OX Cloud Plugins Technical Documentation for
1.11.11-rev5

2022-05-11

Copyright notice

Contents

1 General Information

1.1 Warnings

Warning
It is mandatory to restart the **open-xchange** service on all middleware nodes after performing the update.

Warning
Custom configuration or template files are potentially not updated automatically. After the update, please always check for files with a **.dpkg-new** or **.rpmnew** suffix and merge the changes manually. Configuration file changes are listed in their own respective section below but don't include changes to template files. For details about all the configuration files and templates shipped as part of this delivery, please read the relevant section of each package.

1.2 Delivery Comment

This delivery was requested with following comment:

```
Cloud Plugins 1.11.11 Feature Delivery for Core 7.10.6
```

1.3 Install Package Repository

This delivery is part of a restricted software repository:

https://software.open-xchange.com/components/cloud-plugins/stable/1.11.11/DebianBuster
https://software.open-xchange.com/components/cloud-plugins/stable/1.11.11/DebianStretch
https://software.open-xchange.com/components/cloud-plugins/stable/1.11.11/RHEL7

1.4 Build Dependencies

This delivery was build and tested with following dependencies:

```
AppSuite:node-10,plugins-1.7.1-rev5,backend-7.10.6-rev14
```

1.5 Notice

Info
Some configurations can be changed without restarting the service, please call following command for getting a list of supported settings.

```
/opt/open-xchange/sbin/listreloadables
```

Please use following command to enable capable and changed configurations on a running system.

```
/opt/open-xchange/sbin/reloadconfiguration
```

2 Announcements Plugin

Bundle Identifier	com.openexchange.cloudplugins.announcements, com.openexchange.cloudplugins.announcements.json, com.openexchange.cloudplugins.announcements.rdb , com.openexchange.cloudplugins.announcements.ws
Package(s)	open-xchange-cloudplugins

Required capabilities	`com.openexchange.capability.announcements`
Available since	1.11.8

This feature allows an admin user to `create`/`update`/`delete` announcements that will be shown to all end users, see https://documentation.open-xchange.com/components/cloudplugins/stable/ for the documentation of the Admin REST API.

2.1 Configuration

This feature needs a registered Global Database in order to store the announcements added via the Admin REST API. If there are more then one global databases configured and or it is not desired to use the `default` database, the following configuration option has to be set accordingly:

```
1  com.openexchange.cloudplugins.announcements.globaldbContextGroup
```

Besides the capability `com.openexchange.capability.announcements`, the following configuration options are available

```
1  # Whether to enable the announcements feature
2  # Not reloadable.
3  #
4  # Default: false
5  com.openexchange.cloudplugins.announcements.enabled=true
6
7  # The default group name to access the global database.
8  # See /opt/open-xchange/etc/globaldb.yml for more information and
9  # configuration possibilities
10 # Not reloadable.
11 #
12 # Default: default
13 com.openexchange.cloudplugins.announcements.globaldbContextGroup=default
```

Warning
When the announcement bundle isn't started due to whatever reason, it can't be indicated using the `listbundles` or `getmissingservices` commands. This is due to the fact that we have a chicken-and-egg situation in automated deployments. The `allpluginsloaded` tool will not succeed until the globaldb has been registered, but in cannot be registered until all bundles are up and running including the announcements bundles. That's why they will only log errors in case the globaldb is not available, but don't fail.

2.2 HTTP API

The bundle `com.openexchange.cloudplugins.announcements.json` allows the UI to retrieve announcements applicable to the logged in end user and perform operations such as `set as seen` and `set as discarded`

The HTTP API requests provide the userId and contextId via session lookup. The Middleware takes care of determining the "user brand" and retrieve the announcements applicable for that brand Once obtained this information should be cached in the User session.

2.2.1 Usage

It is expected that when the user logs in the UI will make a `list` request to obtain all announcements that are applicable to the user.

Info
Please note that this will include all currently valid announcements that the user has not yet discarded (even if already previously seen).

The UI will show the first announcement in a popup dialog and display a counter in the notification area if more are available.

The user may view the announcements and simply close it (`seen` action) or he may select “do not show this again” prior to closing it (`discard` action).

The UI will periodically refresh and will need to find out whether any new announcements have been created or have entered their validity date range. The UI will then perform a `list_unseen` which will return only valid announcement that have not yet been marked as seen or discarded.

Info
Please note that the announcement message may contain HTML which the middleware will have already sanitized.

2.2.2 API Calls Summary

Parameter	Description
`server`	Hostname or IP of the server.
`sessionId`	The session id of the user.

