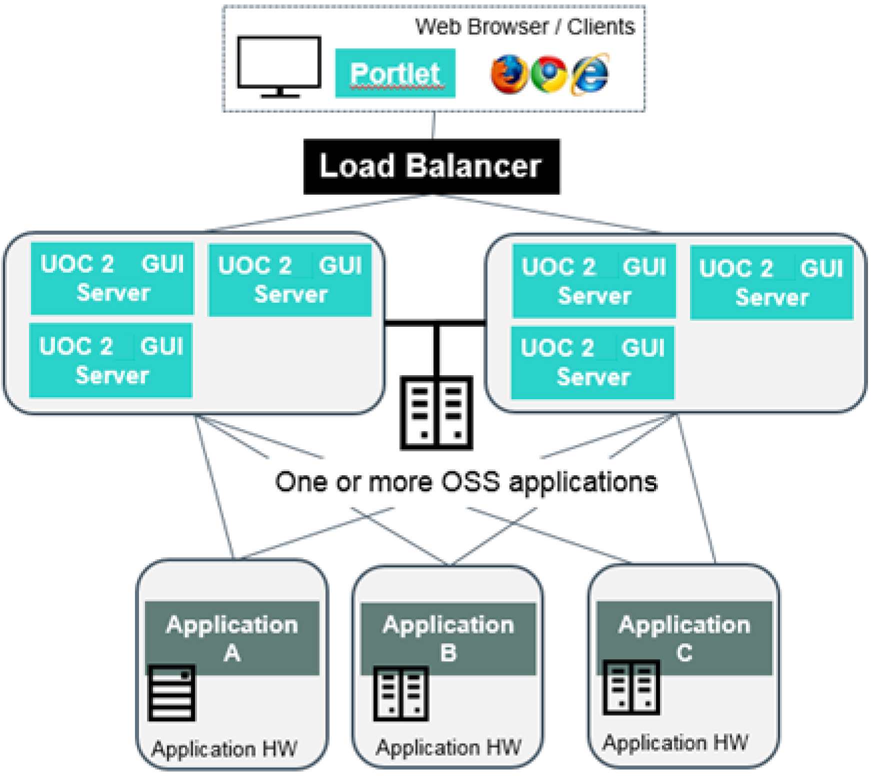


Figure 23: Load Balancer – Multiples UOC Server instances running on multples machines

14.3 Recommended Applications

For the UOC platform, the recommended load balancer application is HAProxy or NGINX.

HAProxy is the Reliable, High Performance TCP/HTTP Load Balancer. You can find more details on their official web site: http://www.haproxy.org/

HAProxy is a free standard open source software load balancer that functions as a fast proxy server and provides high availability for TCP and HTTP based applications. The application is based on a single-process (event-driven) model and is supported by the Target OS (Redhat 6.5) and many other Linux distributions. The documentation of this application is spot on and is well maintained.

HAProxy is widely adopted by the web community because of the excellent performance gain and the advanced customized features (load balancing algorithm, high-availability management …) that it provides.

NOTE: The version described in this document is the version 1.4 which is available in the EPEL repository

NGINX (pronounced engine-x) is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. It's known for its high performance, stability, rich feature set, simple configuration and low resource consumption. See https://www.nginx.com

14.4 HAProxy

The root credentials are needed for installing the load balancer application's packages and to configure its settings.

14.4.1 Installation

The load balancer HAProxy is available within the EPEL repository.

To install, this command must be executed:

```
$ yum install haproxy
```

14.4.2 Validation

To validate the installation of HAProxy, the following command must display the following output:

```
$ rpm -qa haproxy
haproxy-1.4.24-2.el6.x86_64
```

14.4.3 Uninstallation

To uninstall HAProxy, this command must be executed :

```
$ rpm -e haproxy-1.4.24-2.el6.x86_64
```

14.4.4 Red Hat Linux Firewall Settings

Netfilter is a host-based firewall for Linux operating systems. It is included as part of the Linux distribution and it is activated by default on RHEL6. This firewall is controlled by the program called iptables. Netfilter filtering takes place at the kernel level, before a program can even process the data from the network packet.

Therefore, when iptables is up and filtering packets, its settings should be modified in order to let the UOC server work properly. In particular, incoming HTTP(s) request on the UOC server port should be allowed.

Please refer to your system admin manual for configuring the firewall if necessary.

```
$ man iptables
```

14.4.5 Administration

14.4.5.1 Start the load balancer

To start the load balancer, run the following command with the user uoc.

```
$ service haproxy start
```

14.4.5.2 Stop the load balancer

To start the load balancer, run the following command with the user uoc.

```
$ service haproxy stop
```

14.4.5.3 Check the load balancer configuration

The following command allows the user to check if the configuration file located in /etc/haproxy is valid.

```
$ service haproxy check
```

14.4.5.4 Setup Automatic start

To start the load balancer automatically when the machine boots, following command must be executed.

```
$ chkconfig haproxy on
```

14.4.6 Configuration

The configuration file of HAProxy is located in /etc/haproxy/haproxy.cfg

The complete description of each configuration parameter is available in the documentation page of HAProxy.

See https://cbonte.github.io/haproxy-dconv/configuration-1.4.html

Here is a sample configuration file of HAProxy that may need some adjustment depending on your system requirements:

```
#-------------------------------------------------------
# Global settings
#-------------------------------------------------------
global
    log         127.0.0.1 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

defaults
    mode                    http
    log                     global
    option                  httplog
    option                  dontlognull
    option http-server-close
    option forwardfor       except 16.17.100.86/8
    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
```

This first section of the configuration file contains the global configuration attributes (log options, user profile, requests timouts, connection handling, …) . These configuration are applied on all the backend clusters.

```
#---------------------------------------------------------
# main frontend which proxys to the backends
#---------------------------------------------------------
frontend  <server_name>
bind *:80
reqadd X-Forwarded-Proto:\ http
default_backend    <server_name>
```

This section defines the <url:port> that is bound to the load balancer application. It's the proxy entry point as viewed by the clients.

```
#--------------------------------------------------------
# Least connected  balancing between the various
# backends
#--------------------------------------------------------
backend <server_name>
mode http

stats enable
stats hide-version
stats uri /stats
stats realm Haproxy\ Statistics
stats auth haproxy:redhat

balance leastconn

option httpchk
option  httpclose
option forwardfor

server  <server_name>:<port_number> check inter 20 fall 5 rise 2 backup
server  <server_name>:<port_number> check inter 20 fall 5 rise 2
server  <server_name>:<port_number> check inter 20 fall 5 rise 2
server  <server_name>:<port_number> check inter 20 fall 5 rise 2
server  <server_name>:<port_number> check inter 20 fall 5 rise 2
```

HAProxy can recognize and configure multiple clusters, each cluster is referenced by the attribute backend. The list of servers that a cluster contains is appended to the attribute option forwardfor.

Each server is identified by its url:port pair and can have the following attributes :

Attribut	Value	Description
backup		When "backup" is present on a server line, the server is only used in load balancing when all other non-backup servers are unavailable. Requests coming with a persistence cookie referencing the server will always be served though. By default, only the first operational backup server is used.
check		This option enables health checks on the server. By default, a server is always considered available. If "check" is set, the server will receive periodic health checks to ensure that it is really able to serve requests. The default address and port to send the tests to are those of the server, and the default source is the same as the one defined in the backend. The request method is defined in the backend using the "httpchk", "smtpchk", "mysql-check" and "ssl-hello-chk" options. Please refer to those options and parameters in HAProxy documentation for more information.
inter	20 (seconds)	Define the health-check time interval.
fall	5	It states that a server will be considered as dead after 5 consecutive unsuccessful health checks. This value defaults to 3 if unspecified.
rise	5	It states that a server will be considered as operational after 5 consecutive successful health checks. This value defaults to 2 if unspecified.

Figure 24: HAProxy - High Availability Attributes.

Also, for each backend cluster, it's possible to define a load balancing algorithm with the attribute balance. The algorithm specifies the politic of the client requests redirection to the UOC server instances.

HAProxy defines several load balancing algorithms, but since there are no HTTP session's persistence requirements, this document will present only the two pertinent algorithms.

Algorithm	Description
roundrobin	Each server is used in turns, according to their weights. This is the smoothest and fairest algorithm when the server's processing time remains equally distributed. This algorithm is dynamic, which means that server weights may be adjusted on the fly for slow starts for instance. It is limited by Design to 4095 active servers per backend. Note that in some large farms, when a server becomes up after having been down for a very short time, it may sometimes take a few hundred requests for it to be re-integrated into the farm and start receiving traffic. This is normal, though very rare.
leastconn	The server with the lowest number of connections receives the connection. Round-robin is performed within groups of servers of the same load to ensure that all servers will be used. Use of this algorithm is recommended where very long sessions are Expected, such as LDAP, SQL, TSE, etc... But is not very well suited for protocols using short sessions such as HTTP. This algorithm is dynamic, which means that server weights may be adjusted on the fly for slow starts for instance.

Figure 25: Load balancing algorithms.

NOTE: the exhaustive algorithm's list supported by HAProxy is presented in their web site https://cbonte.github.io/haproxy-dconv/configuration-1.4.html#4.2-balance

14.4.7 Logging

HAProxy use the system package Syslog to track and archive the Client / Server request exchanges and the state of the UOC server instances. In the configuration file of HAProxy located in **/etc/haproxy/haproxy.cfg**.

It is possible to define the address of the machine that backup the log files. By default it is:

```
global
    log          127.0.0.1 local2
```

In this case, it means that HAProxy send the login details to the Syslog module that is located in the same machine.

So, the Syslog module must be configured to listen to this information. The configuration is done by creating the file /etc/rsyslog.d/haproxy.cfg and adding this content:

```
$ModLoad imudp
$UDPServerRun 514
$template Haproxy,"%msg%\n"
local2.=info -/var/log/haproxy.log;Haproxy
local2.notice -/var/log/haproxy-status.log;Haproxy
### keep logs in localhost ##
local2.* ~
```

In this configuration, it's possible to observe the following:

- Syslog is enabled to listen for the HAProxy stats and exchanges on the UDP port 514 for all IP addresses. Optionally, it's possible to limit the IP address to 127.0.0.1 by adding:

```
$UDPServerAddress 127.0.0.1
```

- Separate log files are defined and regrouped by log level (info, notice, warning ...). The exhaustive list of log levels is presented below :

Value	Severity	Keyword	Description
0	Emergency	emerg	This level should not be used by applications.
1	Alert	alert	Should be corrected immediately
2	Critical	crit	A failure in the system's primary application.
3	Error	err	An application has exceeded its file storage limit and attempts to write are failing.
4	Warning	warning	May indicate that an error will occur if action is not taken.
5	Notice	notice	Events that are unusual, but not error conditions.
6	Informational	info	Normal operational messages that require no action.
7	Debugging	debug	Information useful to developers for debugging the application.

Figure 26: Log levels.

Now that the Syslog module is connected to HAProxy, it will be possible to format the log entries by creating the file /etc/rsyslog.d/haproxy.conf file containing:

```
local2.*    /var/log/haproxy.log
```

To make these modification effective, a restart of Syslog service is required:

```
# service rsyslog restart
Shutting down system logger:                      [  OK  ]
Starting system logger:                           [  OK  ]
# ls -l /var/log | grep haproxy
-rw-------. 1 root root 131 17 jul. 10:43 haproxy.log
-rw-------. 1 root root 106 17 jul. 10:42 haproxy-status.log.log
```

14.5 NGINX

NGINX (pronounced engine-x) is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. It's known for its high performance, stability, rich feature set, simple configuration and low resource consumption. See https://www.nginx.com

14.5.1 Features

A complete set of web serving, proxying, acceleration, and load-balancing capabilities for both HTTP-based and TCP-based services

Protocols and Performance

- HTTP/1.1, HTTP/2, HTTPS, WebSocket
- IMAP, POP3, SMTP with HTTPS and external HTTP-based authentication
- IPv4 and IPv6
- 1 million concurrent connections
- 10,000+ virtual servers multi-tenancy
- Connection multiplexing pools for low-latency communications

Load Balancing

- Full Layer 7 reverse proxy
- Load balancing of HTTP traffic, including HTTP, HTTPS, FastCGI, memcached, SCGI, uWSGI
- Load balancing of TCP traffic
- Load balancing of UDP traffic
- URL/URI content-based request routing
- Reverse proxy and load balancer with choice of algorithms (Round Robin, Hash, IP Hash, Least Connections, Least Time)
- Session persistence with cookie-insert, session-learn, and defined-route methods
- Session draining for easy maintenance
- Health monitoring of backend applications with synthetic transactions and slow start

High Availability

- Active-standby clusters using VRRP (keepalived)
- Live binary upgrades to eliminate downtime
- Graceful restart with non-stop request processing

Security

- Bandwidth, connection, and request policing for HTTP and TCP services
- Protocol isolation and request filtering
- Header scrubbing and manipulation
- IP-based access control lists (ACLs)
- Proxying requests with NTLM authentication

SSL/TLS Processing

- SSL, SNI, TLSv1.1, and TLSv1.2
- Support for RSA, DSA, ECC, and Perfect Forward Secrecy key exchange
- Client and server-side certificate validation for client-side and upstream-side connections
- OCSP stapling

Logging, Monitoring, and Configuration

- On-the-fly reconfiguration of upstream server pools, with changes optionally persistent across restarts and configuration reloads
- Live activity monitoring
- GeoIP configuration decisions
- Logging of HTTP transactions locally or to syslog

Recommended Hardware

NGINX Plus was designed and optimized for use on generic server hardware. For an edge server capable of serving 3 to 6 Gbps of live traffic and 20,000 to 50,000 requests per second, we commonly recommend the following:

- 2 modern x86_64 CPUs with 4 to 8 cores per CPU
- 16 to 32 GB RAM
- 6 x 250-GB SSD drives (if required for cache and storage)
- 10-GbE Intel networking card

14.5.2 Install NGINX

Follow the instructions for your operating system: https://www.nginx.com/resources/wiki/start/topics/tutorials/install/.

By default, NGINX is installed in the folder /etc/nginx.

Basic NGINX commands:

```
# service nginx start                    //Start NGINX
# service nginx stop                     //Stop NGINX
# service nginx status             //Check NGINX status
# service nginx reload             //Reload NGINX configuration
```

If this is not already the case, we need to create in the folder /etc/nginx the folder `sites-available` where all server blocks (i.e. virtual hosts configurations) will be stored in and the folder `sites-enabled` that will hold active server blocks (symbolic links to server blocks stored in the first folder), as well as modify the NGINX configuration to take the folder `sites-enabled` into account.

```
# mkdir sites-available
# mkdir sites-enabled
```

Then edit the file nginx.conf and add the following line at the end of the http block:

```
include /etc/nginx/sites-enabled/*.conf;
```

Example:

```
http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local]
"$request" '
                      '$status $body_bytes_sent "$http_referer"
                      '"$http_user_agent" "$http_x_forwarded_for"';
       access_log  /var/log/nginx/access.log  main;

       sendfile        on;
       #tcp_nopush     on;
       keepalive_timeout  65;
       #gzip  on;

       include /etc/nginx/conf.d/*.conf;
       include /etc/nginx/sites-enabled/*;
}
```

Chapter 15
Node Process Manager Tools

This chapter explore the installation and usage of an external tools that have been tested and can ease deployment and management of UOC servers. These tools are not included in the default kit but it details how you can leverage these tools in your solution and give you concrete examples of usage.

15.1 Forever JS

Forever JS is a very simple CLI tool for ensuring that a given script runs continuously (i.e. forever). You can check their official website for detailed information: https://github.com/foreverjs/forever

It monitors the script specified in command line and rerun the script if the NodeJS is down (i.e. UOC)

15.1.1 Installation

The latest tool version is installable via NPM command line:

```
$ npm install forever -g
```

15.1.2 Administration

You can use forever to run scripts continuously this way:

```
$ forever start server.js
```

And stop them with the stop command:

```
$ forever stop server.js
```

You can see all the available option using:

```
$ forever -h
```

15.2 Process Manager 2 (pm2)

PM2 is an advanced production process manager for Node JS. It has built in monitoring, clustering mode, hot reload, log management, deployment workflow and much more.

You can check their official website for detailed information: http://pm2.keymetrics.io/

15.2.1 Features

PM2 provides a complete feature set for production environment:

- Behavior configuration
- Source map support
- PaaS Compatible
- Watch & Reload
- Log management
- Monitoring
- Module System
- Max memory reload
- Cluster Mode
- Hot reload
- Development workflow
- Startup Scripts
- Auto completion
- Deployment workflow
- Keymetrics monitoring
- API

Once PM2 is started, it will automatically create these folders:

- $HOME/.pm2 will contain all PM2 related files
- $HOME/.pm2/logs will contain all applications logs
- $HOME/.pm2/pids will contain all applications pids
- $HOME/.pm2/pm2.log PM2 logs
- $HOME/.pm2/pm2.pid PM2 pid
- $HOME/.pm2/rpc.sock Socket file for remote commands
- $HOME/.pm2/pub.sock Socket file for publishable events
- $HOME/.pm2/conf.js PM2 Configuration

Here are some commands:

```
# Fork mode
$ pm2 start app.js --name my-api # Name process


# Cluster mode
$ pm2 start app.js -i 0        # Will start maximum processes with LB
depending on available CPUs
```

```
$ pm2 start app.js -i max      # Same as above, but deprecated yet.

# Listing
$ pm2 list               # Display all processes status
$ pm2 jlist              # Print process list in raw JSON
$ pm2 prettylist         # Print process list in beautified JSON

$ pm2 describe 0         # Display all informations about a specific
process

$ pm2 monit              # Monitor all processes

# Logs
$ pm2 logs [--raw]       # Display all processes logs in streaming
$ pm2 flush              # Empty all log file
$ pm2 reloadLogs         # Reload all logs

# Actions
$ pm2 stop all           # Stop all processes
$ pm2 restart all        # Restart all processes

$ pm2 reload all         # Will 0s downtime reload (for NETWORKED apps)
$ pm2 gracefulReload all # Send exit message then reload (for networked
apps)

$ pm2 stop 0             # Stop specific process id
$ pm2 restart 0          # Restart specific process id

$ pm2 delete 0           # Will remove process from pm2 list
$ pm2 delete all         # Will remove all processes from pm2 list

# Misc
$ pm2 reset <process>    # Reset meta data (restarted time...)
$ pm2 updatePM2          # Update in memory pm2
$ pm2 ping               # Ensure pm2 daemon has been launched
$ pm2 sendSignal SIGUSR2 my-app # Send system signal to script
$ pm2 start app.js --no-daemon
$ pm2 start app.js --no-vizion
$ pm2 start app.js --no-autorestart
```

PM2 is a process manager. It manages your applications states, so you can start, stop, restart and *delete* processes and also list all running processes.

Complete options can be displayed using

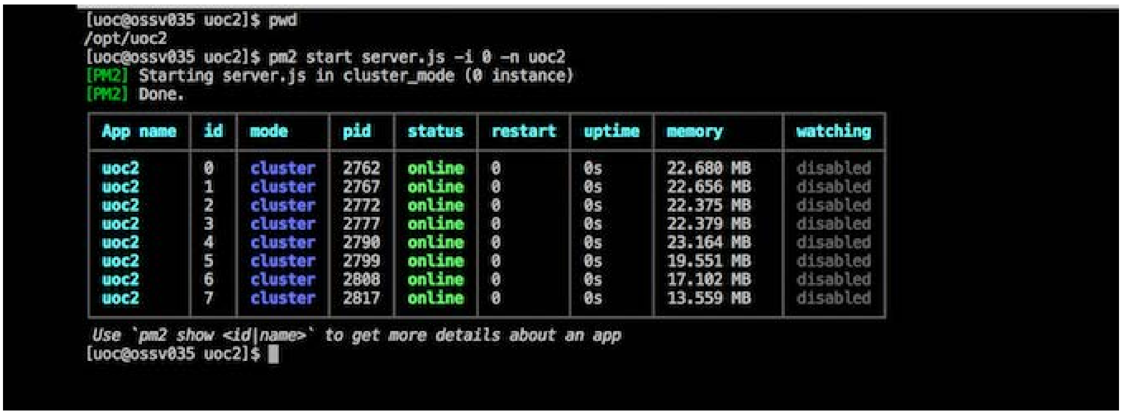
```
$ pm2 -h
```

15.2.2 Installation

The latest PM2 stable version is installable via NPM command line:

```
$ npm install pm2 -g
```

You can start uoc in command line:



```
[uoc@ossv035 uoc2]$ pwd
/opt/uoc2
[uoc@ossv035 uoc2]$ pm2 start server.js -i 0 -n uoc2
[PM2] Starting server.js in cluster_mode (0 instance)
[PM2] Done.
┌──────────┬────┬─────────┬──────┬────────┬─────────┬────────┬─────────────┬──────────┐
│ App name │ id │ mode    │ pid  │ status │ restart │ uptime │ memory      │ watching │
├──────────┼────┼─────────┼──────┼────────┼─────────┼────────┼─────────────┼──────────┤
│ uoc2     │ 0  │ cluster │ 2762 │ online │ 0       │ 0s     │ 22.680 MB   │ disabled │
│ uoc2     │ 1  │ cluster │ 2767 │ online │ 0       │ 0s     │ 22.656 MB   │ disabled │
│ uoc2     │ 2  │ cluster │ 2772 │ online │ 0       │ 0s     │ 22.375 MB   │ disabled │
│ uoc2     │ 3  │ cluster │ 2777 │ online │ 0       │ 0s     │ 22.379 MB   │ disabled │
│ uoc2     │ 4  │ cluster │ 2790 │ online │ 0       │ 0s     │ 23.164 MB   │ disabled │
│ uoc2     │ 5  │ cluster │ 2799 │ online │ 0       │ 0s     │ 19.551 MB   │ disabled │
│ uoc2     │ 6  │ cluster │ 2808 │ online │ 0       │ 0s     │ 17.102 MB   │ disabled │
│ uoc2     │ 7  │ cluster │ 2817 │ online │ 0       │ 0s     │ 13.559 MB   │ disabled │
└──────────┴────┴─────────┴──────┴────────┴─────────┴────────┴─────────────┴──────────┘
 Use `pm2 show <id|name>` to get more details about an app
[uoc@ossv035 uoc2]$ █
```

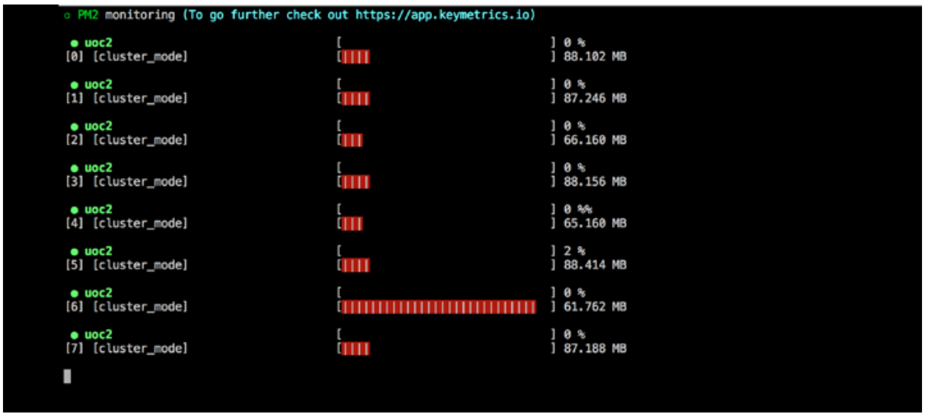
Figure 27: Example of uoc start in cluster mode with PM2

NOTE: it is recommended to cusotmize the startup script and use a PM2 JSON configuration file to ease the stop/start. Please check the PM2 web site http://pm2.keymetrics.io/docs/usage/application-declaration/

15.2.3 Monitoring

PM2 gives you a simple way to monitor the resource usage of your application. You can monitor memory and cpu very easily, straight from your terminal.

```
$ pm2 monit
```



```
○ PM2 monitoring (To go further check out https://app.keymetrics.io)

 ● uoc2                                   [                              ] 0 %
[0] [cluster_mode]                        [||||                          ] 88.102 MB

 ● uoc2                                   [                              ] 0 %
[1] [cluster_mode]                        [||||                          ] 87.246 MB

 ● uoc2                                   [                              ] 0 %
[2] [cluster_mode]                        [|||                           ] 66.160 MB

 ● uoc2                                   [                              ] 0 %
[3] [cluster_mode]                        [||||                          ] 88.156 MB

 ● uoc2                                   [                              ] 0 %%
[4] [cluster_mode]                        [|||                           ] 65.160 MB

 ● uoc2                                   [                              ] 2 %
[5] [cluster_mode]                        [||||                          ] 88.414 MB

 ● uoc2                                   [                              ] 0 %
[6] [cluster_mode]                        [||||||||||||||||||||||||||||| ] 61.762 MB

 ● uoc2                                   [                              ] 0 %
[7] [cluster_mode]                        [||||                          ] 87.188 MB

█
```

Figure 28: Monitoring UOC with pm2 example

Chapter 16
High Availability

This chapter explore the setup and options to manage the high availability (HA) for the Unified Console. It presents key information to start with and recommended configuration and tools.