Please use the **actionURL** prefix below for following table:

```
http://{server}/appsuite/api/announcement?session={sessionId}&action=
```

API Call Action	Description
GET {actionURL}**list**	This is intended to be used once just after the User logs in to initialize the dialog user functionality. It retrieves all currently valid announcements that the user has not yet discarded. Announcements already seen will be included
GET {actionURL}**list_unseen**	This is intended to be used on UI refresh and only after the “list” call has been used once. This retrieves all “unseen announcements” only
PUT {actionURL}**seen**	This call sets the given version of the specific version ID to SEEN status
PUT {actionURL}**discard**	This call sets the given version of the specific version ID to DISCARDED status

2.2.3 Example of the payload returned by `list` and `list_unseen` requests

```
{
    "data": [
        {
            "announcement": "Ad astra per aspera",
            "summary": "summary or title",
            "discardable": true,
            "id": 12345,
            "version": 1
        },
        {
            "announcement": "Carpe diem",
            "summary": "summary or title",
            "discardable": false,
            "id": 12346,
            "version": 2
        }
    ]
}
```

2.2.4 Example of the payload returned by `seen` and `discard` requests

```
{
    "id": 789,
    "version": 2
}
```

3 Cloud Plugins MX Checker Connector

Bundle Identifier	`com.openexchange.cloudplugins.mx.checker`
Package	`open-xchange-cloudplugins-mx-checker`
Dependant Package	`open-xchange-plugins-mx-checker`
Available since	1.11.8

3.1 Overview

This connector uses the MX Checker Framework to relay MX and SPF validity by comparing the respective DNS records for the user's brand and domain. It does so by verifying that a user's domain MX records have no symmetric different with the brand's records. It also validates the user's domain SPF records to ensure outbound mail has the highest likelihood of successful delivery.

3.2 Background

A mail exchanger record (MX record) is a resource record in the Domain Name Server (DNS) which specifies the mail transfer agent(s) responsible for accepting email messages on behalf of a domain name.

A Sender Policy Framework record (SPF record) is a type of Text (TXT) resource record in the DNS, which identifies which mail transfer agent(s) are permitted to send mail on behalf of a domain.

3.3 Configuration

All configuration properties are reloadable.

3.3.1 Connector Configuration

The OX Cloud MX Checker Connector identifier is "`oxcloud_mx_checker`", so you will need to set this as the plugins connector property for all users who will need it.

Setting it like below will set it on the server level, but for shared environments it should be set at the user or context level:

```
# Determines which connector will be used for a user
# This setting is config-cascade aware to support different implementations for each user.
# Default is <none> which means that the feature is disabled for a user
com.openexchange.plugins.mx.checker.connector=oxcloud_mx_checker
```

3.3.2 ActionUrl Configuration

To enable sending users a "fix this" URL in the UI when either the SPF or the MX are invalid or propagating, one can define the following *config-cascade* aware configuration properties:

Property	Description
`com.openexchange.cloudplugins.mx.checker.url.propagating`	used when the MX or SPF result is valid and propagating

Property	Description
com.openexchange.cloudplugins.mx.checker.url.invalidmx	used for the MX result when it's invalid (unless both MX and SPF are invalid
com.openexchange.cloudplugins.mx.checker.url.invalidspf	used for the SPF result when it's invalid (unless both MX and SPF are invalid)
com.openexchange.cloudplugins.mx.checker.url.invalid	used when both MX and SPF are invalid, or for the MX and/or SPF result when it's invalid and when .invalidmx or .invalidspf is not defined, as a fallback
com.openexchange.cloudplugins.mx.checker.url.valid	used for the MX and/or SPF result when it's valid

If the property is not defined, no actionUrl will be set.

If the property is defined, it is used as a text template, with the following placeholders being replaced before returning the value as a URL in the actionUrl attribute:

Placeholder	Value
${brand}	the brand name
${domain}	the domain of the user's primary email address
${userId}	the user's numeric id
${contextId}	the user's numeric context id
${login}	the user's login string (that was used to authenticate)
${locale}	the user's locale (e.g. de-DE), or an empty string if not set
${language}	the user's preferred language (e.g. de), or an empty string if not set

Example:

```
1  com.openexchange.cloudplugins.mx.checker.url.invalid=https://help.acme.com/mxcheck/invalid
       /${domain}.jsp?login=${login}
```

3.3.3 Caching Configuration

Next, set the caching properties:

```
1  # The max number of cached user DNS info result when valid.
2  # Use 0 for no cache.
3  #
4  # Default: 10000
5  #
6  com.openexchange.cloudplugins.mx.checker.cache.max=10000
7
8  # The amount of time in seconds after a result is written to the cache
9  # until it expires.
10 # Use 0 for no cache.
11 #
12 # Default: 3600
13 #
14 com.openexchange.cloudplugins.mx.checker.cache.expire.seconds=3600
```

3.4 Caching

As mentioned above, there is an option to using caching of positive validation results. To enable, set the cache.max and expire.seconds properties to greater than 0 (or 0 to disable caching). Whenever a validation request is made and both MX and SPF records are valid, it will cache that result for the configured amount of time. The next time a validation request is initiated, if it is within that amount of time, it will use the cached result instead of making the more expensive DNS queries.