16.1 Overview

"A high-availability (HA) solution masks the effects of a hardware or software failure and maintains the availability of applications and the integrity of data so that users perceive no downtime"

For UOC, it means that all critical components of UOC must be highly available:

- UOC GUI server (Node.js server)
- GUI database (Apache CouchDB)
- Identity provider (external to UOC)
- Notification Server (Redis - optional)
- Integrated applications via their plugins

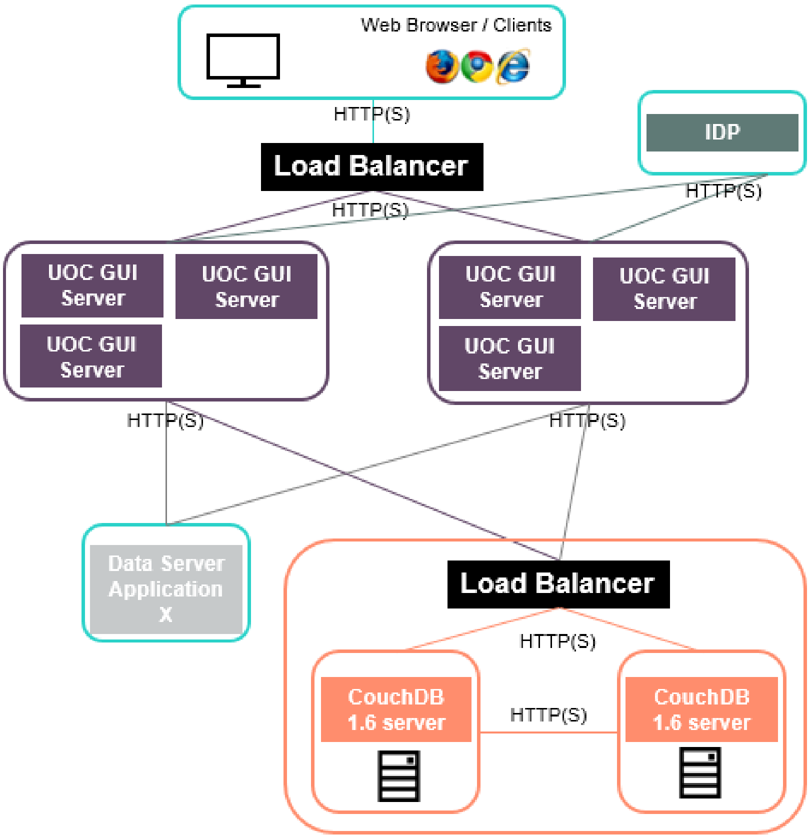


Figure 29: High Availability in the context of Unified Console

In HA mode, Unified OSS Console servers are deployed in **active/active** mode (no cold stand-by)

CouchDB can be deployed in 2 modes: Primary primary replication or Primary replica replication

16.2 UOC Server

Multiple instances on several machines will be required. A load balancer on top of all UOC Servers will distribute the overload and manage the availability to find an active one.

Please refer to the dedicated Chapter 14 Load Balancer to get all the details and recommendations.

16.2.1 Nginx configuration

Example of configuration for NGINX Load Balancer with the following setting:

- NGINX server listens on port 3000
- Round robin load balancing method (default)
- Session affinity using IP hash
- Proxy_headers to support socket.io connections consistency

```
[root@ossv085 sites-enabled]# more /etc/nginx/sites-enabled/uoc
server {
    listen      3000;
    server_name localhost;

    location / {
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_http_version 1.1;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $host;
        proxy_pass http://uocs;
    }
}

upstream uocs {
        ip_hash;
        server          ossv159.gre.hpecorp.net:3050;
        server          ossv159.gre.hpecorp.net:3051;
        server          ossv159.gre.hpecorp.net:3052;
}
```

16.2.2 Platform Monitoring REST API

16.2.2.1 Get Platform Health Status

The UOC server provides a very simple REST API to indicate his health status. This API returns a simple response 'OK' that can be easily be parsed by a load balancer or any external tools to manage the high availability.

Description	HTTP Verb	Rest API
Return the health status	GET	/V1.0/monitoring/server/check

16.2.2.2 Get Plugin Health Status

Plugins can provide a REST API to indicate their health status and also be able to customize this api to check all the critical components linked to the plugin (database, external data servers …).

This API returns a simple response 'OK' that can be easy to parse for a load balancer or any external tools to manage the high availability at plugin level.

Description	HTTP Verb	Rest API
Return the health status 'OK'	GET	/V1.0/domains/*<domainId>*/check

16.3 Identity Provider

Multiple instances on several machines will be required. A load balancer on top of all IDP Servers will distribute the overload and manage the availability to find an active one.

Please refer to the dedicated Chapter 14 Load Balancer to get all the details and recommendations.

16.4 Apache CouchDB Database

The GUI database CouchDB stores all the information needed for the UI layer only. It is all the GUI definitions (workspaces, categories, views …), the launch definitions (external applications to launch) and the RBAC definitions (roles and permissions).

IMPORTANT: This server is mandatory to be able to start and run UOC

Useful reference for detailed information:

- CouchDB, the definitive guide - http://guide.couchdb.org/
- Apache CouchDB documentation - http://docs.couchdb.org/
- CouchDB Wiki - https://wiki.apache.org/couchdb/

To ensure the high availability of the CouchDB database, at least 2 servers are required, synchronizing their data using replication. It can be done in two main ways:

- Primary primary replication: All CouchDB servers can handle any query from users. They are all active instances and propagate changes at one another. (Recommended solution)
- Primary replica replication: There is one CouchDB server that is used as reference for updates and propagates changes to other CouchDB servers. The first instance is active whereas the others are passive. Passive instances are only read-only for users.

  When an availability issue is detected with the primary server, a backup server (can be a former passive instance) becomes the new primary and replaces the faulty primary server. Continuous replication insures the contents of the database is synchronized with the latest changes.

In the replication mode, CouchDB propagates changes asynchronously. All changes are not reflected immediately on all CouchDB servers (few seconds to minutes depending on the network latency between CouchDB servers).

It is also possible to leverage the session affinity (also known as sticky sessions) helps to keep user experience coherent.

Best practices are:

- To proceed with large changes in an “offline” mode e.g. a brand new set of workspaces or views that needs to be created or modified, it is preferable to plan the completeness of this changes apart and then perform the whole update in one time and with no users connected if possible.
- As usual apply the well-know best practices of regular backups as described in this guide (see Chapter 13)

16.4.1 Primary-Primary Replication

From a strict CouchDB perspective, the Primary-Primary replication is active/active.

It means that all the CouchdDB servers are available for users and continuously synchronizing. The replication propagates changes to other databases to update them accordingly based on a dedicated settings.

Pros

- This is the promoted approach by CouchDB that is built for this kind of architecture
- If any CouchDB instance fails, other CouchDB instances continue to update data (no single point of failure)
- Load balancer configuration is easy. Any CouchDB instance can handle requests from any UOC server
- This mode is compatible with session affinity approach

Cons

- Many replications need to be configured: N CouchDB instances equals to N – 1 replications to configure for each CouchDB instance

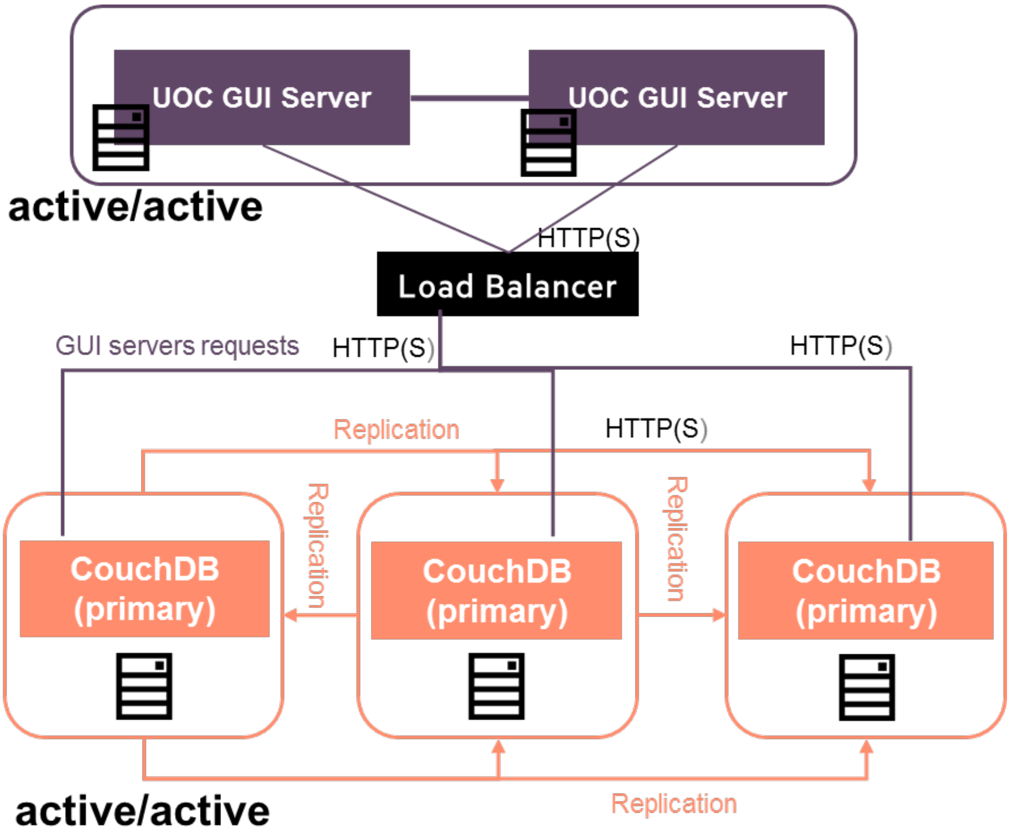


Figure 30: Primary-Primary Replication

Example of configuration for NGINX Load Balancer with the following setting:

- NGINX server listens on port 5980
- Gzip compression enabled
- Round robin load balancing method (default)
- Session affinity using IP hash

```
[root@ossv085 sites-enabled]# more /etc/nginx/sites-enabled/couchdb_primary_primary
server {
    listen      5980;
    server_name localhost;
    gzip        on;
    gzip_types  application/json;

    location / {
        proxy_pass      http://primary;
    }
}
```

```
upstream primary {
        ip_hash;
        server          ossv072.gre.hpecorp.net:5984;
        server          ossv085.gre.hpecorp.net:5984;
        server          ossv086.gre.hpecorp.net:5984;
}
```

16.4.2 Primary-Replica Replication

From a strict CouchDB perspective, the primary replica replication is active/passive

It means that the primary CouchDB server (active) is continuously synchronizing the other servers (passive) and only the primary server is available for user updates. The replication propagates changes from primary to replica instances to keep them updated. A load balancer with HA feature checks if there is an issue with the primary server to select a backup server to become active (master)

Pros

- Few replications to configure → N CouchDB instances means N replications in the primary CouchDB instance
- Conflict errors are made explicit when there are concurrent modifications on the same document

Cons

- The primary CouchDB instance is a single point of failure. Backups are needed in case this instance fails to avoid compromising the deployment
- Load balancing a little more complex to configure (routing based on HTTP verbs & route patterns)
- This mode is incompatible with session affinity approach

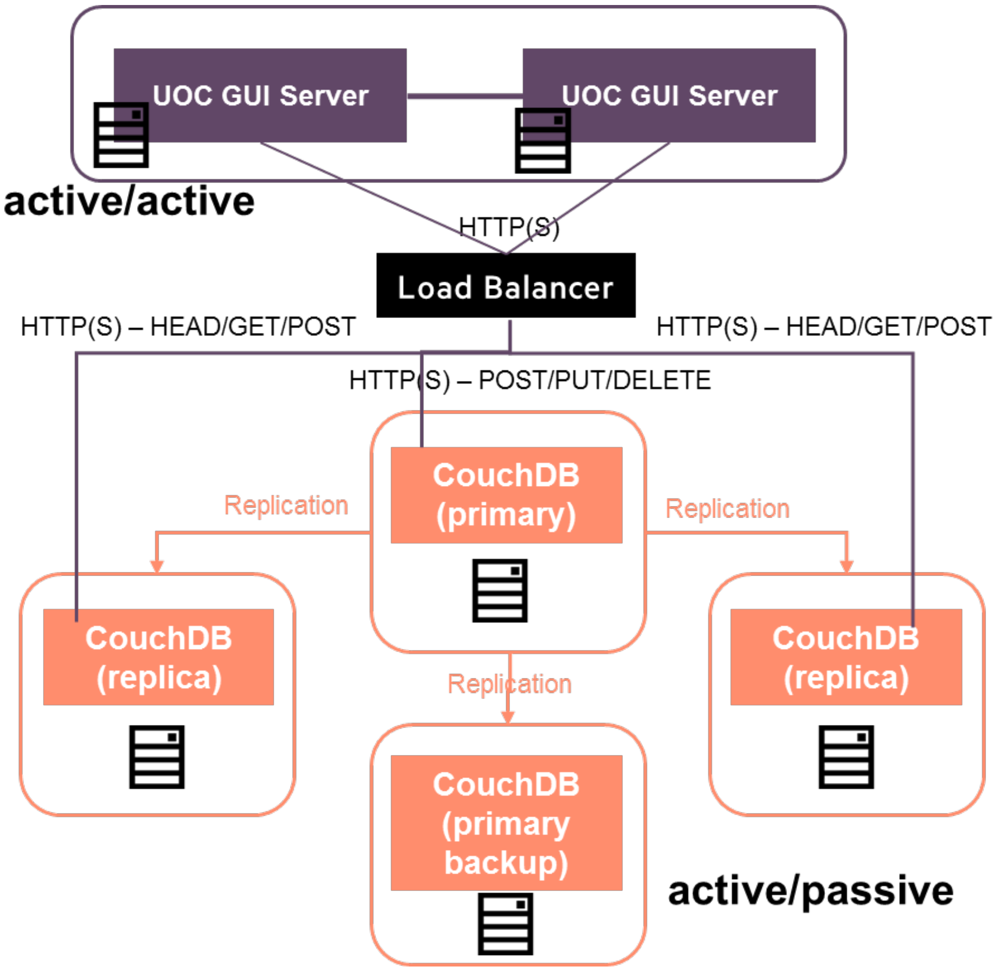


Figure 31: Primary-Replica Replication

Example of configuration for NGINX Load Balancer with the following setting:

- NGINX server listens on port 5980
- Gzip compression enabled
- CouchDB view requests routed to replica
- GET/HEAD requests routed to replica
- Non GET/HEAD requests routed to primary
- Backup for primary & replica CouchDB servers

```
[root@ossv085 sites-enabled]# more /etc/nginx/sites-
enabled/couchdb_primary_replica
server {
    listen      5980;
    server_name localhost;
    gzip        on;
    gzip_types  application/json;

    location ~ \/(.+)\/_design\/(.+)\/_view\/(.+) {
        proxy_pass http://all;
    }

    location / {
        proxy_pass http://all;

        limit_except GET HEAD {
              proxy_pass http://primary;
        }
    }
}

upstream primary {
        server          ossv072.gre.hpecorp.net:5984;
        server          ossv085.gre.hpecorp.net:5984 backup;
}

upstream all {
        server          ossv086.gre.hpecorp.net:5984;
        server          ossv072.gre.hpecorp.net:5984 backup;
        server          ossv085.gre.hpecorp.net:5984 backup;
}
```

16.4.3 UOC Scripts

To ease CouchDB setup in HA mode, a shell script **couchdb_replication.sh** (located in <install_dir>/scripts) is provided allowing to:

- Initialize a CouchDB server with UOC databases
- Replicate all UOC databases from a CouchDB server to another one
- Cancel replications of a CouchDB server

Below sections describe how to use couchdb_replication.sh script to replicate a CouchDB server.

IMPORTANT: It is assumed all your CouchDB servers are installed properly and accessible through the network. An admin user must also be created in each CouchDB server as described in 9.2.2 Create/Modify admin user.

16.4.4 Initialize UOC Databases

UOC databases needs to be initialized in all replica CouchDB servers before starting replicating. To do so, please run couchdb_replication.sh as root and select the first option.

Example:

```
# ./couchdb_replication.sh
This script provides some utilities for CouchDB replication in HA
environments.

Please select:

1) Initialize Unified Console databases on a CouchDB server
2) Replicate a CouchDB server
3) Cancel replications on a CouchDB server
4) Quit
#? 1

Unified Console databases will be destroyed and recreated.
Continue? (y/n)? y

Server protocol ? http
Server host ? ossv085.gre.hpecorp.net
Server port ? 5984
CouchDB admin username ? admin
CouchDB admin password ? admin

Initializing User [user]...
User [user] created successfully with the role [user]
Initializing databases [categories, permissions, roles, users, views,
workspaces]...
Database [categories] has been destroyed and recreated successfully
_design/categories has been created successfully
Database [permissions] has been destroyed and recreated successfully
_design/permissions has been created successfully
Database [roles] has been destroyed and recreated successfully
_design/roles has been created successfully
Database [users] has been destroyed and recreated successfully
_design/users has been created successfully
Database [views] has been destroyed and recreated successfully
_design/views has been created successfully
Database [workspaces] has been destroyed and recreated successfully
_design/workspaces has been created successfully
Database [launches] has been destroyed and recreated successfully
```

```
_design/launches has been created successfully
Database [launch-categories] has been destroyed and recreated
successfully
_design/launch-categories has been created successfully
 All databases have been successfully created and initialized (empty)
```

NOTE: This script dbinit.js is the same as the one run during installation and setup of UOC with interactive questions to avoid to update manually the configuration of UOC.

To check if databases has been created successfully, please run the following command:

```
$ curl <couchdb_protocol>://<couchdb_host>:<couchdb_port>/_all_dbs
```

Example:

```
$ curl http://ossv085.gre.hpecorp.net:5984/_all_dbs
["_replicator","_users","categories","launch-
categories","launches","permissions","roles","users","views","workspaces
"]
```

16.4.5 Replicate UOC Databases

To replicate a CouchDB server to another one, please run couchdb_replication.sh as root and select the second option.

IMPORTANT: The CouchDB server are about to replicate to must be empty (except data from a previous replication to this CouchDB server) else it may cause some issues, including duplicate data.

To check this, please open the CouchDB administration console with your web browser (url: <couchdb_protocol>://<couchdb_host>:<couchdb_port>/_utils) (e.g. http://ossv072.gre.hpecorp.net:5984/_utils).
Log in with your CouchDB admin credentials and check UOC databases contents.

If you need to clear UOC databases contents, reinitialize as described in the previous paragraph.

Example:

```
# ./couchdb_replication.sh
This script provides some utilities for CouchDB replication in HA
environments.


Please select :


1) Initialize Unified Console databases on a CouchDB server
2) Replicate a CouchDB server
3) Cancel replications on a CouchDB server
4) Quit
#? 2
```

```
Unified Console databases will be replicated from one CouchDB server to
another.
Make sure the CouchDB server you are about to replicate to is empty.
Reinitialize Unified Console databases if needed.
Continue? (y/n)? y


Source server protocol ? http
Source server host ? ossv072.gre.hpecorp.net
Source server port ? 5984
Source replicator database ? _replicator
Source CouchDB admin username ? admin
Source CouchDB admin password? admin


Target server protocol ? http
Target server host ? ossv085.gre.hpecorp.net
Target server port ? 5985
Target replicator database ? _replicator
Target CouchDB admin username ? admin
Target CouchDB admin password? Admin


Continuous replication? y


Done
```

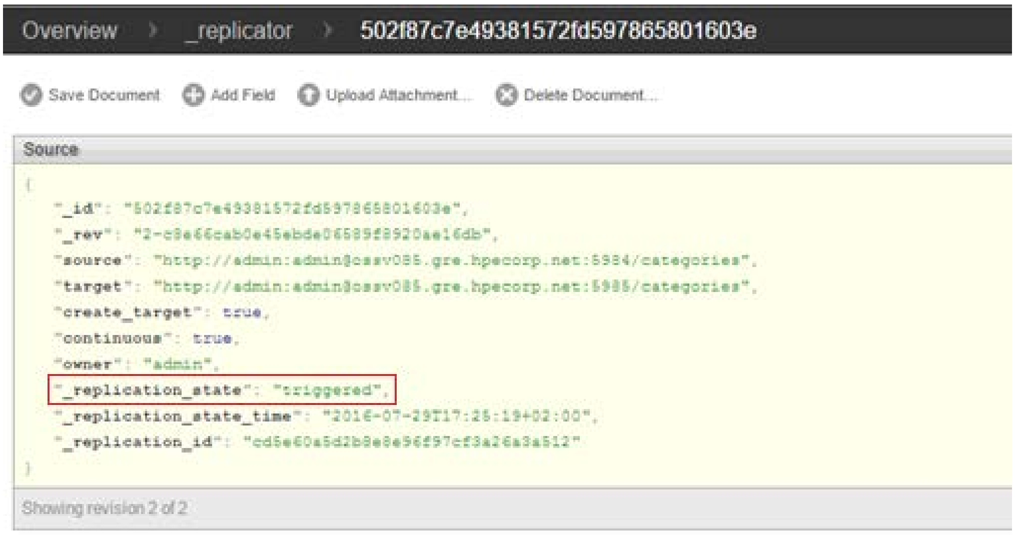
NOTE: _replicator is the default replicator database of CouchDB. Unless CouchDB configuration have been changed, this is the value you have to use.

NOTE: A continuous replication stays active indefinitely, watching for further changes to occur and transferring them. On the contrary, a replication that is not continuous (also known as one-shot replication) is only a one step process.

To ensure all the replications have started successfully, open Futon with your web browser to (url: <couchdb_protocol>://<couchdb_host>:<couchdb_port>/_utils) (e.g. http://ossv072.gre.hpecorp.net:5984/_utils).

Log in with your CouchDB admin credentials.

Open the _replicator database (or your custom replicator database) and ensure all the documents (except _design) have a _replication_state equals to "triggered".



```
{
   "_id": "502f87c7e49381572fd597865801603e",
   "_rev": "2-c9e66cab0e45ebde06589f8920ae16db",
   "source": "http://admin:admin@ossv085.gre.hpecorp.net:5984/categories",
   "target": "http://admin:admin@ossv085.gre.hpecorp.net:5985/categories",
   "create_target": true,
   "continuous": true,
   "owner": "admin",
   "_replication_state": "triggered",
   "_replication_state_time": "2016-07-29T17:25:19+02:00",
   "_replication_id": "cd5e60a5d2b9e8e96f97cf3a26a3a512"
}
```

Figure 32: CouchDB Futon Replication Interface – Setup a replication

You can also check CouchDB logs to be sure no errors happened during the replication process (<couchdb_installdir>/ var/log/couchdb/couch.log).

16.4.6 Cancel Replications

To cancel continuous replications of a CouchDB server, please run couchdb_replication.sh as root and select the third option.

Example:

```
# ./couchdb_replication.sh
This script provides some utilities for CouchDB replication in HA
environments.


Please select :


1) Initialize Unified Console databases on a CouchDB server
2) Replicate a CouchDB server
3) Cancel replications on a CouchDB server
4) Quit
#? 3


Continuous replications will be cancelled.
Continue? (y/n)? y


Server protocol ? http
Server host ? ossv085.gre.hpecorp.net
Server port ? 5984
```

```
Target replicator database ? _replicator
CouchDB admin username ? admin
CouchDB admin password ? admin

Database _replicator has been cleared successfully
Done
```

16.4.7 No interactive mode

The commands initialize, replicate and cancel replication can also be run in a prompt less mode using directly node.js scripts (I.e. without asking for user input), especially useful for automation. Node.js scripts are located in <install_dir>/install/database/HA. There are 3 scripts: **dbreplicate.js**, **dbinit.js** and **dbcancelreplications.js**.

dbinit.js

Usage:

```
# node dbinit.js
```

Command line options:

Option	Description
protocol	CouchDB server protocol - http or https
host	CouchDB server host
port	CouchDB server port
username	CouchDB admin username
password	CouchDB admin password

Example:

```
# node dbinit.js --protocol=http --host=ossv086.gre.hpecorp.net --port=5984
--username=admin --password=admin
```

dbreplicate.js

Usage:

```
# node dbreplicate.js
```

Command line options:

Option	Description
srcprotocol	Source CouchDB protocol – http or https
srchost	Source CouchDB host
srcport	Source CouchDB port
srcreplicator	Source CouchDB replicator database
srcusername	Source CouchDB admin username
srcpassword	Source CouchDB admin password
destprotocol	Target CouchDB protocol
desthost	Target CouchDB host
destport	Target CouchDB port

destreplicator	Target CouchDB replicator database
destusername	Target CouchDB admin username
destpassword	Target CouchDB admin password
continuous	Indicates if the replication shall be continuous or not

Example:

```
# node dbreplicate --srcprotocol=http --srchost=ossv085.gre.hpecorp.net --
srcport=5984 --srcreplicator=_replicator --srcusername=admin --
srcpassword=admin --destprotocol=http --desthost=ossv086.gre.hpecorp.net --
destport=5984 --destreplicator=_replicator --destusername=admin --
destpassword=admin --continuous=false
```

dbcancelreplications.js

Usage:

```
# node dbcancelreplications.js
```

Command line options:

Option	Description
protocol	CouchDB server protocol - http or https
host	CouchDB server host
port	CouchDB server port
replicator	CouchDB replicator database
username	CouchDB admin username
password	CouchDB admin password

Example:

```
# node dbcancelreplications.js --protocol=http --
host=ossv085.gre.hpecorp.net --port=5984 --replicator=_replicator --
username=admin --password=admin
```

16.4.8 UOC Configuration with CouchDB in HA mode

In a HA deployment, UOC2 servers (Node.js) communicates with CouchDB servers through a load balancer.

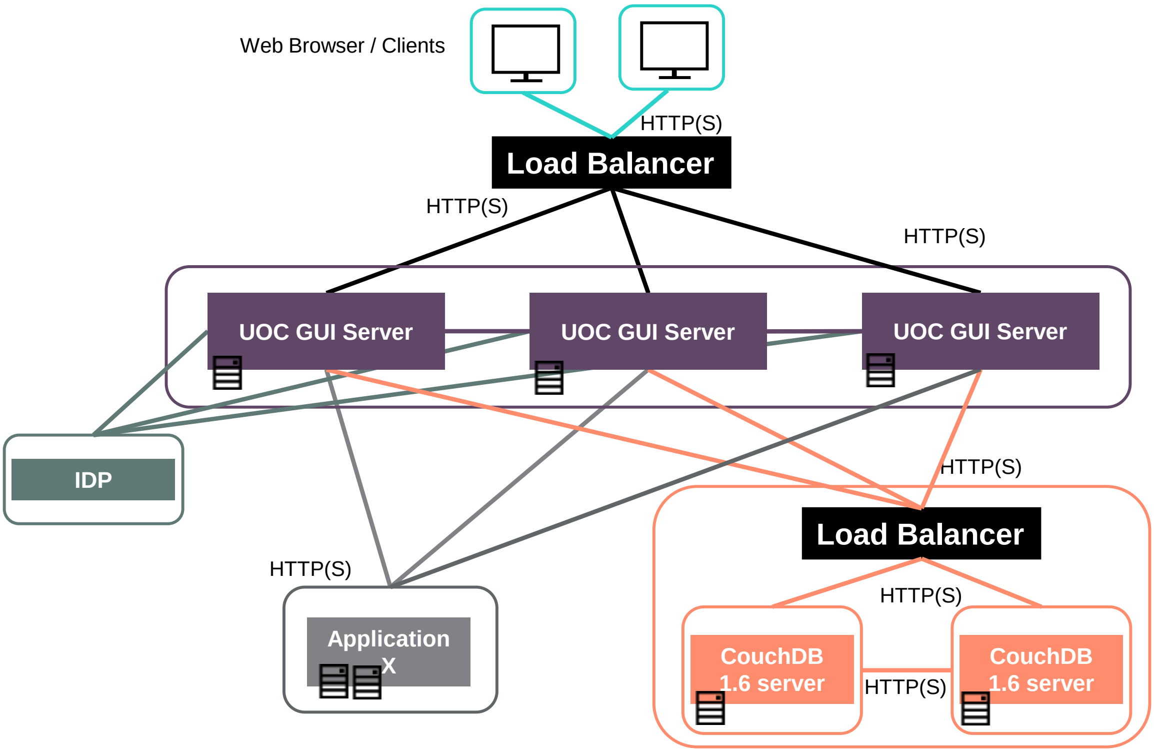


Figure 33: CouchDB configuration in HA context

Please edit <uoc2_installdir>/server/public/conf/config.json and modify database settings to match your load balancer location.

Example:

```
"database": {
      "protocol": "http",
      "host": "ossv085.gre.hpecorp.net",
      "port": "5980",
      ...
}
```

16.4.9 CouchDB Replication Configuration Parameters

Source: https://wiki.apache.org/couchdb/Replication

These parameters can be specified globally in the `default.ini` configuration file:

* **worker_processes** - The number of process the replicator uses (per replication) to transfer documents from the source to the target database. Higher values can imply better throughput (due to more parallelism of network and disk IO) at the expense of more memory and eventually CPU. Default value is 4.

* **worker_batch_size** - Workers process batches with the size defined by this parameter (the size corresponds to number of `_changes` feed rows). Larger batch sizes can offer better performance, while lower values imply that checkpointing is done more frequently. Default value is 500.

* **http_connections** - The maximum number of HTTP connections per replication. For push replications, the effective number of HTTP connections used is min (worker_processes + 1, http_connections). For pull replications, the effective number of connections used corresponds to this parameter's value. Default value is 20.

* **connection_timeout** - The maximum period of inactivity for a connection in milliseconds. If a connection is idle for this period of time, its current request will be retried. Default value is 30000 milliseconds (30 seconds).

* **retries_per_request** - The maximum number of retries per request. Before a retry, the replicator will wait for a short period of time before repeating the request. This period of time doubles between each consecutive retry attempt. This period of time never goes beyond 5 minutes and its minimum value (before the first retry is attempted) is 0.25 seconds. The default value of this parameter is 10 attempts.

* **socket_options** - A list of options to pass to the connection sockets. The available options can be found in the documentation for the Erlang function setopts/2 of the inet module. Default value is `[{keepalive, true}, {nodelay, false}]`.

* **verify_ssl_certificates** - Whether the replicator should validate or not peer SSL certificates. Default value is `false`.

* **ssl_certificate_max_depth** - The maximum allowed depth for peer SSL certificates. This option only has effect if the option **'verify_ssl_certificates** is enabled. Default value is 3.

* **cert_file**, **key_file**, **password** - These options allow the replicator to authenticate to the other peer with an SSL certificate. The first one is a path to a certificate in the PEM format, the second is a path to a file containing the PEM encoded private key, and the third is a password needed to access the key file if this file is password protected. By default these options are disabled.

Chapter 17
Security Guide

This chapter highlights security topics and give all the useful information to harden the UOC platform. Please read it carefully to make sure you have all the information and knowledge on how to secure the HPE Unified OSS Console.

17.1 Overview

Similar to other software solutions, HPE Unified OSS Console is deployed in a computing and networking environment composed of Virtual or Physical Linux servers, linked together by a computing network which needs to be secured before the HPE Unified OSS Console solution is deployed.

This chapter introduces the concept of a secure HPE Unified OSS Console deployment and discusses the planning and architecture to implement such deployment. It is strongly recommended that you read this chapter before proceeding to deploying your HPE Unified OSS Console solution.

The security guidelines relate to both single machine (where all HPE Unified OSS Console servers are installed on the same machine) and distributed (where all HPE Unified OSS Console servers are installed on separate machines) deployments of HPE Unified OSS Console. It also addresses the case when HPE Unified OSS Console is used solely in the Service Provider environment and the case when HPE Unified OSS Console views are accessed by external users over the Internet.

17.2 Terminology

Given terminology may vary from one person another, or depending on the context the following terms and their meaning are defined below.

User Security	Refers to security mechanism supported by HPE Unified OSS Console which enables control of who the user is (authentication), what the user can do (authorization), and what the user did (auditing1).
Privacy	Refers to personally identifiable information about individuals (subscribers), including their behavior, service usage, location, etc. subject to privacy laws which vary from country to country. Private data refers uniquely identifiable data relating to a person or persons which is collected, stored, processed, maintained, and made visible by a system. For example, data can be considered as private when it combines at the same time, the subscriber identity and the consumed services and visited urls, or the subscriber identity and its geographical location, etc. Note that the HPE Unified OSS Console platform alone does not store private data. In the context of an UOC solution, it is the domain specific server(s) integrated as part of the end to end solution which may process, store and expose private data through its integration to HPE Unified OSS Console.

[1] Another wide spread terminology is Accounting (in the context of AAA and related protocols such as Diameter). Here the scope is at OSS software application level rather than network protocol.

	Make sure that if any domain specific server which is part of the end to end solution processes, exposes and/or store private data is properly configured from a privacy standpoint and that all security capabilities of the Unified OSS Console are configured securely when such data is exposed by through the Unified OSS Console.
Encryption	Refers to the process of encoding messages (or information) in such a way those systems external to HPE Unified OSS Console and non HPE Unified OSS Console users (persons, hackers) cannot read it, but that authorized HPE Unified OSS Console Users can read. HPE Unified OSS Console uses encryption to protect exchanged data (between UOC modules and/or between UOC modules and external systems), stored data, as well as user access to the data.
Audit	Refers in this document to the ability to log and track access to HPE Unified OSS Console files, HPE Unified OSS Console directories, and HPE Unified OSS Console resources of the systems where HPE Unified OSS Console is installed. Note that auditing is not part of HPE Unified OSS Console per se, but can be setup at the operating system level to log all user actions.
Hardening	Refers to providing various means of protection of a computer system to eliminate as many security risks as possible. This is typically done by removing all non-essential software programs and utilities from the computer(s) by configuring the system and network components properly, deleting unused files and applying the latest patches.

Finally, note that where words are written in italic, they refer to HPE Unified OSS Console specific files/and or command and/or syntax.

17.3 Before You Start

To best use the security guidelines given here for your particular organization, you should do the following before starting your deployment

1. Evaluate the security risk/security state of the computing and network environment where you are going to deploy HPE Unified OSS Console, and use the conclusions when deciding how to best integrate the HPE Unified OSS Console into your network.
2. Review all the security HPE Unified OSS Console guidelines: a good understanding of HPE Unified OSS Console security capabilities will facilitate designing a solid plan to deploy a secure HPE Unified OSS Console solution

NOTE: the security information provided in this chapter **is not intended as a guide to making a security risk assessment** for your computerized systems. Should you require a risk assessment for your computer and networking environment, HP Enterprise security solutions has a comprehensive offer covering Managed Security Services and Security Consulting. Contact your HPE Sales Representative to know more.

The following table comprises a list of security best practices that HPE recommends in the computing and networking environment. This list is provided for information and does not replace a security risk assessment guide as mentioned above.

Topic	Best Practice
Accounts	Limit the number of local accounts. Integrate the appliance with your enterprise Identity Provider solution and set them the correct level of rights. Manage the user accounts proactively (forced password renewal, forced password complexity, disabling and deleting obsolete users accounts) Several anonymous users are created like uoc, couchdb and redis. To avoid an attacker to exploit component vulnerabilities, it is recommended to set the minimal privileges to these accounts. A Unix administrator can define limited-privilege roles and associated these roles to users in charge of their administration (http load balancer, Apache CouchDB, uoc, Domain data server(s)…)
Passwords	Change the local HPE Unified OSS Console accounts passwords periodically, according to your password policies. Ensure that passwords are long enough and include at least three of these types of characters: ◦ Numeric character ◦ Lowercase alphabetic character ◦ Uppercase alphabetic character ◦ Special character
System management & auditing	Perform regular O/S patch updates Install & regularly update antivirus engines and software Monitor actively the systems logs, audit files and anti-virus logs to detect any abnormalities Protect key assets such as file systems, databases & storage
Nonessential services	Remove or disable nonessential services in the management environment. Ensure that you continue to minimize services when you configure HPE Unified OSS Console systems and domain specific servers systems (including network ports not in use) to significantly reduce the number of ways your environment could be attacked. Regarding Red Hat Linux Operating system, follow the Red Hat O/S security recommendations (https://access.redhat.com/) such as Security guides, patches, recommended updates, hardening recommendations.
Proactive reviews and updates to security features and patches	Ensure that a process is in place to pro-actively and periodically determine if software, firmware & security updates are available. Install updates for all components in your environment on a regular basis. • Subscribe to the HPE security bulletin as described in the related section later on in this document • Subscribe to receiving email notifications of security and enhancement updates advised on the Red Hat Customer Portal https://access.redhat.com/security/ • Consider subscribing to a vulnerability & exposure site (such as the Common Vulnerabilities and Exposures official site)
Network	HPE recommends a strict separation of the management LAN and production LAN, using VLAN or firewall technology (or both) to maintain the separation. Please refer to HPE Unified OSS Console deployment section. Grant management LAN access to authorized personnel only: Infrastructure administrators, Network administrators, and Server administrators. Disable SSH ports Review the TLS (SSL) configuration (ciphers and algorithms) and configure TSL to provide communications security over the network. Do not connect any management systems, inclusive of HPE Unified OSS Console directly to the Internet. If users from your internal organization require to access HPE Unified OSS Console (or other management functionality) to the Internet, use a corporate VPN (virtual private network) that provides firewall protection.

	If the HPE Unified OSS Console deployment requires that HPE Unified OSS Console functionality is exposed to external parties like Tenants, follow the additional security guidelines for these cases (see relevant section further in the document) If connected on the internet then use this site for a good test (https://www.ssllabs.com/ssltest/), get permission before running the test. Regularly monitor network traffic for suspicious activity
Certificates	Use certificates signed by a trusted certificate authority (CA) to ensure the integrity and authenticity of your HTTPS connections. ● between users' browsers and HPE Unified OSS Console server(s) ● between HPE Unified OSS Console server(s) and domain specific servers, ● between HPE Unified OSS Console server(s) and the GUI Database (Apache CouchDB) ● between users' browsers and the Identity Provider (SAML based) Ideally, you should use your company's existing CA and import their trusted certificates. The trusted root CA certificate should be deployed to user's browsers that will contact systems and devices that will need to perform certificate validation.
Accidental actions and other events	Implement systematic backup policy. Employ qualified, skilled, and trained staff Use in-house resources for administration. Secure infrastructure from fire, flood, earthquake, and have disaster recovery plans
Turn on HPE Unified OSS Console security features	Configure all HPE Unified OSS Console security features as described in this document.
Private data	Note that the HPE Unified OSS Console platform alone does not store private data. In the context of an UOC solution, it is the domain specific server(s) integrated as part of the end to end solution which may process, store and expose private data through its integration to HPE Unified OSS Console. Make sure that if any domain specific server which is part of the end to end solution processes, exposes and/or store private data is properly configured from a privacy standpoint and that all security capabilities of the Unified OSS Console are configured securely when such data is exposed by through the Unified OSS Console.

Figure 34: Best practices general considerations

17.4 Subscribing to HPE Security Bulletins

Procedure for Subscribing to Security Bulletins

1. Open a browser to go to the HPE Updates page

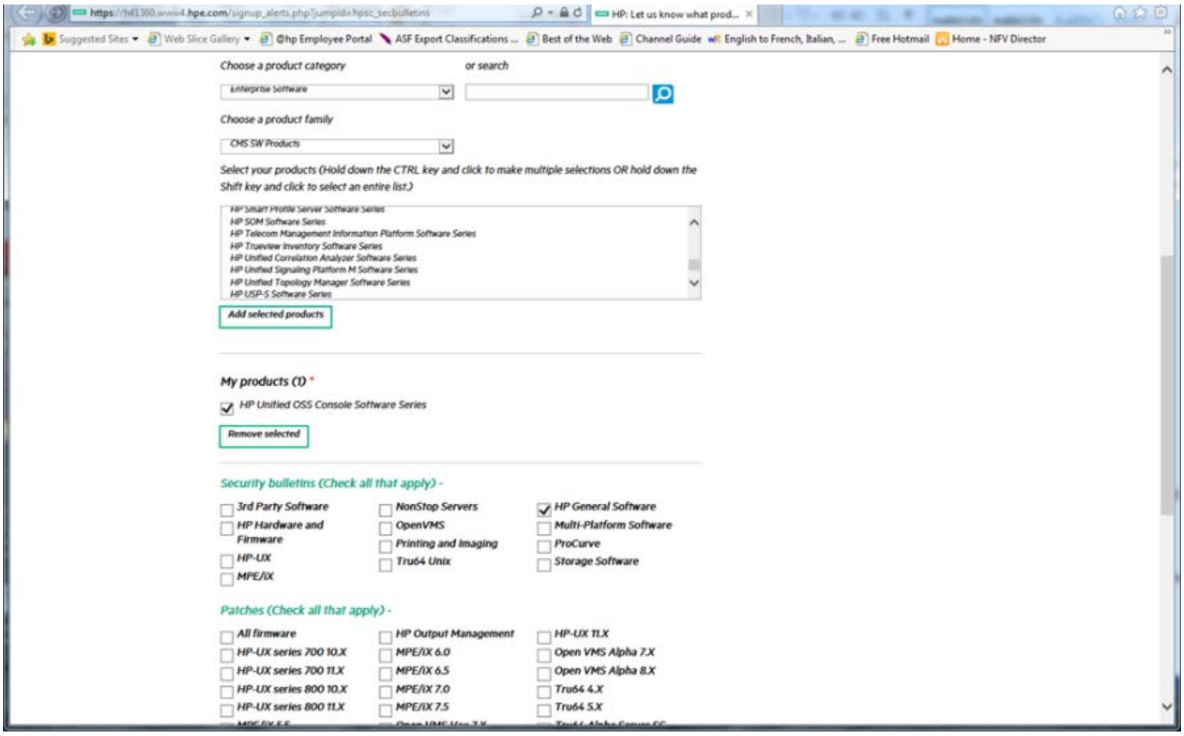
https://h41360.www4.hpe.com/signup_alerts.php?jumpid=hpsc_secbulletins

2. Do one of the following:
   - Sign in if you are a registered customer.
   - Enter your email address to sign-up now.

3. Select the following fields
   - Product Category: Enterprise Software
   - Product Family: CMS SW Products
   - Select your product(s) – For HPE Unified OSS Console, select “ **HPE Unified OSS Console Software Series**”

4. Select subscription
   - Select the relevant security bulleting you want to subscribe to (HPE General Software and Multi-Platform Software)
   - Select OS “patches” if relevant to your solution

5. Click “subscribe”


https://h41360.www4.hpe.com/signup_alerts.php?jumpid=hpsc_secbulletins
HP: Let us know what prod...
Suggested Sites
Web Slice Gallery
@hp Employee Portal
ASF Export Classifications ...
Best of the Web
Channel Guide
English to French, Italian, ...
Free Hotmail
Home - NFV Director

Choose a product category: Enterprise Software

or search

Choose a product family: CMS SW Products

Select your products (Hold down the CTRL key and click to make multiple selections OR hold down the Shift key and click to select an entire list.)