3.5 Flow

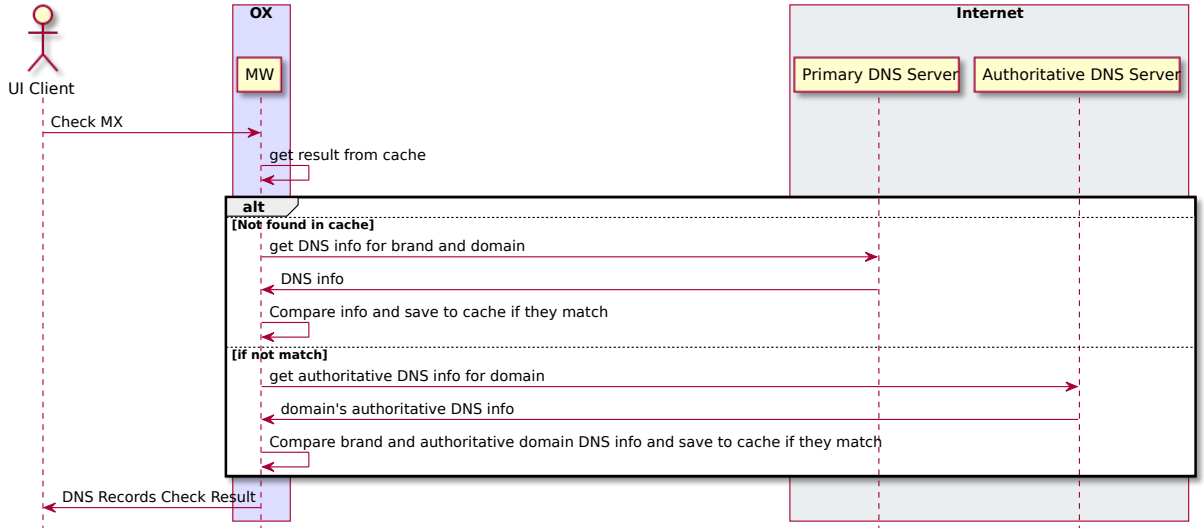

Figure 1: PlantUML 1

3.6 Testing/QA

There is a mocked DNSLookup implementation provided that can be enabled by setting the hidden property:

```
1 com.openexchange.cloudplugins.mx.checker.dnsLookup=mock
```

Then simply create users under any brand with the following domains:

Domain	Result
`invalid`	mx and spf invalid
`invalidmx`	mx invalid and spf valid
`invalidspf`	mx valid and spf invalid
`propagating`	mx and spf invalid with propagating true with 49 seconds
*	mx and spf valid

4 Shipped Version

4.1 Package open-xchange-cloudplugins

The Open-Xchange cloud plugin bundles
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-admin-reseller (<<7.10.7)
open-xchange-admin-soap-reseller (>=7.10.6)
open-xchange-cassandra (<<7.10.7)
open-xchange-cassandra (>=7.10.6)
open-xchange-mailfilter (<<7.10.7)
open-xchange-mailfilter (>=7.10.6)
open-xchange-rest (<<7.10.7)
open-xchange-rest (>=7.10.6)
```

Conflicts with:

```
open-xchange-authentication-database
open-xchange-authentication-imap
open-xchange-authentication-ldap
```

4.1.1 General Functionality

The package open-xchange-cloudplugins contains a set of bundles to manage an integrated platform consisting of OX App Suite components and dovecot. It uses some additional software components in order to do that.

- openldap to store information for mail routing used by dovecot and postfix and to act as authentication source
- postfix as mail server
- nginx to act as a proxy for http, imap, pop3 and smtp
- cassandra to store data that often changes like last login attempts, etc.

The package itself contains some OSGi bundles explained in the following sections and the file `cloudplugins.properties` contain configurations shared among each other.

4.1.2 Admin Plugins

Those plugins extend the OX provisioning capabilities. They maintain the data required into the openldap server to do mail routing and authentication using the 4.1.8 LDAP Management functionality. Whenever a context or user is created, deleted or changed, the corresponding data in the ldap server is also updated.

```
com.openexchange.admin.cloudplugins
com.openexchange.admin.cloudplugins.console
```

4.1.2.1 SOAP API

Also in these bundles, there's another SOAP API providing cloud-plugins specific methods. Please see for further details the OXaaS SOAP API Documentation.

4.1.2.2 Admin REST API

Starting with release 1.6.3, there's now also a REST API, check `https://documentation.open-xchange.com` for a link.

Info

Note that in order for this API to be accessible, you need to add an entry like `ProxyPass /oxaas balancer://oxcluster/oxaas` to your `proxy_http.conf`.

4.1.3 Authentication Plugin

This plugin authenticates against the ldap server. To determine the user it needs to know the brand the user is belonging to. This is done using a configurable HTTP host header. This host header is added by the nginx proxy which sits in front of the ox middleware. Please see corresponding file `cloudplugins-authentication.properties` for configuration details.

```
com.openexchange.authentication.cloudplugins
```

4.1.4 Masterauth Servlet

This servlet allows to create an ox http session on behalf of a configurable master login and password without knowing the individual users login and password.

```
com.openexchange.cloudplugins.api.masterauth
```

This servlet provides the following API call which returns a valid OX session.