HP Smart Profile Server Software Series
HP SOM Software Series
HP Telecom Management Information Platform Software Series
HP Trueview Inventory Software Series
HP Unified Correlation Analyzer Software Series
HP Unified Signaling Platform M Software Series
HP Unified Topology Manager Software Series
HP USP-S Software Series

Add selected products

My products (1) *

☑ HP Unified OSS Console Software Series

Remove selected

Security bulletins (Check all that apply) -

□ 3rd Party Software
□ HP Hardware and Firmware
□ HP-UX
□ MPE/iX
□ NonStop Servers
□ OpenVMS
□ Printing and Imaging
□ Tru64 Unix
☑ HP General Software
□ Multi-Platform Software
□ ProCurve
□ Storage Software

Patches (Check all that apply) -

□ All firmware
□ HP-UX series 700 10.X
□ HP-UX series 700 11.X
□ HP-UX series 800 10.X
□ HP-UX series 800 11.X
□ HP Output Management
□ MPE/iX 6.0
□ MPE/iX 6.5
□ MPE/iX 7.0
□ MPE/iX 7.5
□ HP-UX 11.X
□ Open VMS Alpha 7.X
□ Open VMS Alpha 8.X
□ Tru64 4.X
□ Tru64 5.X

Figure 35: HPE Security Bulletins

17.5 Deployments

Consider deploying the HPE Unified OSS Console solution into a secured IT environment.

NOTE: an HPE Unified OSS Console solution includes the HPE Unified OSS Console software plus a number of pre-integrated domain specific applications servers. A secure solution deployment must also address the domain specific servers which are part of the solution. For security guidelines and features supported by these servers, please refer to the domain specific servers user manuals.

HPE Unified OSS Console have been tested with high level of security in an enterprise network. All HTTP connections have been configured with SSL and the HPE authorities' certificate. All local data files have been protected by admin rights to access and allows modification. All default password or passphrase have been changed with a high complexity level.

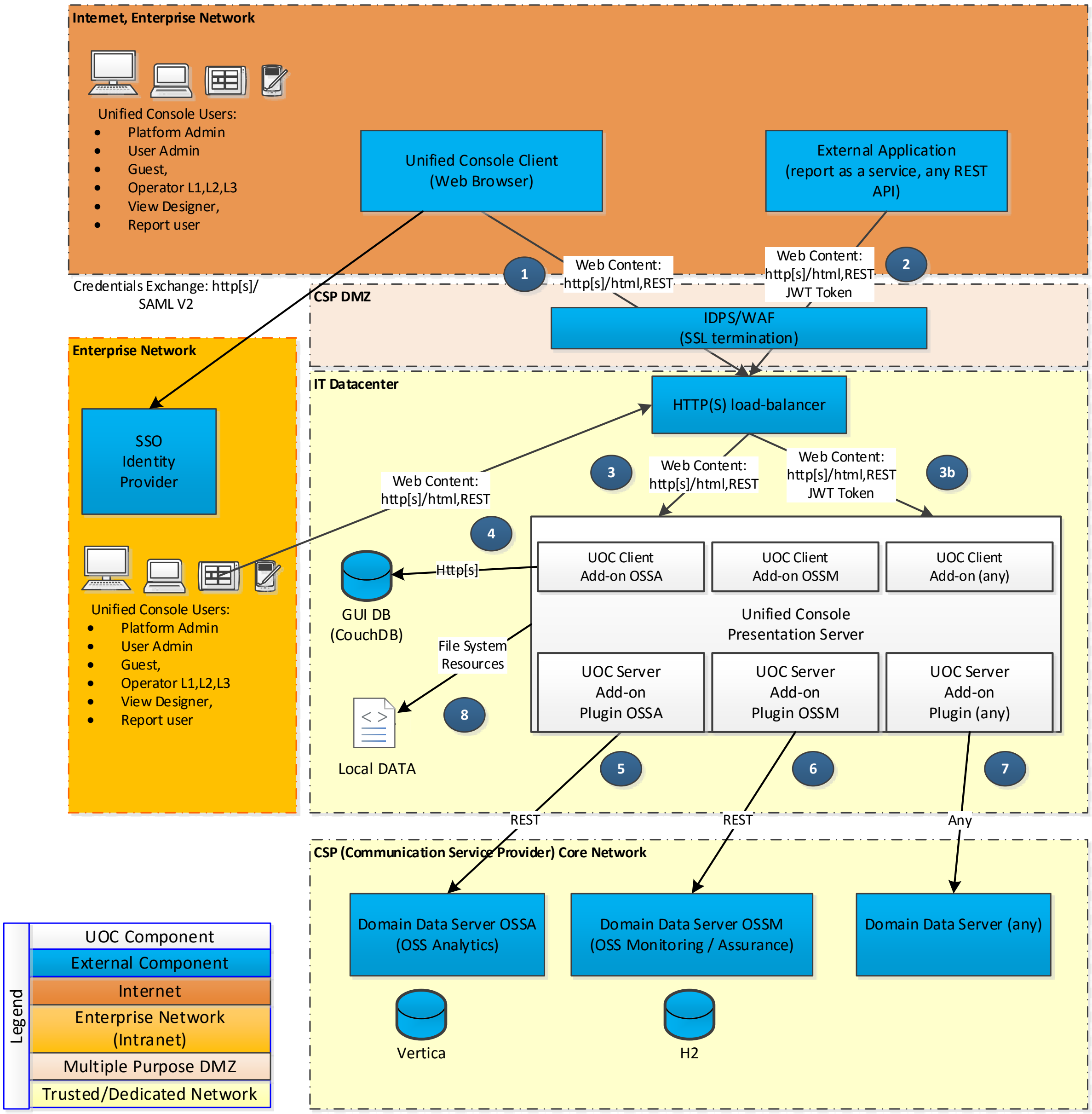

Internet, Enterprise Network
Unified Console Users:
Platform Admin
User Admin
Guest,
Operator L1,L2,L3
View Designer,
Report user
Unified Console Client (Web Browser)
External Application (report as a service, any REST API)
Credentials Exchange: http[s]/ SAML V2
1
Web Content: http[s]/html,REST
Web Content: http[s]/html,REST JWT Token
2
CSP DMZ
IDPS/WAF (SSL termination)
Enterprise Network
SSO Identity Provider
Unified Console Users:
Platform Admin
User Admin
Guest,
Operator L1,L2,L3
View Designer,
Report user
IT Datacenter
HTTP(S) load-balancer
Web Content: http[s]/html,REST
3
Web Content: http[s]/html,REST
Web Content: http[s]/html,REST JWT Token
3b
4
Http[s]
GUI DB (CouchDB)
UOC Client Add-on OSSA
UOC Client Add-on OSSM
UOC Client Add-on (any)
Unified Console Presentation Server
File System Resources
UOC Server Add-on Plugin OSSA
UOC Server Add-on Plugin OSSM
UOC Server Add-on Plugin (any)
8
Local DATA
5
6
7
REST
REST
Any
CSP (Communication Service Provider) Core Network
Domain Data Server OSSA (OSS Analytics)
Domain Data Server OSSM (OSS Monitoring / Assurance)
Domain Data Server (any)
Vertica
H2
Legend
UOC Component
External Component
Internet
Enterprise Network (Intranet)
Multiple Purpose DMZ
Trusted/Dedicated Network

Figure 36: UOC Server – Security in HPE Unified OSS Console deployments

HPE Unified OSS Console supports different deployment use cases:

1. Case when HPE Unified OSS Console is used solely within the Service Provider environment
2. Case when HPE Unified OSS Console is deployed within the Service Provider environment and within the Service Provider's tenants over a trusted network (VPN or IDPS[2]/WAF[3])
3. Case when HPE Unified OSS Console is accessed over the internet

IMPORTANT:
If UOC default security level is not dedicated to the internet access. If you need to expose UOC from the internet, you need to deploy a dedicated trusted network with controller access behind a DMZ, containing an intrusion Detection and Prevention system or a Web Application Firewall that filters or blocks malicious (layer 2 up to 7) traffic from the internet

IMPORTANT:
If the Load Balancer is deployed in the solution, it is also recommended to use HTTPS and terminate all SSL traffic by the Load Balancer.

IMPORTANT:
It has been clearly identified that the connection between UOC Server and Domain or data server(s) must use the HTTPS protocol increase significatively the security and confidentiality between these 2 components. Please check carefully the 17.11 Secure Socket Layer (SSL/TLS) chapter to understand the steps required to install such configuration (configure SSL/TLS, generate certificates, install them…)

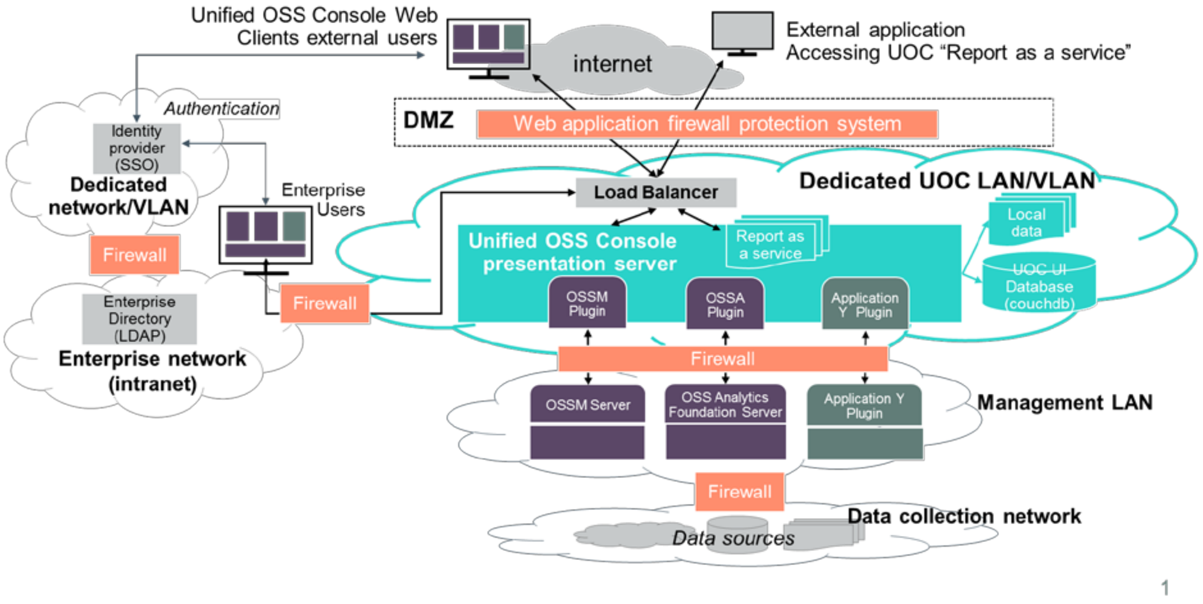


Figure 37: UOC Console accessed over the internet

[2] IDPS : Intrusion Detection and Prevention Systems
[3] WAF: Web Application Firewall

17.6 Security features summary

The following lists the HPE Unified OSS Console security features and guidance that should be followed for HPE Unified OSS Console V2.3 deployments.

- Use of Single Sign On (SSO) through integration to the Customer Identity Provider using HPE Unified OSS Console Security assertion markup language (SAML) interface
- Configure HPE Unified OSS Console URL-safe communications by configuring the JSON Web Token so that a JWT Token is generated for the user session. Your secret passphrase must be changed.
- Ensure communication between users and systems is configured with https, uses SSL/TLS (SSL v2 and v3 are disabled due to a security vulnerability. So uses TLS 1.1 or 1.2) and certificate SHA-2.
- Manage HPE Unified OSS Console users securely: avoid user account sharing, configure named HPE Unified OSS Console users, and restrict their rights by configuring their roles and permissions.
- Actively monitor HPE Unified OSS Console user activities by regularly reviewing the HPE Unified OSS Console audit logs: sessions.log and security.log.
- Setup limited privilege for roles and users created for the solution (UOC, CouchDB, data servers…)
- Lock and delete User accounts as per their employee, role change, employment termination and any event requiring that a user or set of users should stop access and use HPE Unified OSS Console.
- Configure the solution leveraging SSL (SHA-2) especially between UOC Server and associated Domain or Data servers.

Details regarding these HPE unified OSS Console security features are explained in previous sections of this document.

Further to HPE Unified OSS Console alone, also configure the domain specific servers(s) access rights, user roles as well as, where applicable, access to privacy data. Please refer to respective domain specific User Installation guides, security section.

17.7 Authentication

For user authentication the product supports two options: either SAML/SSO based authentication via integration with an external SAML identity provider or authentication using a username and password database stored locally in the product (Local Authentication).

The SAML/SSO based authentication option is recommended for a production environment. The Local Authentication option is not recommended as it does not check passwords for a minimum length, complexity or policy and it increases the administration cost and risk of maintaining users and their access rights up to date.

JSON Web tokens are supported for authentication of external application requests. In this case authentication is performed by the use of a shared secret passphrase shared with the application which is then used to add an authentication code to each request made by the application. The application must be a trusted and controlled application as the secret passphrase is the same for all applications and the application can make a request on behalf of any user.

17.7.1 SAML / SSO Authentication option

SAML (Security Assertion Markup Language): Unified OSS Console provides an integration with identity providers through the SAML V2.0 protocol. Users can be managed externally (LDAP, files …) and the product supports the SSO using this identity provider. It is the recommended production mode for large volume of users.

Note: The Open source project Picket Link (http://picketlink.org) and keycloak (http://www.keycloak.org/ are options as an identity provider and has been tested with our solution.

See 17.7.4 How to configure SAML / SSO

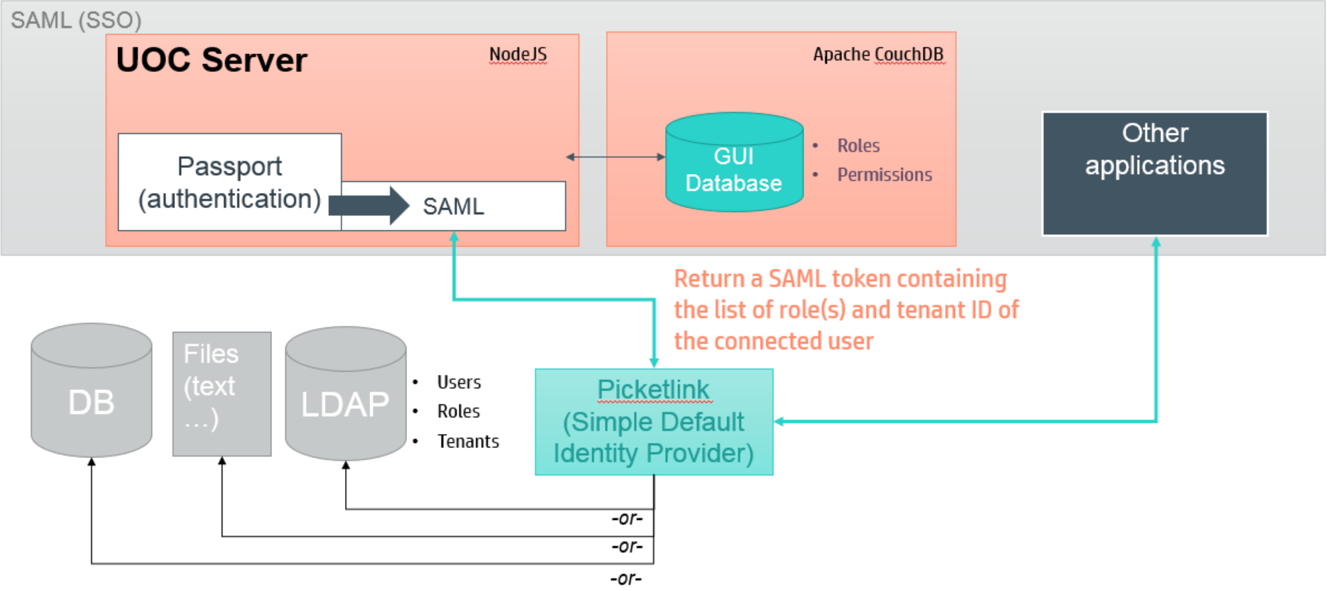

SAML (SSO)
UOC Server
NodeJS
Passport (authentication)
SAML
Apache CouchDB
GUI Database
Roles
Permissions
Other applications
Return a SAML token containing the list of role(s) and tenant ID of the connected user
DB
Files (text …)
LDAP
Users
Roles
Tenants
Picketlink (Simple Default Identity Provider)
-or-
-or-
-or-

Figure 38: UOC Authentication modes (SAML)

IMPORTANT:
SAML access tokens are sent from the Identify Provider via the user's browser to the UOC Server. The SAML access token must be digitally signed and UOC configured to only accept digitally signed access tokens to prevent users from tampering with them as they transit via the user's browser.

17.7.2 Local Authentication option

Local: It is a built-in authentication mode based on a local Document database in charge of managing the users and their associated roles. It is mainly for demo purpose or very small deployments. This mode does not support the SSO and does not provide a high level of security. It is not recommended to use it in production.

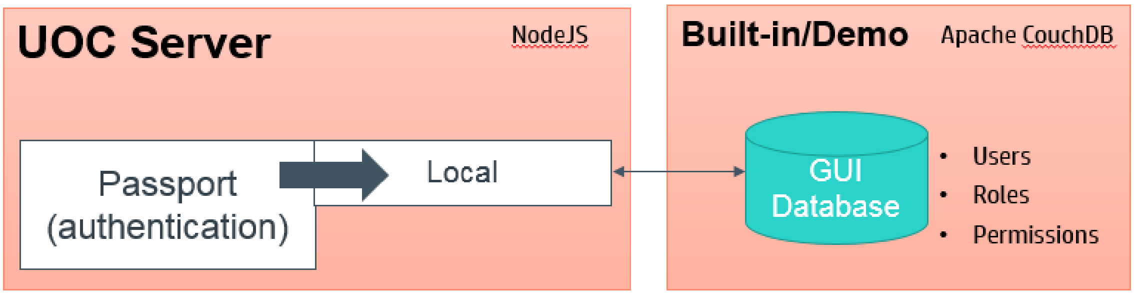


Figure 39: UOC Authentication modes (Local)

17.7.3 How to configure Local Authentication

To enable the local authentication mode, please edit the file **<install_data_dir>/server/public/conf/config.json** and set the authentication mode to 'local'. Also, ensure your database settings are accurate (couchdb host, user, password)

Example of **<install_data_dir>/server/public/conf/config.json**:

```
…
  "database": {
    "protocol": "http",
    "host": "127.0.0.1",
    "port": "5984",
    "username": "user",
    "password": "user"
  },
  "authentication": {
    "mode": " local ",
    …
  },…
```

These information are protected by UNIX rights (uoc user). Anyone should not have access to this sensitive information.

17.7.4 How to configure SAML / SSO option

To enable the SAML authentication mode, please edit the file **<install_data_dir>/server/public/conf/config.json**

Example of config.json:

```
…
"authentication": {
      "mode": "saml"
  },
  "saml": {
     "idp": {
        "entryPoint": "http://localhost:8080/idp",
        "identifierFormat": "urn:oasis:names:tc:SAML:2.0:nameid-format:entity",
        "acceptedClockSkewMs": 0,
     },
     "sp": {
        "issuer": "http://localhost:3000"
     }
  },
…
```

The attribute 'mode' under 'authentication' sets the SAML authentication mode.

The 'saml' part defines several mandatory options regarding the SAML authentication. The 'idp' subpart concerns your Identity Provider whereas 'sp' concerns your Service Provider that is your UOC server.

About the 'idp' subpart:

- 'entryPoint' sets your Identity Provider entry point for authentication requests. Authentication requests are carried in the URL query string of an HTTP GET request.
- 'identifierFormat' indicates what SAML name identifier format you want to use.
- acceptedClockSkewMs (optional) allows to set milliseconds of skew that is acceptable between the UOC server and the Identity Provider when checking OnBefore and NotOnOrAfter assertion condition validity timestamps

About the ‘sp’ subpart:

- ‘issuer’ defines the EntityID of the UOC server (Service Provider), usually the URL to access to your UOC application.

The certificate to be used to validate the SAML assertions signed by the Identity Provider must also be configured.

The following file needs to be edited **<install_data_dir>/server/public/conf/config.json.**

All the certificates must be copied in the folder **<install_data_dir>/server/public/ssl.**

Only certificates in the PEM format are supported as to the SAML configuration.

Example of SAML configuration with SAML assertion digital signature checking:

```
"saml": {
    "idp": {
      "entryPoint": "http://localhost:8080/idp",
      "identifierFormat": "urn:oasis:names:tc:SAML:2.0:nameid-format:entity",
      "certificate": "idpcert.pem"
    },
    "sp": {
      "issuer": "http://localhost:3000",
      "privateKey": "nodekey.pem"
    },
    "signature": true,
  },
```

About the ‘idp’ part:

- ‘certificate’ sets the public certificate of your Identity Provider. This certificate is used for checking the generated SAML assertions signed by and received from the Identity Provider to make sure they haven’t been modified.

About the ‘sp’ part:

- ‘privateKey’ sets the private key of your UOC server. This is used for signing SAML requests sent to the Identify Provider.

17.7.5 How to configure the JSON Web Token

JSON Web Token (JWT) is a compact URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JSON object that is digitally signed using JSON Web Signature (JWS).

You can get more details about this token on the website: https://tools.ietf.org/html/draft-ietf-oauth-json-web-token-32

Our JSON Web token integrates several information like:

- Issued token generation date
- Expiration token date
- User identifier
- User name
- A set of user roles (list of role identifier)

To configure this token generation, please edit the file <install_data_dir>/server/public/conf/config.json and set the jwt section.

You can change the algorithm used by the generator (default is HS256), the secret passphrase and the expiration time (in minutes)

Example of config.json:

```
...
  "authentication": {
                 ...
    "jwt": {
      "algorithm": "HS256",
      "expiresInMinutes": 1440,
      "secret": "TheWalkingSkeleton"
    }
  },

...
```

NOTE: It is possible to customize the JSON Web Token algorithm (default is HS256) , expiration time (default is 24h) and secret passphrase in the same configuration file

IMPORTANT:
The secret passphrase is dynamically generated during the first installation to make sure each installation have a unique secret passphrase.
It is in charge of the platform administrator to customize this passphrase.

17.7.6 How to generate JSON Web Token

Mainly for the usage of the UOC RESTAPI, it is mandatory to generate a valid JSON Web token. It is possible to generate valid JSON Web Token externally using multiple tools. You can use for example the following web sites:

http://jwt.io/

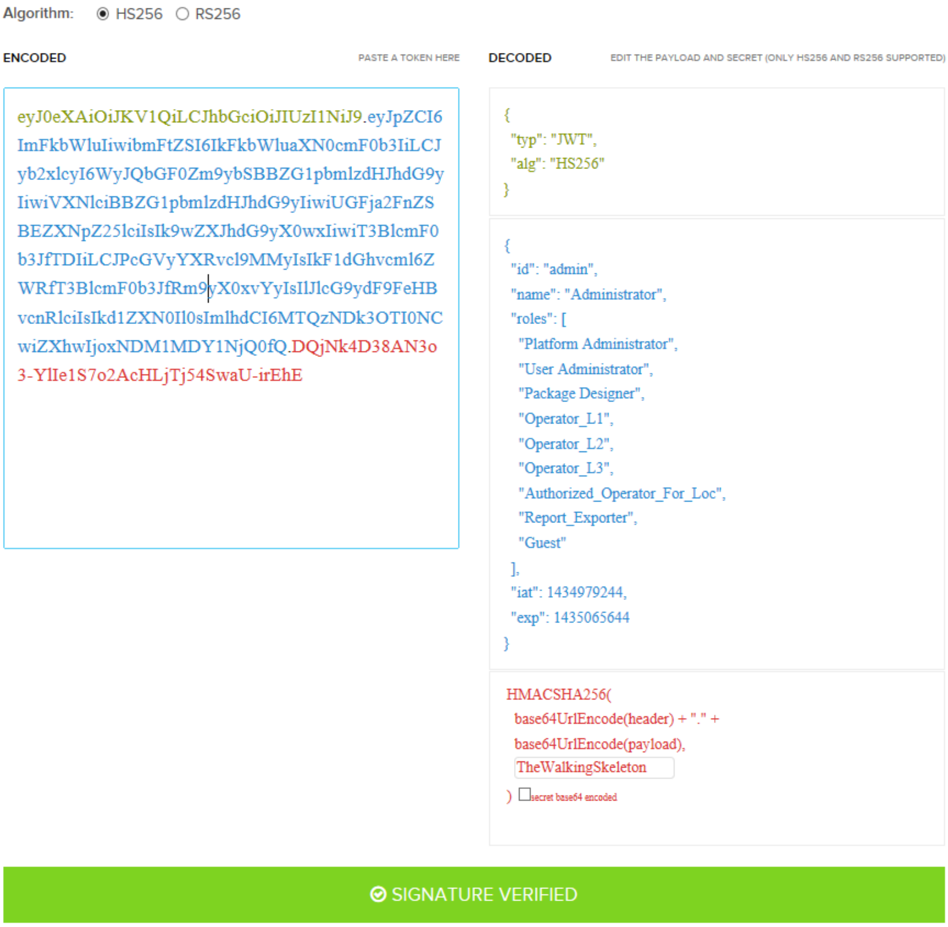
http://jwtbuilder.jamiekurtz.com/

The generated token can be used to call the UOC REST API. For example, it is very useful if you want to export regularly PDF reports from UOC.

NOTE: To generate a valid token, you must ensure the configuration of the authentication match the parameters you define for your external tools (passphrase, user id, role identifiers…)

IMPORTANT:
The application using the JSON Web Token to access the UOC REST API must be a trusted and controlled application as the secret passphrase is the same for all applications and the application can make a request on behalf of any user.

It is also possible to get a valid token in the preferences of the user.



Algorithm: ● HS256 ○ RS256

ENCODED
PASTE A TOKEN HERE
DECODED
EDIT THE PAYLOAD AND SECRET (ONLY HS256 AND RS256 SUPPORTED)

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6
ImFkbWluIiwibmFtZSI6IkFkbWluaXN0cmF0b3IiLCJ
yb2xlcyI6WyJQbGF0Zm9ybSBBZG1pbmlzdHJhdG9y
IiwiVXNlciBBZG1pbmlzdHJhdG9yIiwiUGFja2FnZS
BEZXNpZ25lciIsIk9wZXJhdG9yX0wxIiwiT3BlcmF0
b3JfTDIiLCJPcGVyYXRvcl9MMyIsIkF1dGhvcml6Z
WRfT3BlcmF0b3JfRm9yX0xvYyIsIlJlcG9ydF9FeHB
vcnRlciIsIkd1ZXN0Il0sImlhdCI6MTQzNDk3OTI0NC
wiZXhwIjoxNDM1MDY1NjQ0fQ.DQjNk4D38AN3o
3-YlIe1S7o2AcHLjTj54SwaU-irEhE
```

```
{
 "typ": "JWT",
 "alg": "HS256"
}
```

```
{
 "id": "admin",
 "name": "Administrator",
 "roles": [
  "Platform Administrator",
  "User Administrator",
  "Package Designer",
  "Operator_L1",
  "Operator_L2",
  "Operator_L3",
  "Authorized_Operator_For_Loc",
  "Report_Exporter",
  "Guest"
 ],
 "iat": 1434979244,
 "exp": 1435065644
}
```

```
HMACSHA256(
 base64UrlEncode(header) + "." +
 base64UrlEncode(payload),
 TheWalkingSkeleton
) □secret base64 encoded
```

Figure 40: Example of generation of JSON Web Token (http://jwt.io)

It is mandatory to follow this syntax to make sure the token is valid in the Unified OSS Console:

```
{
 "id": "admin",
 "name": "Administrator",
 "roles": [
   "Platform Administrator",
   "User Administrator",
   "Package Designer",
   "Operator_L1",
   "Operator_L2",
   "Operator_L3",
 ],
 "iat": 1434979244,
 "exp": 1435065644
}
```

Where:

- Id is the identifier of the user
- Name is the name of the user
- Roles are the list of role identifier associated to the user
- Iat is the issued-at time
- Exp is the expiration date/time

17.7.7 Enable Session Sliding

Session sliding allows to renew automatically at regular intervals the user session (JWT token) as long as the user is active. If a user is inactive beyond a maximum amount of time, the latter is disconnected by UOC. Inactivity detection is based on client-side events like mouse moves, scrolling and key presses.

To enable session sliding, please edit the file **<install_data_dir>/server/public/conf/config.json**

Example of **<install_data_dir>/server/public/conf/config.json**:

```
{

   "authentication": {
      ....
      "session": {
            “sliding”: false,
            “idleTimeout”: “30m”,
            “refreshInterval”: “10m”
      }
   },
   ...
}
```

The property ‘sliding’ allows to enable session sliding instead of sessions with a static expiration time (default is false).

‘idleTimeout’ specifies the maximum amount of time a user can be inactive before being disconnected and ‘refreshInterval’, the refresh period of sessions.

TIP: It is strongly recommended to adjust JWT token lifetime when activating session sliding. Usually in this mode, session's lifetime are shorter given that sessions are refreshed over time.
Besides, please ensure in the configuration that the session refresh interval is smaller than JWT token lifetime in order to avoid trying to refresh a session already expired.

CAUTION: In SAML authentication mode, the Identity Provider session is not refreshed by the session sliding mechanism, there is no synchronization between the two sessions. This may cause some issues at logout if the Identity Provider session expires in the meantime depending on your Identity Provider configuration / implementation.

17.8 User and Password Management

If the local authentication mode is used, all the user sensitive information like the user password is stored in the GUI Database (CouchDB).

The passwords are stored in a protected form – a 128 bit long key, hashed using the PBKDF2 algorithm.

In order to hash the passwords, a salt is generated using a cryptographically secure pseudo-random function and is 64 bits long as recommended by the standard. The salt is prepended to the hashed password (64 bits long) and saved in the database.

NOTE: it is strongly recommended to turn on the SSL encryption between UOC Server and the GUI Database to avoid the user passwords being exposed to users with access to the UoC server network.

17.9 Role Based Access Control (RBAC)

The product integrate a strong role based access control that drive the user interface and the access to information. The Definition is based on the standard ANSI INCITS 359-2004.

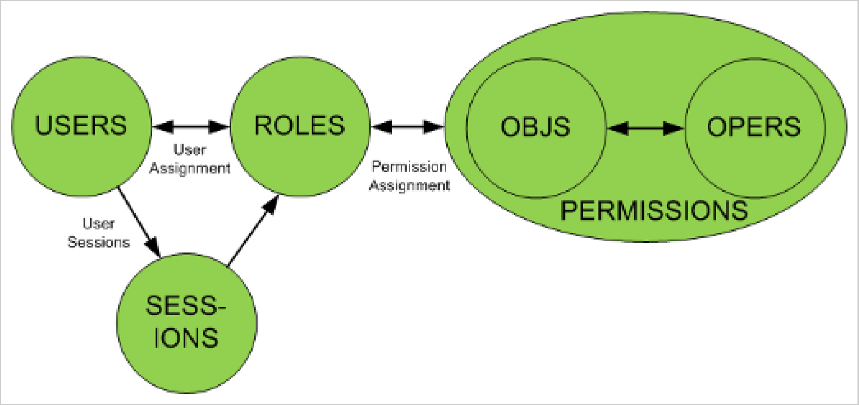


Figure 41: RBAC - ANSI INCITS 359-2004

A role is a job function or a title which defines an authority level

A User will have one or several roles (ex: Operator Level 1)

A role has one or several permissions

A permission is an approval of a mode of access to a resource. It is defined by an operation and an object or resource (ex: create user)

As soon as the end user has been authenticated, a list of his roles are checked, a list of permissions are loaded and the graphical interface always apply these access right for display and actions (ex: Only a user administration can have access to administration page in charge of creating users in the platform, only an operator level 3 can open a set of advanced analysis dashboards, etc….)

17.9.1 Roles

A set of roles is defined by default:

- Guest
- User Administrator
- Platform Administrator
- Operator Level 1
- Operator Level 2
- Operator Level 3
- Package (or value pack) Designer
- View Designer
- Report Exporter

This list can be extended through your identity provider used by SAML authentication, or by using the Role Administration page in the application in case you are using the local built-in authentication.

TIP: All roles defined in a specific domain should be prefixed by the add-on identifier to avoid any conflict on the UOC platform.

For each role, a set of permission are defined to secure the user interface.

The list of roles associated to a user will impact the user interface available and operations he is able to execute.

It is strongly recommended to tune fine grain the role and permissions of a user for security reasons.

Example: An operator level 1 has very few available actions and limited access to specific dashboard. It will also impact the available list of dimensions and facts in the analysis tool.

17.9.2 Permissions

UOC has an internal list of pre-defined permissions that will impact the User interface. When an administrator define a new role, he needs to associate some of these permissions to indicate to UOC what actions are available for the connected user.

Here is the list of available permissions:

Group	Operation	Object	Identifier	Description
Workspace Management	Browse	Workspace	browse_workspace	Allows the user to browse and display workspaces
	Create	Workspace	create_workspace	Allows the user to create workspaces
	Delete	Workspace	delete_workspace	Allows the user to delete workspaces
	Save	Workspace	save_workspace	Allows the user to save or save as… workspaces
	Edit	Workspace	edit_workspace	Allows the user to edit properties of workspaces
	Manage	Private workspace	manage_private_workspace	Allows the user to create, update and delete private workspaces
	Create	View	create_view	Allows the user to create new views
	Delete	View	delete_view	Allows the use to delete views
	Edit	View	edit_view	Allows the user to edit views
	Add	View	add_view	Allows the user to add views to existing workspaces
	Remove	View	remove_view	Allows the user to remove views from an existing workspace
	Configure	Widget	configure_widget	Allows the user to access to the configuration panel of a widget
	Configure	Datasource	configure_datasource	Allows the user to select the data to analyze
	Configure	Filter	configure_filter	Allows the user to define filters on dimension for the data to analyze
	Configure	Top	configure_top	Allows the user to define top filters for the data to analyze
	Export	Data	export_data	Allows the user to export data
	Export	Report	export_report	Allows the user to export report
Launch Category Management	Browse	Launch Category	browse_launch_category	Allows the user to browse and display workspaces
	Create	Launch Category	create_ launch_category	Allows the user to create launch categories

	Delete	Launch Category	delete_ launch_category	Allows the user to delete launch categories
	Save	Launch Category	save_ launch_category	Allows the user to save or save as… launch categories
	Edit	Launch Category	edit_ launch_category	Allows the user to edit properties of launch categories
Category Management	Browse	Category	browse_category	Allows the user to browse and display workspaces
	Create	Category	create_ category	Allows the user to create workspace categories
	Delete	Category	delete_ category	Allows the user to delete workspace categories
	Save	Category	save_ category	Allows the user to save or save as… workspace categories
	Edit	Category	edit_ category	Allows the user to edit properties of workspace categories
Theme Management	Configure	Theme	configure_theme	Allows a user to modify the selected theme browsing the available list of themes.
Package Management	Browse	Package	browse_package	Allows a user to browse packages available in the platform
Add-ons Management	Browse	Layout	browse_layout	Allows a user to browse layout available in add-ons
	Browse	Widget	browse_widget	Allows a user to browse widgets available in add-ons
	Browse	Plugin	browse_plugin	Allows a user to browse plugins available in add-ons
	Browse	Menu item	browse_menu_item	Allows a user to browse menu items available in add-ons
	Browse	Menu bar	browse_menu_bar	Allows a user to browse menu bars available in add-ons
	Browse	Menu theme	browse_theme	Allows a user to browse themes available in add-ons
	Browse	Menu module	browse_module	Allows a user to browse modules available in add-ons
User Management	Browse	User	browse_user	Allows a user to browse available users on the platform (local authentication mode only)
	Create	User	create_user	Allows a user to create a new user on the platform (local authentication mode only)
	Delete	User	delete_user	Allows a user to delete an existing user on the platform (local authentication mode only)
	Edit	User	edit_user	Allows a user to edit an existing user on the platform (local authentication mode only)
Role Management	Browse	Role	browse_role	Allows a user to browse available roles on the platform
	Create	Role	create_role	Allows a user to create a new role on the platform
	Delete	Role	delete_role	Allows a user to delete an existing role on the platform
	Edit	Role	edit_role	Allows a user to edit an existing role on the platform
Platform Management	Edit	Setting	edit_setting	Allows a user to change settings on the platform
	Browse	token	browse_token	Allows a user to browse his authentication token on the platform
Launch Management	Create	Launch	create_launch	Allows a user to create new launches
	Delete	Launch	delete_launch	Allows a user to delete launches

	Edit	Launch	edit_launch	Allows a user to edit launches
	Execute	Launch	execute_launch	Allows a user to execute launches

Figure 42: RBAC –List of user interface permissions

17.10 Apache CouchDB Database

A Apache CouchDB server hosts named databases, which store documents. Each document is uniquely named in the database, and Apache CouchDB provides a RESTful HTTP API for reading and updating (add, edit, delete) database documents.

All the information related to Apache CouchDB can be found on the official web site: http://couchdb.apache.org

17.10.1 Built-in Administration

Administration can easily administrate the database using Futon, the built-in administration interface.

http:<host>:<port>/_utils

Example: http://127.0.0.1:5984/_utils

Futon provides full access to all of CouchDB's features. Futon let you create and destroy database; view and edit documents. Futon is also protected by a user/password. It is strongly recommended to use complex password to secure access direct to this database.

NOTE: this GUI database does not contains sensitive data but definitions used by the web application. So, it is recommended to enable the SSL support, do not use default user/password to increase the security level.

17.10.2 Configuration

It is possible to configure the Apache CouchDB database to use for UOC server editing the following configuration file.

<install_dir>/server/public/conf/config.json

```
...
  "database": {
    "protocol": "http",
    "host": "127.0.0.1",
    "port": "5984",
    "username": "user",
    "password": "user",
    "adminPassword": "password_of_couchdb_admin_user"
  },
...
```

It is possible to setup the protocol to use (default is http). It is strongly recommended to enable the SSL support to access to the GUI database. It is recommended to configure the TLS 1.2 to grant a secure configuration.

It is possible to configure the host, port, user and password of the Apache CouchDB user.

It is possible to specify the password of the Apache CouchDB user “admin”.

This definition is used by the UOC server only to access to the database.

17.11 Secure Socket Layer (SSL/TLS)

17.11.1 Overview

The Unified Console supports SSL/TLS in all its components to grant a high level of security and encryption. Check carefully the NodeJS version installed because TLS is only available in the latest release.

By default, secured communication across the server’s components is not enabled, since it requires first the administrator to install his platform’s certificate for the SSL authentication.

For very high security requirements, it is strongly recommended to activate SSL with authorities certificates (certified) to protect all the http channel.

IMPORTANT: HTTPS is the HTTP protocol over TLS/SSL and provided by NodeJS.
Node.js provides SSLv2 and SSLv3 protocol support by default, but these protocols are disabled. They are considered insecure and could be easily compromised as was shown by CVE-2014-3566.
It is then mandatory to use TLS v1.1 or V1.2 to have a secure solution. You can have detailed information on the NodeJS web site: https://nodejs.org/docs/latest-v0.10.x/api/tls.html#tls_protocol_support

NOTE: It is strongly recommended to use SHA-2 certificates **(256 bits)** in 2016 and encourage administrators to migrate to SHA-2 if SHA-1 certificates are still used. Most of the CA authorities will not deliver anymore SHA-1 certificates after January 2016, and some Operating System (MS Windows…) will not accept them anymore

The Unified OSS Console supports SSL encryption at several communication channel to grant a high level of security in production mode.

- UOC Server ←→ Web browser
- UOC Server ←→GUI Database
- UOC Server ←→ Identity Provider (IdP) / SSO
- UOC Server ←→Domain or data server (ex: OSS Analytics server)
- UOC Server ←→Notification Server

Certificates can be generated and configured in the platform configuration to enable the SSL mode. These certificates can be self-signed but it strongly recommended to use a trusted authority to generate appropriate trusted certificates.

NOTE: All certificates need to be copied in the <install_data_dir>/server/public/ssl.
Certificates are stored in a certificates directory on the UOC Server. These certificates can be self-signed (less secure) or trusted (recommended).

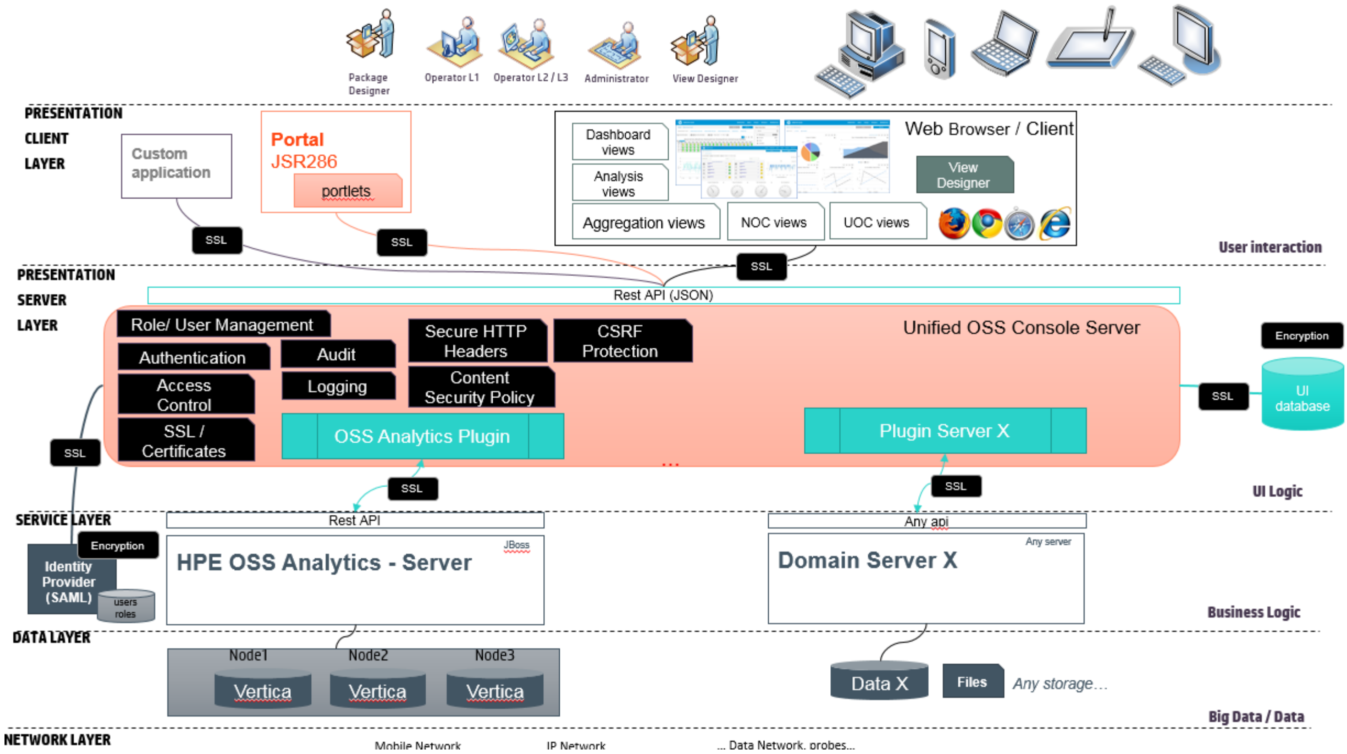


Figure 43: HPE Unified OSS Console SSL Support

17.11.2 SSL Certificates

There are 2 types of certificates:

- **Self-signed**: A self-signed certificate is an identity certificate that is signed by the same entity whose identity it certifies. This term has nothing to do with the identity of the person or organization that actually performed the signing procedure. In technical terms a self-signed certificate is one signed with its own private key.
- **Certificate Authority signed**: A digital signature from a certificate authority (CA) attests that a particular public key certificate is valid (i.e., contains correct information). Users, or their software on their behalf, check that the private key used to sign some certificate matches the public key in the CA's certificate.

IMPORTANT:
It is strongly recommended to use in production certificate provided by a Certificate authority for security reasons and uses SHA-2 algorithm.

17.11.3 Generate Certificates

Here is an example of private key and certificate request (csr) generation with command-lines.

First step is to generate a private key file, then generate a CSR (Certificate Signing Request). A CSR or Certificate Signing request is a block of encrypted text that is generated on the server that the certificate will be used on. It contains information that will be included in your certificate such as your organization name, common name (domain name), locality, and country. It also contains the public key that will be included in your certificate. A private key is usually created at the same time that you create the CSR