```
GET /api/oxaas/masterauth/brand/<brand>/context/<contextid>/user/<userid>
```

Info

Note that this only works with toplevel brands! The following entry must be added to the apache proxy configuration:

```
ProxyPass /api/oxaas balancer://oxcluster/api/oxaas
```

The request has to provide a `X-AUTHENTICATION` header containing the value of

```
Base64( HMAC-SHA1( brand-auth-Key, UTF-8-Encoding-Of( StringToSign ) ) );
```

```
StringToSign = brand " " +
contextid + " " +
userid;
```

brand-auth-key is the value of the ldap nginxAuthKey Attribute of the brand entry. In addition to that, the requestor has to implement HTTP Basic auth using a configured master login and password. How to generate the X-AUTHENTICATION header with perl:

```
perl -MMIME::Base64 -MDigest::HMAC_SHA1=hmac_sha1
-e 'print encode_base64(hmac_sha1("example.com 24 3","secret"))." ;"'
```

How to test with wget:

```
wget -dv --user=admin --password=secret
--header='X-AUTHENTICATION: 6cmrSTcWCcyOV7mzkSgHDbrk8RI='
http://example.com/api/oxaas/masterauth/brand/example.com/context/24/user/3 -O -
```

The `cloudplugins-master-auth-servlet.propertie` configuration file belongs to this servlet.

4.1.5 Cassandra

This bundle utilizes the cassandra bundle from the ox middleware to read and write data from/to a cassandra cluster.

```
com.openexchange.cloudplugins.cassandra
```

As of now, the following data structures have to be created in order to use this bundle.

```
1  $ cqlsh
2
3  create keyspace ox WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' :
       1 };
4  use ox;
5  CREATE TYPE login_info ( login_time timestamp, ip_address text);
6  CREATE TABLE logins (ox_id text PRIMARY KEY, brand text, logins map<text, frozen<
       login_info>> );
7  CREATE TABLE failure_logins (ox_id text PRIMARY KEY, brand text, login_errors map<text,
       int>);
8  CREATE TABLE alias_log ( alias text PRIMARY KEY, brand text, creation_date timestamp,
       deletion_date timestamp, ox_id text );
9  CREATE TABLE quota_usage ( ox_id text,  type text, usage bigint, count bigint, primary key
        ((ox_id), type));
10 CREATE TABLE user_property ( ox_id text PRIMARY KEY, brand text, sieve_autoforward int,
       sieve_forward_status int );
```

```
11 CREATE TABLE permission_change_history_by_brand ( brand text, ox_id text, id timeuuid,
       enabled frozen<set<text>>, disabled frozen<set<text>>, reason text, ip_address text,
       auth_user text, client_ip text, client_user text, PRIMARY KEY (brand, id) );
12 CREATE TABLE permission_change_history_by_oxid ( brand text, ox_id text, id timeuuid,
       enabled frozen<set<text>>, disabled frozen<set<text>>, reason text, ip_address text,
       auth_user text, client_ip text, client_user text, PRIMARY KEY (ox_id, id) );
13 CREATE TABLE app_passwords_history ( ox_id text, cid int, id text, usage_time timestamp,
       client text, user_agent text, ip_address text, PRIMARY KEY (cid, ox_id, id) );
```

The complete bundle can be turned off and some functionality must be configured per brand. See corresponding configuration file `cloudplugins-cassandra.properties`.

4.1.6 Dovecot

This is an interface to the doveadm REST API of dovecot. Right now it only implements querying the mail quota usage of users.

```
com.openexchange.cloudplugins.dovecot
```

The `doveadm-config.properties` configuration file belongs to this interface.

4.1.7 Mailmapping

The mail mapping is required to integrate with OX Guard. It resolves an email address into a context and userid.

```
com.openexchange.cloudplugins.mailmapping
```

4.1.8 LDAP Management

This provides methods to maintain data in the LDAP server.

```
com.openexchange.cloudplugins.management
```

4.1.8.1 LDAP structure

The ldap schema used with these bundles can be found in the `ldap` folder within this git repository. The ldap tree consists of three main branches:

- 1. The brand tree contains the brand accounts with all users belonging to each brand below.
- 2. The configuration tree containing configuration entries like mailstores, mailservers, etc.
- 3. The context tree containing all contexts of all brands in the entire system.

A brand is a the same as a subadmin in the ox reseller bundle. This reseller bundle is a requirement to run the cloud-plugins environment. Due to the fact that all users within a brand are in the tree below a brand, user logins must not be unique within contexts as it us usually in an open-xchange environment, they must be unique within each brand. For details about the reseller bundle please see the Reseller Bundle Documentation.

4.1.9 Nginx Servlet

This provides a servlet used by Nginx to authenticate external IMAP, POP and SMTP users against the ox cloud-plugins scheme.

```
com.openexchange.cloudplugins.nginx.auth.servlet
```

Where users directly using OX webmail are authenticated using the 4.1.3 Authetication Plugin, external IMAP, POP and SMTP logins must also be mapped from the internal `uid@contextid` representation to a login string the corresponding brand prefers. This might be an email address, a phone number, or whatever else. Nginx acts as a IMAP, POP and SMTP proxy in front of dovecot and uses the nginx auth servlet as an authentication and transformation source to provide dovecot with the

internal login id after it successfully authenticated the user. See corresponding configuration file `nginx-auth-servlet.properties`.

4.1.10 Cloud Report

This is a central tool to get service usage informations on user level. Therefore the administrator has different options to generate this report, which are explained later. The report procession is designed to be highly efficient in regard of memory and cpu usage. If not configured otherwise, the used resources should never interfere with the processes of the operating system. While in procession, parts of the report are stored on hdd, merged into the resulting report and deleted, when the report is done. The report is processing each relevant schema in its own thread and all schemas in parallel for maximum speed. See corresponding configuration file `cloudplugins_report.properties`.