```
openssl genrsa -des3 -out server.key 2048
openssl req -new -key server.key -out server.csr
mv server.key server.key.org
openssl rsa -in server.key.org -out server.key
```

Second step is to generate a SSL certificate from the CSR, then obtain a certificate file from the certificate request file (csr ➔ crt).

Generate a Self-Signed SSL Certificate

```
openssl x509 -req –sha256 -days 365 -in server.csr -signkey server.key -
out server.crt
```

Generate a Authority Signed SSL Certificate

Ask your signing authority provider: submit the server.csr and get the SSL certificate server.crt

17.11.4 Self-signed Certificate Error

A self-signed certificate is still considered as unsecure, and uoc will reject by default such certificate (SELF_SIGNED_CERT_IN_CHAIN error on the console). Of course, it is recommended to provide a valid and secure certificate, i.e. generated from a CA authority, but if you really approve this certificate, you will need to explicitly indicate to UOC that this certificate is valid and should not be rejected.

To work around this, set the NODE_TLS_REJECT_UNAUTHORIZED environment variable to 0, and restart UOC.

```
Export NODE_TLS_REJECT_UNAUTHORIZED=0
uoc2 start
```

IMPORTANT: Secure CA certificates must be used in production to grant the maximal security and protection. This workaround must only be used for tests and demo purpose in very specific cases.

17.11.5 GUI Database (Apache CouchDB)

To enable SSL support in the GUI database (Apache CouchDB server), please edit <couchdb_install_dir>/etc/couchdb/local.ini

Uncomment the httpsd line and set the path to your private key and certificate file then restart your CouchDB server.

```
...

[daemons]

; enable SSL support by uncommenting the following line and supply the PEM's below.

; the default ssl port CouchDB listens on is 6984
```

```
httpsd = {couch_httpd, start_link, [https]}



[ssl]

cert_file = /path/to/ssl/files/server.crt

key_file = /path/to/ssl/files/server.key

...
```

17.11.6 UOC ←→ GUI Database

To configure SSL/TLS access between the UOC server and GUI database, please edit protocol, host and port as needed as well as ssl properties in **<install_dir>/server/public/conf/config.json.**

Certificates must be copied in **<install_dir>/server/public/ssl.**

```
...
   "database": {
      "protocol": "https",
      "host": "127.0.0.1",
      "port": "6984",
      "username": "user",
      "password": "user",
      "ssl": {
            “strictSSL”: true,
            "caCertFile": "gui_db_ca.crt",
            “secureProtocol”: “TLSv1_2_method”
      }
   },
...
```

Database security properties

property	Description
strictSSL	If true, GUI database certificate will be verified against the list of supplied certificate authorities.
caCertFile	A string or array of strings referencing trusted certificates in PEM format, located at **<install_dir>/server/public/ssl**. If this is omitted several well-known "root" certificate authorities (like VeriSign) will be used. These are used to authorize connections.
secureProtocol	The SSL method to use, e.g., TLSv1_2_method to force TLS version 1.2. The possible values depend on the version of OpenSSL installed in the environment and are defined in the constant SSL_METHODS. See: https://www.openssl.org/docs/manmaster/ssl/ssl.html#DEALING-WITH-PROTOCOL-METHODS

NOTE: When using self-signed certificates, you may encounter a ‘DEPTH_ZERO_SELF_SIGNED_CERT’ error. To skirt this error, you can set the strictSSL property to false. This must not be used in production for full security. We recommend to use a proper certificate authority in production.

17.11.7 UOC ←→ Web browser

To enable SSL/TLS support for the UOC server, please edit protocol and port as needed, as well as privateKey, certificate and secureProtocol properties in <install_dir>/server/public/conf/config.json.

Certificates and private key must be copied in <install_dir>/server/public/ssl.

```
…
"server": {
        "protocol" : "https",
        "port" : "uoc_port",
        "privateKey" : "server.key",
        "certificate" : "server.crt",
        “secureProtocol”: “TLSv1_2_method”
   },
…
```

Server security properties

property	Description
privateKey	A string referencing the UOC server private key file in PEM format, located at **<install_dir>/server/public/ssl.**
certificate	A string referencing the UOC server certificate file in PEM format, located at **<install_dir>/server/public/ssl**.
secureProtocol	The SSL method to use, e.g., TLSv1_2_method to force TLS version 1.2. The possible values depend on the version of OpenSSL installed in the environment and are defined in the constant SSL_METHODS. See: https://www.openssl.org/docs/manmaster/ssl/ssl.html#DEALING-WITH-PROTOCOL-METHODS

17.11.8 UOC ←→ Identity Provider (IdP) / SSO

To configure SSL/TLS access between the UOC server and the Identity Provider, please set the entryPoint attribute to the HTTPS entry point of your Identity Provider.

UOC server also supports signature and encryption of SAML assertions (RSA with SHA-1 or SHA-256). To enable these features, please edit **<install_data_dir>/server/public/conf/config.json.**

Certificates and private key must be copied in **<install_dir>/server/public/ssl.**

```
...
"saml": {
   "idp": {
      "entryPoint": " <idp_server_https_entry_point>",
      "identifierFormat": "urn:oasis:names:tc:SAML:2.0:nameid-format:entity",
      "certificate": "idp.crt"
   },
   "sp": {
      "issuer": "<uoc_server_full_address>",
      "privateKey": "server.key"
   },
   "signature": true,
   "encryption": true
},
...
```

See 17.7.4 How to configure SAML / SSO for the description of basic settings, only additional security properties are described below.

Identity Provider (idp) security properties

property	Description
certificate	A string referencing the Identity Provider public certificate file in PEM format, located at <install_dir>/server/public/ssl.

Service Provider (sp) security properties

property	Description
privateKey	A string referencing the UOC server private key file in PEM format, located at <install_dir>/server/public/ssl.

SAML security properties

property	Description
signature	If true, SAML assertions will be digitally signed. SHA-1 and SHA-256 algorithms are currently supported. Requires privateKey and certificate properties to be set.
encryption	If true, SAML assertion will be encrypted using RSA algorithm. Requires privateKey property to be set.

17.11.9 UOC ←→Domain or data server (ex: OSS Analytics server)

To configure SSL/TLS access between the UOC server and the OSSA server(s), please edit protocol, host and port as needed as well as ssl properties in <install_dir>/server/public/conf/config.json.

Certificates must be copied in <install_dir>/server/public/ssl.

```
{
    "servers": {
        "ossa": {
            "protocol": "https",
            "host": "<ossa_server_full_hostname_or_ip_address>",
            "port": "<ossa_server_port>",
            "ssl": {
               “strictSSL”: “true”,
               "caCertFile": "ossa_server_ca.crt",
               “secureProtocol”: “TLSv1_2_method”
            }
        },
        ....
    }
}
```

OSSA server security properties

property	Description
strictSSL	If true, OSSA server certificate will be verified against the list of supplied certificate authorities.
caCertFile	A string or array of strings referencing trusted certificates in PEM format, located at **<install_dir>/server/public/ssl**. If this is omitted several well-known "root" certificate authorities (like VeriSign) will be used. These are used to authorize connections.
secureProtocol	The SSL method to use, e.g., TLSv1_2_method to force TLS version 1.2. The possible values depend on the version of OpenSSL installed in the environment and are defined in the constant SSL_METHODS. See: https://www.openssl.org/docs/manmaster/ssl/ssl.html#DEALING-WITH-PROTOCOL-METHODS

17.11.10 UOC ←→Notification server (Redis server)

To configure SSL/TLS access between the UOC server and the notification server, please edit host, port and password as needed as well as ssl properties in **<install_dir>/server/public/conf/config.json.**

Certificates must be copied in **<install_dir>/server/public/ssl.**

NOTE: Redis does not support TLS/SSL natively. To enable TLS/SSL support, a SSL tunnel must be configured on top of Redis. See 9.3.4.4 SSL tunneling

```
{...
  "notifications": {
    "publishSubscribeServer": {
      "active": true,
      "host": ""<notification_server_full_hostname_or_ip_address>","
      "port": "<notification_server_port>",
      "password": "<notification_server_password>",
      "ssl": {
        "privateKey": "server.key",
        "certificate": "server.crt",
        "strictSSL": true,
        "caCertFile": ["CA.crt"],
        "secureProtocol": "TLSv1_2_method"
      }
    }
  }...
}
```

Notification server security properties

property	Description
privateKey	A string referencing the UOC server private key file in PEM format, located at <install_dir>/server/public/ssl.
certificate	A string referencing the UOC server certificate file in PEM format, located at <install_dir>/server/public/ssl.
strictSSL	If true, Redis certificate will be verified against the list of supplied certificate authorities.
caCertFile	A string or array of strings referencing trusted certificates in PEM format, located at **<install_dir>/server/public/ssl**. If this is omitted several well-known "root" certificate authorities (like VeriSign) will be used. These are used to authorize connections.

secureProtocol	The SSL method to use, e.g., TLSv1_2_method to force TLS version 1.2. The possible values depend on the version of OpenSSL installed in the environment and are defined in the constant SSL_METHODS. See: https://www.openssl.org/docs/manmaster/ssl/ssl.html#DEALING-WITH-PROTOCOL-METHODS

17.12 Content Security Policy

Content Security Policy (CSP) is a defense based on the W3C specification offering the possibility to instruct the web browser from which location some type of resources are allowed to be loaded.

It helps to mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks, leading to data theft, malware injection etc. Its main goal is to prevent anything unintended being injected into our page. This can include frames, images, tracking scripts, and opening the door to XSS vulnerabilities.

CSP works by setting whitelists in the Content-Security-Policy response header to define sources of trusted content. The browser is then only allowed to execute or render resources from those trusted sources. This means that if someone were to successfully inject their script into your page, the browser would not execute it as the source would not be in the whitelist.

The main types of resources that can be controlled with CSP are:

- JavaScript
- Stylesheets
- Images
- AJAX, WebSockets, etc.
- Fonts
- Plugins
- HTML5 Media Elements
- Frames

This standard is now widely supported by the modern web browsers; browsers that do not support CSP simply ignore it, functioning as usual. Please check your browser with the online Content Security Policy Browser Tests at https://content-security-policy.com/browser-test

UOC allows a secure configuration for each resource type, according to the CSP specification.

This policy can be configured in **<install_dir>/server/public/conf/config.json**.

```
{
            ...
            "csp": {
                        "defaultSrc": ["'self'"],
                        "styleSrc": ["'self'", "'unsafe-inline'"],
                         "scriptSrc": ["'self'", "'unsafe-eval'"],
                        "imgSrc": ["'self'", "data:"],
                         "childSrc": ["*"],
                         "frameSrc": ["*"],
                         "baseUri": ["'self'"],
                         "formAction": ["'self'"]
   }
}
```

CSP directives are defined under the “csp” attribute. Attribute names are standard CSP attributes written in snake case.

Directive	Description
defaultSrc	This is the default policy for loading all content. Whatever is defined here applies to all the other type unless you set them to 'none'. In our case we are saying that any content coming from the same origin is allowed.
scriptSrc	This is the policy for controlling the valid sources of JavaScript. In our case we are setting *.google-analytics.com as an allowed source for scripts. This is in addition to the same origin rule we defined in defaultSrc.
styleSrc	This is the policy for controlling the valid sources of stylesheets. What we are doing here is using the 'unsafe-inline' keyword to allow inline stylesheets. If this was not set, any inline stylesheets would not work.
imgSrc	This is the policy for controlling the valid sources of images. Just like scriptSrc, we are setting the allowed domain for images.
connectSrc	This is the policy for controlling the valid sources of AJAX, WebSockets, or EventSource. In this case, we are using the 'none' keyword to state that no content of this type should be allowed. This overrides the defaultSrc directive.
fontSrc	This is the policy for controlling the valid sources of fonts. It is blank to state the only whitelist to use is the one defined in defaultSrc.
objectSrc	This is the policy for controlling the valid sources of plugins like <object>, <embed>, or <applet>. It is blank to state the only whitelist to use is the one defined in defaultSrc.
mediaSrc	This is the policy for controlling the valid sources of HTML5 media types like <audio> or <video>. It is blank to state the only whitelist to use is the one defined in defaultSrc.
frameSrc	This is the policy for controlling the valid sources of frames. It is blank to state the only whitelist to use is the one defined in defaultSrc. There are four special keywords you can use when defining whitelists: ● 'none' will match nothing ● 'self' will match the current origin but not subdomains ● 'unsafe-inline' allows inline JavaScript and CSS ● 'unsafe-eval' allows things like eval() to work

For more information, please visit the Content Security Policy website at https://content-security-policy.com/ and the official specification at http://www.w3.org/TR/CSP2/.

17.13 Cross-Site Request Forgery

Cross-Site Request Forgery (CSRF) is an attack that impersonates a trusted user in order to send requests of the attacker's choosing. CSRF attacks specifically target state-changing requests; the attacker, in this case, has no way to see the response to the forged request.

The Cross-Site Request Forgery protection is enabled by default. If you want to perform requests on the UOC2 server from outside the GUI, this parameter must be taken into account.

To mitigate this risk, per default UOC2 generates a CSRF token at first request. This token is stored in a regular cookie and embedded in the headers of all the following HTTP requests done to the server from the client side.

Server side, the CSRF token validity is checked before performing a POST/PUT/DELETE request. If the token does not exist or is not valid, the request is aborted.

The CRSF token in stored in a regular cookie named XSRF-TOKEN.

This security protection can be turned on/off in <install_dir>/server/public/conf/config.json.

```
{
    ...
    "crsf": {
        “enable”: true | false
        ...
    }
}
```

disableCrsfProtection property can also be used for this purpose (DEPRECATED)

```
{
    ...
    "disableCrsfProtection": false | true
}
```

By default, all POST/PUT/DELETE routes are protected, including plugin ones. To disable this protection only for some routes that may be called outside the web browser, you can use the property crsf.ignoreRoutes.

```
{
    ...
    crsf: {
        ...
        “ignoreRoutes”: [
            //list all routes to be ignored from crsf protection (e.g. “/V1.0/domains/plugin_simulator/notify”)
        ]
    }
}
```

17.14 X-Frame-Options

X-Frame-Options is a HTTP response header indicating whether or not a browser is allowed to render a page in a <frame>, <iframe> or <object>. Goal is to mitigate the risk of clickjacking attacks controlling if and where your page can be put into a <frame> or <iframe>

This header can be used to avoid clickjacking attacks by ensuring that site contents are not embedded into other sites.

NOTE: frame-ancestors directive (Content Security Policy) can also be used for this purpose but it is not supported by all major browsers yet, contrary to X-Frame-Options header.

Use of X-Frame-Options headers can be turned on/off in <install_dir>/server/public/conf/config.json

```
{
        …
        “X-Frame-Options”: {
                “enable”: true | false,
                “action”: “deny” | “sameorigin” | “allow-from”,
                “domain”: “your authorized domain”
        }
}
```

The property “action” can take 3 values:

action	Description
deny	Prevents any domain from framing UOC. This is the recommended value if you do not have any framing needs
sameorigin	Allows UOC to be displayed in a frame having the same origin as UOC
allow-from	Allows UOC to be displayed for a specific domain

Removing the property X-Frame-Options from the configuration disables the use of X-Frame-Options header.

IMPORTANT: allow-form is not well supported by all major browsers. In this case, it is better to use Content Security Policy (frame-ancestors).

17.15 HTTP Strict Transport Security

HTTP Strict Transport Security (HSTS) is an opt-in security enhancement that is specified by a web application through the use of a special response header. It allows a web server to tell user agents to interact with it in the future only over HTTPS. It controls it by defining a period of time with the Strict-Transport-Security response header.

This is great if you are running an HTTPS site but still have an HTTP endpoint in order to provide redirection to the HTTPS endpoint. This can help reduce the chance of Man-In-The-Middle (MITM) attacks by reducing the frequency of requests being made over insecure channels.

Once a supported browser receives this header that browser will prevent any communications from being sent over HTTP to the specified domain and will instead send all communications over HTTPS. It also prevents HTTPS click through prompts on browsers.

HSTS addresses the following threats:

- User bookmarks or manually types http://example.com and is subject to a man-in-the-middle attacker
  - HSTS automatically redirects HTTP requests to HTTPS for the target domain
- Web application that is intended to be purely HTTPS inadvertently contains HTTP links or serves content over HTTP
  - HSTS automatically redirects HTTP requests to HTTPS for the target domain
- A man-in-the-middle attacker attempts to intercept traffic from a victim user using an invalid certificate and hopes the user will accept the bad certificate
  - HSTS does not allow a user to override the invalid certificate message

IMPORTANT: There is still a window where a user is vulnerable. Before accessing a site the first time, the browser does not know yet what protocol to use between HTTP and HTTPS.
To eliminate this window, browsers embed a HSTS preload list, a list of domains to be accessed using HTTPS. If you want to submit your domain to this list, please visit https://hstspreload.appspot.com/, maintained by Chrome and used by Internet Explorer, Microsoft Edge, Firefox and Safari.

HTTP Strict Transport Policy can be turned on/off in **<install_dir>/server/public/conf/config.json**.

```
{
        ...
        “hsts”: {
                “enable”: true | false,
                “maxAge”: “365d”,
                “includeSubDomains”: “true” | “false”,
                “preload”: “true” | “false”
        }
}
```

property	Description
maxAge	indicates how long HTTPS must be used for the next period of time.
includeSubDomains	indicates if HSTS should be used also for subdomains.
preload	indicates that you consent to use HSTS preload list for your domain, but you still have to submit your domain to the list.

Removing the property hsts from the configuration disables the HTTP Strict Transport Policy.

17.16 UOC Logging

The UOC Server has customizable logger to ease troubleshooting of the UOC Platform. See 19.1 Logging for detailed information to tune correctly the logs.

IMPORTANT: These logs are only accessible to platform administrator for troubleshooting reason and may content sensitive information related to the privacy, especially if the debug level is enabled

TIP: As UOC uses a standard logging system (log4js). It is possible by configuration to redirect these logs to a centralized system to troubleshooting or detect security breach. Please refer to the official log4js web site to get detailed information (https://github.com/nomiddlename/log4js-node)

17.17 UOC Auditing

The UOC server logs special activities and access to UOC resources to ease auditing. It is recommended to tune correctly the UOC audit log policy (file size, rollup files…) to fit your needs.

The UOC platform provides several security audit logs:

- UOC Sessions (login. Logout)
- UOC Resources (accesses and operations)

IMPORTANT: These logs are only accessible to platform administrator and may content sensitive information related to the privacy (IP address…)

TIP: As UOC uses a standard logging system (log4js). It is possible by configuration to redirect these audit logs to a centralized system to troubleshooting or detect security breach. Please refer to the official log4js web site to get detailed information (https://github.com/nomiddlename/log4js-node)

17.17.1 UOC Sessions

All information related to the UOC user session (login, logout…) are stored in a log file in the server at <install_data_dir>/logs/sessions.log

It is possible to see user id, date/time, and IP address to ease the audit.

Example of sessions audit:

```
..

2015-05-09 16:23:46.692 - DUPONT3 - admin has logged out (127.0.0.1)

2015-05-09 16:23:53.796 - DUPONT3 - operator_l1 has logged in (127.0.0.1)

2015-05-09 16:24:40.998 - DUPONT3  - Paul has logged out (127.0.0.1)

2015-05-09 16:24:45.886 - DUPONT3 - admin has logged in (127.0.0.1)

2015-05-09 17:08:45.040 - DUPONT3 – Lilian  has logged in (127.0.0.1)

...
```

It is possible to refine the logging policy for this sessions logs customizing the “session-logger “ appender in the file : <install_data_dir>/server/public/conf/log4js.json

Only log level information is supported but you can customize the format on the line (layout pattern).

Default logging policy:

```
{
    "type": "file",
    "filename": "logs/sessions.log",
                         "level": "INFO",
                         "maxLogSize": 2048000,
                         "backups": 3,
    "category": "sessions-logger",
        "layout": {
              "type": "pattern",
              "pattern": "%d{ISO8601} - %h - %m"
        }
```

17.17.2 UOC Resource access

All UOC resource accesses, operations, applications and data requests are stored in a log file in the server at <install_data_dir>/logs/security.log

It tracks all UOC actions done by a user on resources with date/time, IP address, user id … to ease the audit.

Only log level information is supported but you can customize the format on the line (layout pattern).

Example of security audit:

```
..