```
com.openexchange.cloudplugins.report
```

4.1.10.1 Requirements

The report loads data from OX-DB as well as cassandra and LDAP storage. If no cassandra or OX-DB connection can be established, the report will not start. Without a valid LDAP connection, errors will be logged inside the report.

4.1.10.2 Report Types

Currently, there are two report types.

- Metrics - The original cloud-plugins report which provides user usage information in JSON.
- TKG112 - Provides user alias information in CSV.

These report types are described further below.

4.1.10.3 Usage

In general, there can be only one report processed at a time. A second report will be denied and the ID of the current report will be displayed. This report can be aborted with the corresponding option and all stored data will be deleted. During procession, the current status can be requested and the user will be provided with all finished contexts compared to totals.

Warning

The processed contexts will only be updated when the schema is completely processed to evade potential bottlenecks because of Object locks.

```
1  Usage: createreport
2   -h,--help                                      Prints a help text
3      --environment                               Show info about commandline environment
4      --nonl                                      Remove all newlines (\n) from output
5      --responsetimeout <responsetimeout>         response timeout in seconds for reading
           response from the backend (default 0s; infinite)
6   -A,--adminuser <adminuser>                    * master admin user name
7   -P,--adminpass <adminpass>                    * master admin password
8   -s,--timeframe-start <timeframe-start>          Set the starting date of the desired
       timeframe in which the user logins are considered, format: dd.mm.yyyy.
9   -e,--timeframe-end <timeframe-end>              Set the ending date of the desired
       timeframe in which the user logins are considered, in format: dd.mm.yyyy.
10  -a,--ignore-admin                               Ignore admins and dont show users of
       that category in the report.
11  -d,--show-drive-metrics                         Add drive metrics for every user.
12  -m,--show-mail-metrics                          Add mail metrics for every user.
13  -b,--single-brand <single-brand>                Create a report for the selected brand
       only. Identified by the brand admins sid.
14  -t,--terminate-report                           Terminates the currently processed
       report uuid.
15  -p,--pending-reports                            Get the status of the pending report.
16
17 Entries marked with an asterisk (*) are mandatory.
18 Entries marked with an question mark (?) are mandatory depending on your
```

```
configuration.
Entries marked with a pipe (|) are mandatory for one another which means that
at least one of them must be set.
```

```
Usage: createtkg112report
 -h,--help                                        Prints a help text
    --environment                                 Show info about commandline environment
    --nonl                                        Remove all newlines (\n) from output
    --responsetimeout <responsetimeout>           response timeout in seconds for reading
          response from the backend (default 0s; infinite)
 -A,--adminuser <adminuser>                      * master admin user name
 -P,--adminpass <adminpass>                      * master admin password
 -b,--single-brand <single-brand>                * Brand to create report for. Identified
     by the brand admins sid.
 -t,--terminate-report                             Terminates the currently processed
     report uuid.
 -p,--pending-reports                              Get the status of the pending report.

Entries marked with an asterisk (*) are mandatory.
Entries marked with an question mark (?) are mandatory depending on your
configuration.
Entries marked with a pipe (|) are mandatory for one another which means that
at least one of them must be set.
```

4.1.10.4 Data Format and Storage

The Metrics Report data is stored in JSON format and looks like the example below.

```
{
   "uuid":"28b3573af6734877a448ab614698d115",
   "reportType":"OXaaS-report",
   "timestamps":{
      "start":1498477236283,
      "stop":1498477276490
   },
   "version":{
      "buildDate":"01.01.2017",
      "version":"7.8.3"
   },
   "configs":{
      "options":{
         "show-drive-metrics":false,
         "timeframe-start":1466941236283,
         "show-mail-metrics":false,
         "single-brand":0,
         "ignore-admin":true,
         "timeframe-end":1498477236283
      }
   },
   "errors":{
      "Exception-ID":"ERROR-ID Categories=ERROR Message='Message' exceptionID=Exception-ID
          "
   },
   "oxaas":{
      "capabilitySets":{
         "283724704":"autologin,blacklist,...",...
         },
      "brandname":{
         "totals":{
            "quota" : 2621440000,
   "quotaUsage" : 2522466,
   "mailQuota" : 45365,
   "mailQuotaUsage" : 0
```

```
          },
          "1":{
             "3":{
                "capabilitySet":"283724704",
                "drive" : {
           "mime-types" : {
            "text/plain" : 5,
              "application/zip" : 2
          },
          "file-count-all-versions" : 7,
          "quota" : 104857600,
          "used-quota" : 2511077,
          "file-size-min" : 5,
          "file-count-latest-version" : 7,
          "file-size-avg" : 358725,
          "file-size-max" : 1255526
        },
        "mail" : {
          "mail-quota" : 2048,
          "mail-quota-usage" : 0
        },
                "imap-login":"3@1",
                "login-info":"adam@brandname",
                "email":"adam@brandname",
                "user-logins":{
      "HTTP" : 1497364334092,
         "open-xchange-appsuite" : 1497364334090
                },
                "unified-quota-enabled":true,
                "unified-quota-limit":104859648,
                "unified-quota":2511533
             }
          }
       }
    }
}
```

The TKG112 Report data is stored in CSV format and looks like the example below. It will output an additional file with _info appended that contains configuration information as well as errors.

```
Email_nameID, Email_Displayname, Email_Begin, Email_Address, Email_Create, Email_Remove,
    Email_isactive
ID1, TestUser1, 2016-12-06 14:11:00 +0000, testuser@brand.com, 2016-12-06 14:11:00 +0000,
    , true
ID2, TestUser2, 2016-12-06 14:11:00 +0000, testuser2@brand.com, 2016-12-06 14:11:00 +0000,
     2016-12-06 14:11:00 +0000, false
```

4.1.10.5 TKG112 Data Description

All

- Email_nameID - Varchar (256) - OX ”userName”
- Email_Displayname - Varchar (320) - Displayname used when sending emails
- Email_Begin - Date (YYYY-MM-DD hh:mm:ss TIME_ZONE) - Date of the registration in the OX system
- Email_Address - Varchar (256) - email address including the domain name
- Email_Create - Date (YYYY-MM-DD hh:mm:ss TIME_ZONE) - Date when the email address was created
- Email_Remove - Date (YYYY-MM0-DD hh:mm:ss TIME_ZONE) - Date when the email address

was removed (optional)

- Email_isactive - Boolean true/false - Flag if the email account is active or not. Will be always true when Email_Remove is empty.

4.1.10.6 Metrics Data Description

General

- uuid - The report identifier
- reportType - Type of this report, so far only ”OXaaS-report” possible
- timestamps - The start and end time of the report in milliseconds
- version - The builddate and version of the processing server
- errors - Map of all errors occured during procession. Key is the exception Id and value is further information like message, category and Error-Id

Configs

- show-drive-metrics - true or false
- timeframe-start - The used timeframe start, if not set by the user, one year in the past is used
- show-mail-metrics - true or false
- single-brand - The sid of the brand admin or 0 if not set
- ignore-admin - true or false
- timeframe-end - The used timeframe end, if not set, the starting time of the report is used

OXaaS

- capabilitySets - All capability sets determined by the report. Key is the hashed value of all capabilities in a list
- brandname - The brandname with all userdata for the brand

Per Brand

- totals - Drive and mail quota information for the whole brand, comulated values of all users.
- contexts - All context informations for this brand

Per User

- capabilitySet - The hash value of the capability-set this user has
- drive - All drive data for this user (only present if drive option is true)
- mail - All mail data for this user (only present if mail option is true)
- imap-login - Imap login address
- login-info - Login info, gather from LDAP
- email - users email address
- user-logins - A list of all protocols, the user used to login with the latest timestamp as milliseconds
- unified-quota-enabled - true or false
- unified-quota-limit - This users unified quota limit (only present if unified quota is enabled for this user)
- unified-quota - This users unified quota (only present if unified quota is enabled for this user)

Drive Data

- mime-types - Map of all mimetypes and their amount
- file-count-all-versions - Number of all files and versions for this user
- quota - Quota limit
- used-quota - Used quota
- file-size-min - Smallest file size in this storage
- file-count-latest-version - Number of files, respecting only the latest version
- file-size-avg - Average filesize
- file-size-max - Maximum filesize

Quota Data Sources With unified quota enabled

- drive limit - From LDAP
- drive quota - From Cassandra
- mail limit - From LDAP
- mail quota - From Cassandra

Without Unified Quota

- drive limit - From filestore
- drive quota - From filestore
- mail limit - From LDAP
- mail quota - From Cassandra

4.1.11 Unified Quota

This implements the unified quota feature for cloud-plugins. It requires cassandra to be running and initialized with the `quota_usage` table, see 4.1.5 Cassandra section. All file quota usage is updated into and read from cassandra.

```
com.openexchange.cloudplugins.unifiedquota
```

Please see for information about how to use this bundle the Unifiedquota Documentaion

4.1.12 Passwordchange

This implements the ox password change callback API in order to be able to change passwords in LDAP.

```
com.openexchange.passwordchange.cloudplugins
```

4.1.13 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins
```

4.1.14 Configuration

For details, please see appendix A
/opt/open-xchange/etc/plugin/mailstore-cloudplugins.properties (page 27)
/opt/open-xchange/etc/cloudplugins-authentication.properties (page 28)
/opt/open-xchange/etc/cloudplugins-master-auth-servlet.properties (page 29)
/opt/open-xchange/etc/cloudplugins-cassandra.properties (page 29)
/opt/open-xchange/etc/cloudplugins.properties (page 37)

/opt/open-xchange/etc/doveadm-config.properties (page 37)
/opt/open-xchange/etc/nginx-auth-servlet.properties (page 39)
/opt/open-xchange/etc/cloudquotaservice.properties (page 39)
/opt/open-xchange/etc/cloudquotaservice-cassandra.properties (page 40)
/opt/open-xchange/etc/cloudplugins_report.properties (page 41)

4.2 Package open-xchange-cloudplugins-antiphishing-vadesecure-ldap

Implementation of VadeSecure antiphishing for cloudplugins within LDAP
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-plugins-antiphishing (<<1.8.0)
open-xchange-plugins-antiphishing (>=1.6.6)
open-xchange-plugins-antiphishing-vadesecure (<<1.8.0)
open-xchange-plugins-antiphishing-vadesecure (>=1.6.6)
```

4.2.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-antiphishing-vadesecure-ldap
```