2015-05-02 18:43:33.970 - DUPONT3 - (Node Server) : WIDGET : Browse all widgets (path:client/addons//vodafone/widgets)

2015-05-02 18:43:34.094 - DUPONT3 - (Node Server) : PLUGIN : Browse workspaces for plugin cea

2015-05-02 18:43:57.177 - DUPONT3 - (Node Server) : PLUGIN : Browse all plugins (path:server/addons/plugins/)

2015-05-02 18:43:57.181 - DUPONT3 - User: admin (127.0.0.1) : CATEGORY : Browse all categories defined for workspaces

2015-05-02 18:44:06.390 - DUPONT3 - User: admin (127.0.0.1) : WORKSPACE : Workspace DataFormatting_TimeKpiSeries has been opened

2015-05-02 18:44:06.437 - DUPONT3 - User: admin (127.0.0.1) : VIEW : View DataFormat_TimeKpi has been opened

2015-05-02 18:44:06.498 - DUPONT3 - User: admin (127.0.0.1) : LAYOUT : Layout layout-1-1-1 has been accessed

2015-05-02 18:44:06.536 - DUPONT3 - User: admin (127.0.0.1) : WIDGET : Widget hp-time-selector has been accessed

2015-05-02 18:44:31.250 - DUPONT3 - User: admin (127.0.0.1) : PLUGIN : Get data for plugin ossa with the url
http://abc.hp.com:8080/ossa/packages/MBBQOE_Trial/facts/volume_up_sum/volume_down_sum/dims/BRAND/timewindow/1405505700000/1405689300000?b=1&granularity=15

2015-05-02 18:54:29.892 - DUPONT3 - User: admin (127.0.0.1) : PLUGIN : Get data for plugin ossa with the url
http://abc.hp.com:8080/ossa/packages/MBBQOE_Trial/facts/volume_up_sum/volume_down_sum/dims/BRAND/timewindow/1405505700000/1405689300000?b=1&granularity=15&top=5

…
```

It is possible to refine the logging policy for these session logs customizing the “session-logger “ appender in the file : <install_data_dir>/server/public/conf/log4js.json

Only log level information is supported but you can customize the format on the line (layout pattern).

```
Default logging policy:

{
        "type": "file",
        "filename": "logs/security.log",
        "level": "INFO",
        "maxLogSize": 2048000,
        "backups": 3,
        "category": "security-logger",
               "layout": {
                     "type": "pattern",
                     "pattern": "%d{ISO8601} - %h - %m"
           }
```

17.18 Security Alerts

It is recommended to follow up the security alerts by subscribing to dedicated newsletters.

Do not forget that NodeJS and CouchDB are pre-requisite not supported by HPE, and you are in charge of updating these products if needed. Please go to official web sites for details, documentation, patches, updates, security alerts… it is strongly recommended to subscribe to Node Security news to monitor the security issues

Many security web sites can help you like:

- NodeJS: https://nodejs.org/en/security/ and http://blog.nodejs.org
- NodeSecurity Project (https://nodesecurity.io/)
- Node.js Security Twitter: https://twitter.com/nodesecurity
- Google Group : http://groups.google.com/group/nodejs-sec

IMPORTANT: The customer is responsible for installing OS Security patches to secure the Unified OSS Console products.

Chapter 18
Migration

This chapter describes the changes that have been integrated in the last version of UOC and may generate small changes on existing solution especially the impact of the HPE rebranding of Unified Console.

18.1 Migration V2.2 to V2.3

The HPE Unified Console V2.3 is backward fully compatible with the V2.2.x but the CouchDB format has been enhanced to support new feature (Private workspace). So, it require a migration step to migrate an existing database to the new format.

IMPORTANT: You must use the migrate script (migrate_v2.2_to_v2.3.sh) to migrate the CouchDB.
A validation check has been added at startup of UOC V2.3 to validate the CouchDB has been migrated correctly. If you forgot this step or if the couchDB format is nto correct, UOC server will stop and display an error message.

NOTE: Because all the data of the previous installation is kept, you may need to add manually the new options.

18.1.1 Unified OSS Console Kits

Unified OSS Console V2.2 kits contains 2 rpm kits:

- HPE Unified OSS Console V2.2
- HPE Unified OSS Console V2.2 – Add-on OSS Analytics V1.1

Unified OSS Console V2.3 kits contains 1 rpm kits:

- HPE Unified OSS Console V2.3.

IMPORTANT: The OSSA Add-on kit has been removed from UOC V2.3.

The OSS Analytics add-ons has been moved to the add-on Assurance Monitoring (UOC AM)

The main changes on the packaging of UOC V2.3 is related to the user rights management.

This kit will require to create group: uoc, couchdb, redis and non-login user to limit privileges of all processes and ease auditing and tracking of user ‘s operations. This enhancement allows an administrator to fine tune the sudo commands.

18.1.2 Apache CouchDB Migration

Please refer to the installation steps described in 6.5.2 Migrate an existing UOC V2.2 to UOC V2.3 installation

The CouchDB administrator (user identifier / password) has been removed from the configuration file (database) to enhnce the security and avoid anyoneto find out the admin account of the CouchDB.

See <install_dir>/server/public/conf/config.json

```
     "database": {
…
~~        "adminUsername": "admin",~~
~~        "adminPassword" : "admin"~~
     },
```

All Script that requires these user/password information will ask for them in an interactive way, before executing the script (e.g. dbinit , etc.) for security reasons.

18.1.3 UOC does not deliver anymore OSSA Add-on

The OSS Analytics add-ons has been moved to the Assurance Monitoring (UOC AM) and will be integrated with OSS Monitoring Add-on. (OSSM).

18.1.3.1 Deprecated OSSA Widgets

In UOC V2.3, several widgets installed by the specific Add-on OSS Analytics (OSSA) has been removed and are deprecated.

Widget Identifier	Name
ossa-ccd	OSSA Customer Care Dashboard widget
ossa-session-table	OSSA Session Table widget

Figure 44: Deprecated OSSA widgets in UOC V2.3

18.1.3.2 Deprecated OSSA Layout

The Add-on Layout ccd is deprecated and has been removed.

18.1.4 Configuration Files

Some configurations files have been changed or added in the UOC V2.3:

18.1.4.1 Updated Configuration File

The configuration file has been updated to integrate new features and improvement.

Due to the last Comprehensive Applications Threat Analysis review, UOC integrates several enhancement related to increase the security and mitigates security attacks. All security reports have been analyzed and defect and enhancements implemented.

All these security enhancement have been enabled by default in the latest configuration file <install_dir>/server/public/conf/config.json

IMPORTANT: In case of migration from UOC V2.2, you will need to copy or merge new options manually. All your previous configuration file is kept. <install_dir>/server/public/conf/config.json

A typical error you may have if you do not upgrade your config.json is related to Content Security Protection that will block your WebSocket connection. The new config.sjon define a policy to accept these connection to support real-time notification from the server.

```
Refused to connect to 'ws://my.server.com:3000/socket.io/?EIO=3&transport=websocket&sid=SLX2Xdm_sjWRIDvIAAAA'
because it violates the following Content Security Policy directive: "default-src 'self'". Note that 'connect-src' was not
explicitly set, so 'default-src' is used as a fallback.

Uncaught SecurityError: Failed to construct 'WebSocket': Refused to connect to
'ws://my.server:3000/socket.io/?EIO=3&transport=websocket&sid=SLX2Xdm_sjWRIDvIAAAA' because it violates the
document's Content Security Policy.
```

18.1.4.2 New Platform Monitoring Configuration File

A new file defines if the platform monitoring feature is enabled/disabled (default is disabled).

IMPORTANT: In case of migration from UOC V2.2, you will need to copy this file manually. All your previous configuration file is kept.
<install_dir>/server/public/conf/platform-monitoring.json

18.1.5 Data Definition Files

18.1.5.1 New Notifications Definition File

The real-time notifications is a key feature of the UOC V2.3 and new notification file definition has been added in the installation.

IMPORTANT: In case of migration from UOC V2.2, you will need to create this directory and copy this file manually. <install_dir>/data/notifications

1. Create directory <install_dir>/data/notifications
2. Copy the notifications definitions in <install_dir>data/notifications/notifications.json

18.1.5.2 Updated Permissions Definition File

The list of available permissions have been updated to add new one to manage the private workspace.

IMPORTANT: In case of migration from UOC V2.2, you will need to copy this file manually. All your previous configuration file is kept.
<install_dir>/data/permissions/permissions.json

18.1.5.3 Updated Roles Definition File

The list of available roles have been updated to add new one to manage the private workspace and dynamic reload of packages.

IMPORTANT: In case of migration from UOC V2.2, you will need to copy this file manually. All your previous configuration file is kept.
<install_dir>/data/roles/roles.json

18.1.5.4 Updated Category Definition File

The list of available category have been updated to add new one to manage ‘HPE-Template’ category.

IMPORTANT: In case of migration from UOC V2.2, you will need to copy this file manually. All your previous configuration file is kept.
<install_dir>/data/categories/categories.json

18.1.6 HPE Template Files

UOC V2.3 integrates pre-defined templates (views and workspaces), ready to be used: Notification Center Views, Data Exchange Inspector View, Localization Inspector View, etc.

HPE Template Workspace

- <install_dir>/data/workspaces/HPE-Templates-Notifications-Center.json

HPE Template Views

- <install_dir>/data/views/ HPE-Templates_Data_Exchange_Inspector.json
- <install_dir>/data/views/ HPE-Templates_Localization_Inspector.json
- <install_dir>/data/views/ HPE-Templates_Notification_Center.json
- <install_dir>/data/views/ HPE-Templates_Notification_Slider.json
- <install_dir>/data/views/ HPE-Templates_Notifications_Generator.json

18.1.7 Add-on and Custom Development

18.1.8 Add-on Plugin

If you are owner of a custom plugin in your add-on, there are additional RESTAPI to support that have been integrated in our current plugin code .generator. You may need to support them if you want to access to these features.

18.1.8.1.1 Get Plugin Health Status

Plugins can provide a REST API to indicate their health status and also be able to customize this api to check all the critical components linked to the plugin (database, external data servers …).

This API returns a simple response ‘OK’ that can be easy to parse for a load balancer or any external tools to manage the high availability at plugin level.

Description	
Verb	GET
Path	/V1.0/domains/*<domainId>*/check
Description	Return the health status 'OK'
Response Status Code	200 – OK

18.1.8.1.2 Reload dynamically all packages

The plugin provides a RESTAPI to perform a dynamic reload of all packages associated to a plugin. It will up to date the list of packages available for Unified Console.

Description	
Verb	GET
Path	V1.0/domains/*<domainId>*/reload
Description	Refresh dynamically all the packages associated to the plugin
Response Status Code	200 – OK
Response Object	List of reloaded packages

NOTE: This operation may impact the performance of the platform as it may need a little time to be fully completed

18.1.8.1.3 Reload dynamically a package

Description	
Verb	GET
Path	V1.0/domains/*<domainId>*/packages/<packageId>/reload
Description	Refresh dynamically a package using its unique identifier
Response Status Code	200 – OK

NOTE: This operation may impact the performance of the platform as it may need a little time to be fully completed

18.1.9 Add-on Clients

The UOC V2.3 embed a lot of changes regarding the version of open sources. You can find in the following section, details on these open sources and their official web site to get more information or follow their features and fixes.

All your custom development will need to be tested on these new version and there are some breaking changes in:

- Angular Bootrap: This open source prefixed now all its components with '**uib-**'.

Example: **datepicker** has been renamed **uib-datepicker.**

See detail in their migration guide: https://github.com/angular-ui/bootstrap/wiki/Migration-guide-for-prefixes

- Angular Cache:
  - DSCacheFactory renamed to CacheFactory
  - Module dependency 'angular-data.DSCacheFactory' renamed 'angular-cache'

See details in their change log: https://github.com/jmdobry/angular-cache/blob/master/CHANGELOG.md

IMPORTANT: UOC V2.3 did not use always latest open sources. Some of them still have major issues or pending defect that we need to be fixed before integrating them in UOC:

These open sources have open issues:

- Nano
- Angular bootstrap
- phantomJS

NOTE: Please check the detailed list of open sources. See 18.1.10 Open Sources Migration

18.1.10 Open Sources Migration

The UOC V2.3 embed a lot of changes regarding the version of open sources. You can find in the following section, details on these open sources and their official web site to get more information or follow their features and fixes.

18.1.10.1 Server Open Sources

These open sources are integrated in the UOC V2.3

Open Source	Version	License	Web Site
async	2.0.1	Copyright (c) 2010-2016 Caolan McMahon	https://github.com/caolan/async
body-parser	1.15.2	MIT license	https://github.com/expressjs/body-parser
compression	1.6.2	MIT license	https://github.com/expressjs/compression
cookie-parser	1.4.3	MIT license	https://github.com/expressjs/cookie-parser
cron	1.1.0	MIT license	https://github.com/ncb000gt/node-cron
csurf	1.9.0	MIT license	https://github.com/expressjs/csurf
csvtojson	1.0.0	Copyright (C) 2013 Keyang Xiang	https://github.com/Keyang/node-csvtojson
ejs	2.5.1	MIT license	http://embeddedjs.com/
express	4.14.0	MIT license	http://expressjs.com/
express-jwt	3.4.0	MIT license	https://github.com/auth0/express-jwt
express-validator	2.20.8	MIT license	https://github.com/ctavan/express-validator
frameguard	2.0.0	MIT license	https://github.com/helmetjs/frameguard
helmet	2.1.2	MIT license	https://github.com/helmetjs/helmet
helmet-csp	1.2.2	MIT license	https://github.com/helmetjs/csp
Hsts	1.0.0	MIT license	https://github.com/helmetjs/hsts
jsonschema	1.1.0	MIT license	https://github.com/tdegrunt/jsonschema
jsonwebtoken	7.1.9	MIT license	https://github.com/auth0/node-jsonwebtoken

lodash	3.10.1	Copyright 2012-2014 The Dojo Foundation	https://github.com/lodash/lodash
log4JS	0.6.38	Apache 2.0 License	https://github.com/nomiddlename/log4js-node
moment	2.14.1	Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors	https://github.com/moment/moment
moment-duration-format	1.3.0	MIT license	https://github.com/jsmreese/moment-duration-format
ms	0.7.1	MIT license	https://github.com/rauchg/ms.js
Msgpack-js	0.3.0	MIT license	https://github.com/creationix/msgpack-js
nano	6.1.5	Apache 2.0 License	https://github.com/dscape/nano
nconf	0.8.4	Copyright (c) 2011 Nodejitsu Inc.	https://github.com/flatiron/nconf
passport	0.3.2	MIT license	http://passportjs.org/
passport-local	1.0.0	MIT license	https://github.com/jaredhanson/passport-local
passport-saml	0.15.0	MIT license	https://github.com/bergie/passport-saml
pdfkit	0.8.0	MIT license	https://github.com/devongovett/pdfkit
phantomjs	1.9.19	MIT license	https://github.com/ariya/phantomjs/
q	1.4.1	MIT license	https://github.com/kriskowal/q
Redis	2.6.2	BSD license	https://github.com/antirez/redis
request	2.74.0	Apache 2.0 License	https://github.com/mikeal/request
socket.io	1.4.8	MIT license	https://github.com/socketio/socket.io
socket.io-adapter	0.4.0	MIT license	https://github.com/socketio/socket.io-adapter
socketio-jwt	4.5.0	MIT license	https://github.com/auth0/socketio-jwt
Uid2	0.0.3	MIT license	https://github.com/coreh/uid2
xml2js	0.4.17	MIT license	https://github.com/Leonidas-from-XIV/node-xml2js

Figure 45: Open Sources integrated in UOC V2.3 (Server)

18.1.10.2 Client Open Sources

These open sources are integrated in the UOC server and will run in the web browser (client).

Open Source	Version	License	Web Site
Angular	1.5.8	MIT license	http://angularjs.org/
angular-ui-ace	0.2.3	MIT license	https://github.com/angular-ui/ui-ace

ace-builds	1.2.3	Copyright (c) 2010, Ajax.org B.V. All rights reserved.	https://github.com/ajaxorg/ace-builds
angular-bootstrap	1.1.2	MIT license	https://github.com/angular-ui/bootstrap
angular-cache	4.6.0	MIT license	https://github.com/jmdobry/angular-cache
angular-cookies	1.5.8	MIT license	https://github.com/Elzair/angular-module-cookies
angular-dynamic-locale	0.1.32	MIT license	https://github.com/lgalfaso/angular-dynamic-locale
Angular-leaflet-directive	0.10.0	MIT license	https://github.com/tombatossals/angular-leaflet-directive
angular-resource	1.5.8	MIT license	https://github.com/roylines/node-angular-resource
angular-route	1.5.8	MIT license	https://github.com/Elzair/angular-module-route
angular-sanitize	1.5.8	MIT license	https://github.com/Elzair/angular-module-sanitize
angular-schema-form	0.8.13	MIT license	https://github.com/Textalk/angular-schema-form
angular-translate	2.11.1	MIT license	https://github.com/angular-translate/angular-translate
angular-translate-loader-partial	2.11.1	MIT license	https://github.com/angular-translate/bower-angular-translate-loader-partial
angular-tree-control	0.2.28	MIT license	https://github.com/wix/angular-tree-control
angular-ui-grid	3.2.1	MIT license	https://github.com/angular-ui/ng-grid/blob/master/LICENSE.md
bootstrap	3.3.7	MIT license	http://getbootstrap.com/
checklist-model	0.10.0	MIT license	http://vitalets.github.io/checklist-model
es5-shim	4.5.9	MIT license	https://github.com/es-shims/es5-shim
font-awesome	4.6.3	CC-BY-3.0, MIT	https://github.com/FortAwesome/Font-Awesome
Jasny-bootstrap	3.1.3	Apache License 2.0	https://github.com/jasny/bootstrap
jquery	3.1.0	Copyright 2005, 2014 jQuery Foundation and other contributors, https://jquery.org/	
jquery-knob	1.2.13	MIT license	https://github.com/aterrien/jQuery-Knob
jquery-ui	1.12.0		https://jqueryui.com/

javascript-state-machine	2.3.5	Copyright (c) 2012, 2013, 2014, 2015, Jake Gordon and contributors	https://github.com/jakesgordon/javascript-state-machine
JointJS	0.9.7	Mozilla Public License Version 2.0	http://jointjs.com https://github.com/clientIO/joint
json3	3.3.2	MIT license	http://bestiejs.github.io/json3/
highcharts-ng	0.0.12	MIT license	https://github.com/pablojim/highcharts-ng
requireJS	2.2.0	MIT license / BSD	http://requirejs.org/
Leaftlet	1.0.0-rc.3	Copyright (c) 2010-2016, Vladimir Agafonkin Copyright (c) 2010-2011, CloudMade All rights reserved.	https://github.com/Leaflet/Leaflet
Leaftlet.markercluster	1.0.0-rc.1	MIT license	https://github.com/Leaflet/Leaflet.markercluster
Leaflet-plugins	1.9.1	Copyright (c) 2011-2015, Pavel Shramov, Bruno Bergot	https://github.com/shramov/leaflet-plugins
lodash	3.10.1	Copyright 2012-2014 The Dojo Foundation	https://github.com/lodash/lodash
lodash-deep	1.6.0	MIT license	https://github.com/marklagendijk/lodash-deep
moment	2.14.1	Copyright (c) 2011-2016 Tim Wood, Iskren Chernev, Moment.js contributors	https://github.com/moment/moment
ng-idle	1.2.2	MIT license	https://github.com/HackedByChinese/ng-idle
ng-tags-input	3.1.1	MIT license	https://github.com/mbenford/ngTagsInput
require-css	0.1.8	MIT license	https://github.com/guybedford/require-css
requirejs-plugins	1.0.3	MIT license	https://github.com/millermedeiros/requirejs-plugins
proj4	2.3.15	MIT license	https://github.com/OSGeo/proj.4
text	2.0.15	Copyright jQuery Foundation and other contributors, https://jquery.org/	https://github.com/requirejs/text
socket.io-client	1.4.8	MIT license	https://github.com/socketio/socket.io-client
slickgrid	2.3.2	Copyright (c) 2009-2016 Michael Leibman,	https://github.com/mleibman/SlickGrid https://github.com/6pac/SlickGrid
Summernote	0.8.2	MIT license	http://summernote.org/ https://github.com/summernote/summernote

Figure 46: Open Sources integrated in UOC V2.3 (Client)

18.1.11 Client Widget Icons

UOC V2.3 integrates new components and their associated icons.

IMPORTANT: In case of migration from UOC V2.2, you will need to copy these files manually. All your previous icons file are kept.

See <install_dir>/client/images/widgets and copy all the hpe-* files.

Chapter 19
Troubleshooting

19.1 Logging

The UOC Server has customizable logger to ease troubleshooting of the UOC Platform.

IMPORTANT: These logs are only accessible to platform administrator for troubleshooting reason and may content sensitive information related to the privacy, especially if the debug level is enabled.

19.1.1 Server logs

All logs for the UOC server can be found under **<install_data_dir>/logs/server.log**

It is possible to refine the logging policy for this server logs customizing the “server-logger “ appender in the file : **<install_data_dir>/server/public/conf/log4js.json**

This log file supports multiple level of log (info, warning, error, warning, debug). Default level is warning and you and customize the format on the line (layout pattern).

Any changes on this file will be dynamically apply without restart the UOC server.

Default logging policy:

```
    {
        "type": "file",
        "filename": "logs/server.log",
        "level": "WARN",
        "maxLogSize": 2048000,
        "backups": 3,
        "category": "server-logger"
    }
```

19.1.2 Http requests

All http requests done by clients (web browsers) are logged in the UOC server in the logging directory.

You can find this log under **<install_data_dir>/logs/http.log**

It is possible to refine the logging policy for this server logs customizing the “server-logger “ appender in the file : **<install_data_dir>/server/public/conf/log4js.json**

Default log level is warning and customize the format on the line (layout pattern).

Default logging policy:

```
{
            "type": "file",
            "filename": "logs/http.log",
            "level": "WARN",
            "maxLogSize": 2048000,
            "backups": 3,
            "category": "express-logger"
}
```

19.2 Web Browser Console

If you have issues with the web browser, please check carefully that you are using a supported web navigator. HTML 5 is only supported in very last web browser versions.

There is no logging at the client side and no persistence of console messages. If the problem is persistent, it is recommended to help the support team by opening the web browser console and check any unexpected message (error, warning…). This console can be visible usually by pressing F12 (check your navigator documentation for details).

NOTE: If you notice unexpected behavior after changes done at the UOC Server, It is strongly recommended to clear the web browser cache and restart from a fresh web browser usage.

19.3 Troubleshooting UOC Servers

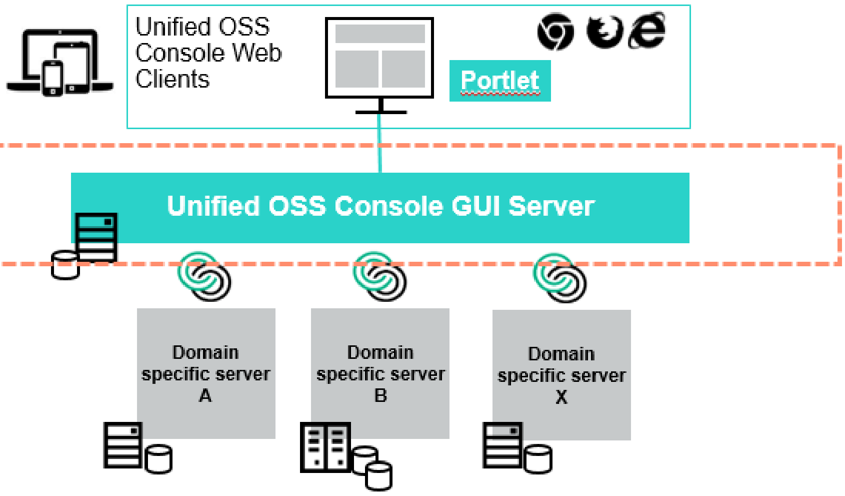


Figure 47: Troubleshooting UOC Server

Output Console displays useful information about the UOC server instance after the administrator start the platform:

```
$ uoc2 start
```

The following information are displayed:

- All RESTAPI dynamically loaded by categories (~core public interfaces)
- All RESTAPI dynamically loaded from installed Plugins (~plugins public interfaces)
- plugin name, version, host server, port server, demo (true | false)
- Listening Port (default is 3000)
- Listening protocol (default is http)
- Server timeout (default is unlimited)
- Node environment (development or production)
- UOC Node root
- Authentication (local or SAML)

- GUI Database information (server, port, protocol)
- Startup parameter (import policy)
- GUI database initialization logs
- All these info are logged in logs files.

NOTE: IF some plugins make the platform unstable, you can disable it at startup (turn off 'active' in configuration file in /server/public/addons/<pluginId>/conf

19.4 Troubleshooting UOC Plugins

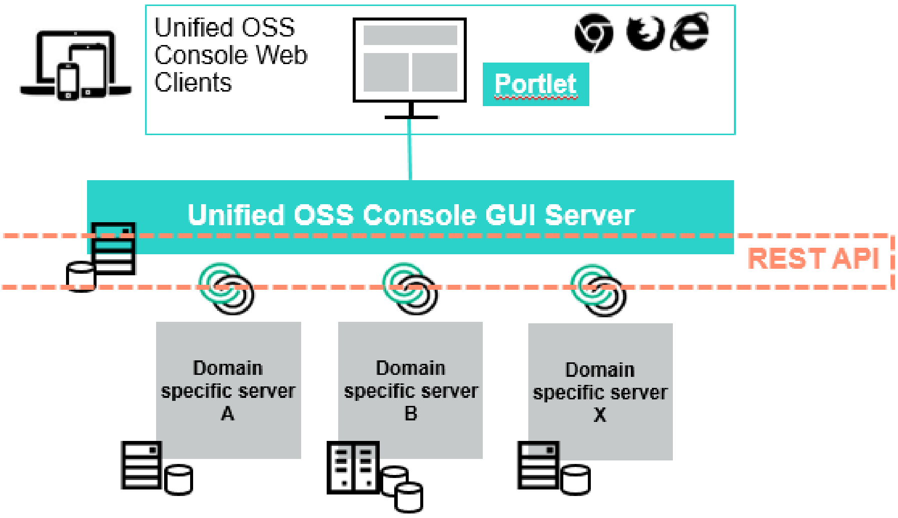


Figure 48: Troubleshooting UOC Plugins

You can check if a specific Plugin interface is working calling directly the public RESTAPI. This interface is used by the web browser to get information and generates GUI.

You can list the available packages for this plugins, and get a list of all packages.

Returns a list of packages for this plugin (ossa) in JSON format

http://localhost:3000/V1.0/domains/ossa/packages

You can get details of 1 package from the list.

http://localhost:3000/V1.0/domains/ossa/packages/MBBQOE_Trial

You can get list of all views for this plugin

http://localhost:3000/V1.0/domains/ossa/packages/views

You can get list of views of this package

http://localhost:3000/V1.0/domains/ossa/packages/MBBQOE_Trial/views

You can get list of all workspaces for this plugin http://localhost:3000/V1.0/domains/ossa/packages/workspaces

You can get list of workspaces of this package

http://localhost:3000/V1.0/domains/ossa/packages/MBBQOE_Trial/workspaces

19.5 Troubleshooting UOC Widgets

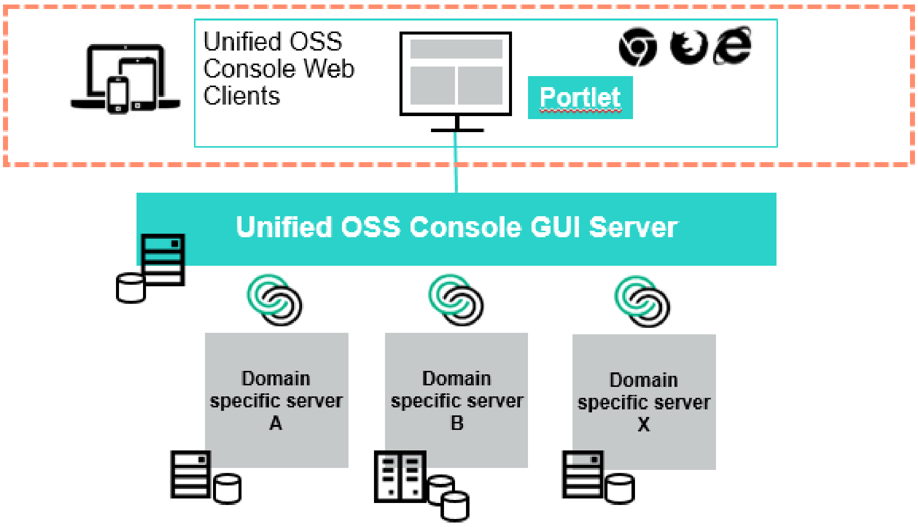


Figure 49: Troubleshooting UOC Widgets

All the client components can be troubleshoot with thee web browser in developer / debug mode (usually F12 key)

The Web Browser console let you display:

- Console logs of the web browser. (by severity)
- URL requested to the UOC Server (response, data…)
- Timeframe of response

The web browser also allow you to clear the cache if you think an issue is linked to a cache pb (css, data,..)

IT also allow you to set breakpoint and debug the code, BUT in the runtime kit, all the code has been minified (to reduce footprint), so it will not very easy without a sdk version to get details of the pb trying to debug the code.

➔ All the http logs and UOC object access are logged by the server in the file <installDir>/logs

Recommandation:

Support of web applications can be tricky, so you need to multiple testing to isolate the root cause.

- Test on several browsers
- Test on several Devices
- Test on several OS

It is strongly recommended to try several web browser if you see graphical issue or limitation. It could be linked to a specific version of OS, patch level from the web browser used.

It is strongly recommended to try on several device if you notice issue or graphical limitation on a specific device (ex: table). It could be linked to a responsive layout that is wrongly implemented and do not display the expected result.

19.6 Troubleshooting Data / Domain Servers

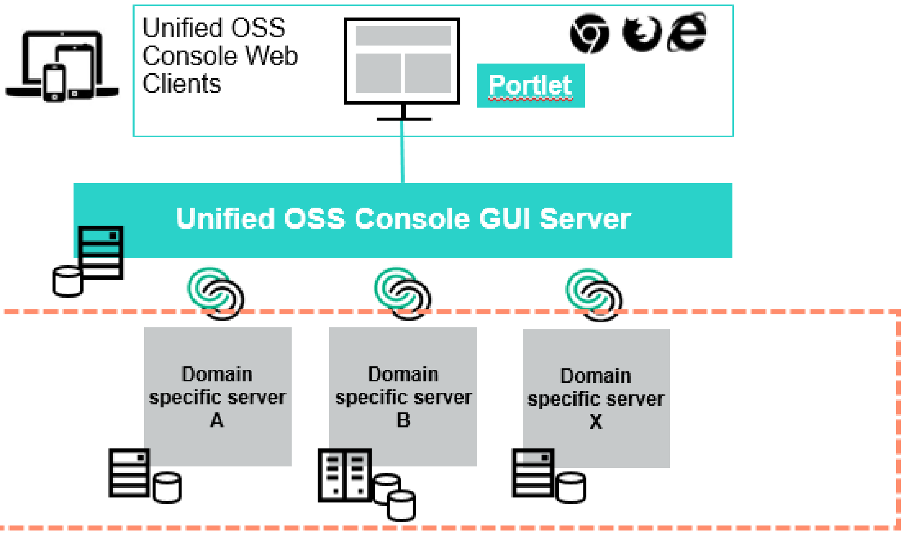


Figure 50: Troubleshooting Data / Domain Servers

To troubleshoot the Data server, you need to understand which server is queried and what is the protocol used. It is totally custom by data server and all the api are heterogeneous. You need to read carefully the documentation to see how troubleshoot such server.

Tips for OSSA Server (OSS Analytics Foundation)

For OSSA Server, you can easily call the REST API to get 'raw' data from the data server.

List available packages (where ossv031.gre.hp.com is an OSS Analytics server)

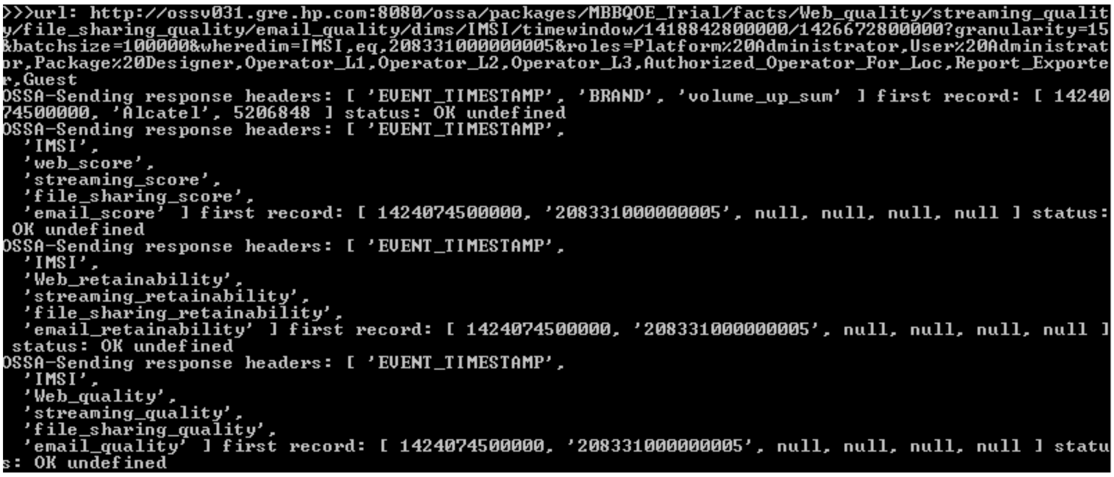
http://ossv031.gre.hp.com:8080/ossa/packages

Details of one specific packages:

http://ossv031.gre.hp.com:8080/ossa/packages/MBBQOE_Trial

You can also collect data with the right syntax. A cookbook exist from the OSSA server team that details all the features

All the request done by the plugin to the data or domain server are logged in the console (server side)



```
>>>url: http://ossv031.gre.hp.com:8080/ossa/packages/MBBQOE_Trial/facts/Web_quality/streaming_qualit
y/file_sharing_quality/email_quality/dims/IMSI/timewindow/1418842800000/1426672800000?granularity=15
&batchsize=100000&wheredim=IMSI,eq,208331000000005&roles=Platform%20Administrator,User%20Administrat
or,Package%20Designer,Operator_L1,Operator_L2,Operator_L3,Authorized_Operator_For_Loc,Report_Exporte
r,Guest
OSSA-Sending response headers: [ 'EVENT_TIMESTAMP', 'BRAND', 'volume_up_sum' ] first record: [ 14240
74500000, 'Alcatel', 5206848 ] status: OK undefined
OSSA-Sending response headers: [ 'EVENT_TIMESTAMP',
  'IMSI',
  'web_score',
  'streaming_score',
  'file_sharing_score',
  'email_score' ] first record: [ 1424074500000, '208331000000005', null, null, null, null ] status:
 OK undefined
OSSA-Sending response headers: [ 'EVENT_TIMESTAMP',
  'IMSI',
  'Web_retainability',
  'streaming_retainability',
  'file_sharing_retainability',
  'email_retainability' ] first record: [ 1424074500000, '208331000000005', null, null, null, null ]
 status: OK undefined
OSSA-Sending response headers: [ 'EVENT_TIMESTAMP',
  'IMSI',
  'Web_quality',
  'streaming_quality',
  'file_sharing_quality',
  'email_quality' ] first record: [ 1424074500000, '208331000000005', null, null, null, null ] statu
s: OK undefined
```

Figure 51: Example of troubleshooting Data / Domain Servers (OSSA)

19.7 Troubleshooting GUI Database (Apache CouchDB)

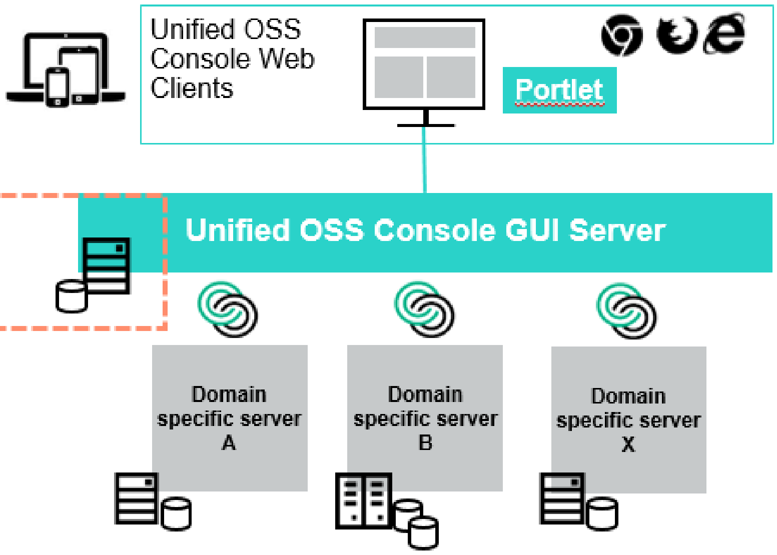


Figure 52: Troubleshooting GUI Database

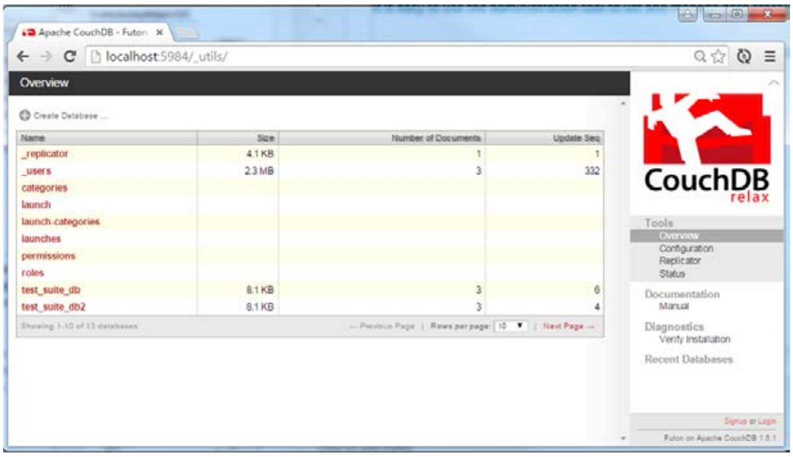
CouchDB stores data as "documents", as one or more field/value pairs expressed as JSON. Field values can be simple things like strings, numbers, or dates; but ordered lists and associative arrays can also be used. Every document in a CouchDB database has a unique id and there is no required document schema.

http://couchdb.apache.org/

It is easy to use the administration tool to list and manage data stored in this database.

Check if the DB is alive: http://localhost:5984 return a JSON answer.

CouchDB includes a web-based front end called Futon which can be accessed by your browser via the following address: http://localhost:5984/_utils/



Name	Size	Number of Documents	Update Seq
_replicator	4.1 KB	1	1
_users	2.3 MB	3	332
categories			
launch			
launch-categories			
launches			
permissions			
roles			
test_suite_db	8.1 KB	3	6
test_suite_db2	8.1 KB	3	4

Showing 1-10 of 13 databases ← Previous Page | Rows per page: 10 | Next Page →

Figure 53: Apache CouchDB Front end Futon

Chapter 20 Frequent errors and solutions

20.1 Upgrading npm: SSL23_GET_SERVER_HELLO:tlsv1 alert access denied

During this installation step, you might have an access denied error.

```
[root@ossv085 ~]# npm install -g npm@latest
npm ERR! Linux 2.6.32-431.el6.x86_64
npm ERR! argv "/usr/bin/node" "/usr/bin/npm" "install" "-g" "npm@latest"
npm ERR! node v4.5.0
npm ERR! npm  v2.15.9
npm ERR! code EPROTO
npm ERR! errno EPROTO
npm ERR! syscall write

npm ERR! write EPROTO 139982692255520:error:14077419:SSL
routines:SSL23_GET_SERVER_HELLO:tlsv1 alert access denied:
npm ERR!
npm ERR!
npm ERR! If you need help, you may report this error at:
npm ERR!     <https://github.com/npm/npm/issues>

npm ERR! Please include the following file with any support request:
npm ERR!     /root/npm-debug.log
```

This is because you are behind a firewall, and you need a proxy. You can:

Export the http_proxy variable:

```
export http_proxy=http://<my_proxy_host>:<my_proxy_port>
```

Or set the proxy in the configuration of npm package:

```
npm config set proxy <your http proxy:port>
npm config set https-proxy <your https proxy:port>
```

You shouldn't have SSL23 error anymore when running

```
npm install -g npm@latest
```

IMPORTANT: Don't forget to unset the http_proxy variable if you use the "export" command:

```
unset http_proxy
```

20.2 Apache CouchDB

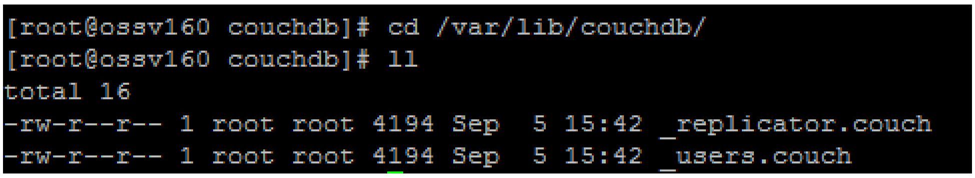
If you have the following error when starting the CouchDB service:

```
# service couchdb start
Redirecting to /bin/systemctl start  couchdb.service
Job for couchdb.service failed because the control process exited with
error code. See "systemctl status couchdb.service" and "journalctl -xe"
for details.
```

You have a right access issue. It is because the first time you tried to launch couchdb, you launched it this way :

```
# couchdb start
```

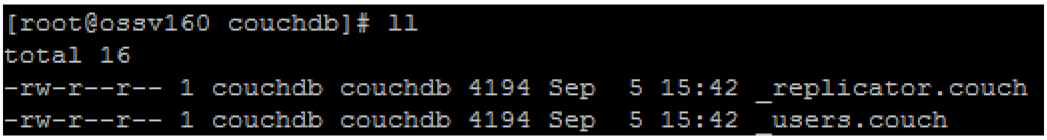
Therefore, it created .couch files in /var/lib/couchdb with a **wrong** right access.



```
[root@ossv160 couchdb]# cd /var/lib/couchdb/
[root@ossv160 couchdb]# ll
total 16
-rw-r--r-- 1 root root 4194 Sep  5 15:42 _replicator.couch
-rw-r--r-- 1 root root 4194 Sep  5 15:42 _users.couch
```

These files should have the couchdb right access. Execute the following command:

```
# chown couchdb:couchdb *
```



```
[root@ossv160 couchdb]# ll
total 16
-rw-r--r-- 1 couchdb couchdb 4194 Sep  5 15:42 _replicator.couch
-rw-r--r-- 1 couchdb couchdb 4194 Sep  5 15:42 _users.couch
```

Check also the files in the following directory:

- /var/log/couchdb
- /var/run/couchdb

TIP: If you want to see the files without permissions for the couchdb user, you must log-in with couchdb account and launch the command “couchdb start” to see the errors.

```
# su – couchdb

# couchdb start

Apache CouchDB 1.6.1 (LogLevel=info) is logging to
/var/log/couchdb/couch.log.

Error opening log file /var/log/couchdb/couch.log: permission
denied{"init terminating in do_boot",{{…
```

Now, you should be able to launch couchdb as a service.