4.2.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/cloudplugins-antiphishing-vadesecure-ldap.properties (page 41)

4.3 Package open-xchange-cloudplugins-blackwhitelist-ldap

Implementation of blacklist whitelist for cloudplugins within LDAP
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-plugins-blackwhitelist (<<1.8.0)
open-xchange-plugins-blackwhitelist (>=1.6.6)
```

4.3.1 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-blackwhitelist-ldap
```

4.3.2 Configuration

For details, please see appendix A
/opt/open-xchange/etc/cloudplugins-blackwhitelist-ldap.properties (page 41)

4.4 Package open-xchange-cloudplugins-forwards-ws

Cloudplugins Admin forwards REST API This package provides a restful API to add/update/delete forwards saved in storage.
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

4.4.1 General Functionality

This plugin provides a middleware restfull API to set mail forwards in the user storage.
List of features implemented by this plugin:

- Main entry point is **/api/oxaas/v1/admin/forwards**
- secured by basic auth mapped to customer login data
- **POST /{contextId}/{alias}** Sets a forward alias
- **PUT /{contextId}/{alias}** Adds recipient to existing forward alias
- **DELETE /{contextId}** Deletes all forward aliases in context
- **DELETE /{contextId}/{alias}** Deletes an alias in a context
- **GET /{contextId}** Returns all forward aliases in a context
- **GET /{contextId}/{alias}** Returns an alias in a context
- **HEAD /{contextId}/{alias}** Checks if an alias in a context is present
- **HEAD /{contextId}/{alias}/{recipient}** Checks, if a recipient of an alias in a context is present

4.4.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-forwards-ws
```

4.4.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/cloudplugins-forwards.properties (page 42)

4.5 Package open-xchange-cloudplugins-keycloak

Keycloak connector This package contains a keycloak connection handler to retrieve access and refresh tokens.
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-oidc (<<7.10.7)
open-xchange-oidc (>=7.10.6)
```

4.5.1 General Functionality

This plugin provides a connector interface to request access and refresh tokens from keycloak. List of features implemented by this plugin:

- Provides ICPKeycloakOAuthAccessTokenService to interact with configureable keycloak endpoints
  - Supports password grant with username and password
  - Supports refresh grant with refresh_token
- Provides additional services to interact with oauth mail handling
  - AuthenticationFailedHandler - will request a new access token, when the imap backend signals, that the current access token is not valid anymore. If that is not possible, the session is terminated
  - SessionInspectorService - will request a new access and refresh token, if the initial access_token provided an expires_in value before the token actually timed out. If that is not possible, the session will be logged out.
- Provides ICPJwtParserService
  - Supports parsing the body of a JWT to read additional provided values from the keycloak endpoint.

4.5.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-keycloak
```

4.5.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/cloudplugins-keycloak.properties (page 44)

4.6 Package open-xchange-cloudplugins-loginproxy-ws

Cloudplugins loginproxy REST API This package provides a restful API for the 2-step login.
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

4.6.1 General Functionality

This plugin provides a middleware restfull API to provide a 2-step login.
List of features implemented by this plugin:

- Main entry point is **/api/oxaas-public/v1/loginproxy**
- not secured, only by IP check rate limit
- **?login=loginValue** provide login pre-check

4.6.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-loginproxy-ws
```

4.6.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/cloudplugins-loginproxy-ws.properties (page 45)
/opt/open-xchange/etc/cloudplugins-loginproxy-forward.yaml (page 45)

4.7 Package open-xchange-cloudplugins-mailfilter

CloudPlugins MailFilter Utilities This package implements a mailfilter interceptor driver framework and provides some useful drivers.
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-mailfilter (<<7.10.7)
open-xchange-mailfilter (>=7.10.6)
open-xchange-rest (<<7.10.7)
open-xchange-rest (>=7.10.6)
```

4.7.1 General Functionality

This plugin provides a mailfilter interceptor driver framework and some useful drivers. List of features implemented by this plugin:

- Registers a MailFilterInterceptor
  - Automatically starts a Driver Manager which tracks MailFilterInterceptor Drivers
  - When a user creates/updates/deletes a filter rule, the driver manager will run each driver that is supported for that user in order of their rank.
- Provides MailFilterInterceptor Drivers - configured via their enabled property
  - RedirectStatusDriver - supports any user in any of the configured brands and tells CloudManagementCassandraService the autoforward status and how many redirects exist.
  - RedirectBlacklistDriver - supports Config Cascade. Blocks users from creating only autoforward or all redirect mail filter rules that use a To Address that is blacklisted.

4.7.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-mailfilter
```

4.7.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/mailfilter-interceptor-drivers.properties (page 46)

4.8 Package open-xchange-cloudplugins-master-auth

Provides a CloudPlugins CloudAuthenticationDriver for master authentication
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

4.8.1 General Functionality

This package supplies a CloudAuthenticationDriver tracked by the open-xchange-cloudplugins package. The driver adds a master authentication mechanism that can be configured by brand. Required configuration:

- com.openexchange.authentication.cloudplugins.user.regex
- com.openexchange.authentication.cloudplugins.authentication.uid.mode.*
- com.openexchange.authentication.cloudplugins.brand.master.auth.<brand> .password

The first two properties are supplied by the open-xchange-cloudplugins package. The last property is new and configured per brand.

4.8.1.1 Example

```
1 com.openexchange.authentication.cloudplugins.brand.master.auth.mybrand.password=secret
```

Configures a master password of ”secret” for the brand ”mybrand”. A brand can only have one suitable CloudAuthenticationDriver, so there must not be a custom driver enabled. This driver is registered higher than the DefaultCloudAuthenticationDriver so it will be used over the default if configured for the brand.

4.8.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-master-auth
```

4.8.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/cloudplugins-masterauth.properties (page 46)

4.9 Package open-xchange-cloudplugins-mx-checker

CloudPlugins MX Checker Connector This package implements an MX Checker Connector for OX Cloud
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-plugins-mx-checker (<<1.8.0)
open-xchange-plugins-mx-checker (>=1.6.6)
```

4.9.1 General Functionality

4.9.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-mx-checker
```

4.9.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/cloudplugins-mx-checker.properties (page 47)

4.10 Package open-xchange-cloudplugins-oidc

OIDC backend for any default Identity Server This package contains multiple OIDC backends for any Identity Server, that fully supports the OIDC protocol.
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-oidc (<<7.10.7)
open-xchange-oidc (>=7.10.6)
```

4.10.1 General Functionality

The plugin provides the backend configuration for OIDC.
List of features implemented by this plugin:

- One or many OIDCBackends
- Supports reloadconfiguration clt without stopping unchanged OIDCBackends
- Can be started in addition to a normal AuthenticationService

4.10.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-oidc
```

4.10.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/cloudplugins-oidc.properties (page 49)

4.11 Package open-xchange-cloudplugins-remote-ldap-auth

Provides remote LDAP CloudPlugins CloudAuthenticationDriver and NginxAuthDriverExtended | DeclarativeNginxAuthDriver
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-ldap-client (<<1.8.0)
open-xchange-ldap-client (>=1.6.6)
```

4.11.1 General Functionality

This package supplies multiple CloudAuthenticationDriver and NginxAuthDriver tracked by the open-xchange-cloudplugins package. The driver adds a remote ldap authentication that can be configured by brand. Required configuration:

- com.openexchange.authentication.cloudplugins.user.regex

The property is supplied by the open-xchange-cloudplugins package.

4.11.1.1 Restrictions

There must only be 1 nginx or Authentication service registered per brand and no other component should register such a service.

4.11.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-remote-ldap-auth
```

4.11.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/cloudplugins-remote-ldap.properties (page 52)

4.12 Package open-xchange-cloudplugins-saml

SAML backend for any default Identity Server This package contains an SAML backend for any Identity Server, that fully supports the SAML protocol.
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-saml-core (<<7.10.7)
open-xchange-saml-core (>=7.10.6)
```

4.12.1 General Functionality

The plugin provides the backend configuration for SAML.
List of features implemented by this plugin:

- One or many SAMLBackends
- Supports reloadconfiguration clt without stopping unchanged SAMLBackends
- Can be started in addition to a normal AuthenticationService

4.12.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-saml
```

4.12.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/cloudplugins-saml.properties (page 56)

4.13 Package open-xchange-cloudplugins-trustedidentity-ldap

Cloud-Plugins Trusted Identity LDAP Support Support for storing Trusted Identity keys in LDAP using Cloud-Plugins.
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-plugins-trustedidentity (<<1.8.0)
open-xchange-plugins-trustedidentity (>=1.6.6)
```

4.13.1 General Functionality

This package provides a Trusted Identity key storage driver that looks up encrypted private keys from the OXaaS LDAP tree and decrypts them using on-disk storage keys.

4.13.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-trustedidentity-ldap
```

4.13.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/trustedidentity-ldap.properties (page 57)

4.14 Package open-xchange-cloudplugins-trustedidentity-ldap-tools

CLI Tools for Cloud-Plugins Trusted Identity LDAP Support CLI Tools for support for storing Trusted Identity keys in LDAP using Cloud-Plugins.
Version: 1.11.11-5
Type: Other

4.14.1 General Functionality

This package provides a Trusted Identity key storage driver that looks up encrypted private keys from the OXaaS LDAP tree and decrypts them using on-disk storage keys.

4.14.2 Installation

Install on nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-cloudplugins-trustedidentity-ldap-tools
```

4.15 Package open-xchange-oxaas-alias

OXaaS alias bundle This package implements OXaaS alias handling.
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-admin (<<7.10.7)
open-xchange-admin (>=7.10.6)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
```

4.15.1 General Functionality

The plugin is available to everyone on the installed system.
List of features implemented by this plugin:

- Alias are provided through internal and external APIs
- add and all requests are backed by a Tarent adapter
- del request is handled internally by using the internal provisioning interfaces
- max concurrent aliases are set by config-cascade aware setting com.openexchange.oxaas.aliasquota with default of 15.

4.15.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-oxaas-alias
```

4.15.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/oxaas-alias.properties (page 58)

4.16 Package open-xchange-oxaas-mail-notify-ws

OXaaS notification mail servlet bundle
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-imap (<<7.10.7)
open-xchange-imap (>=7.10.6)
open-xchange-smtp (<<7.10.7)
open-xchange-smtp (>=7.10.6)
```

4.16.1 General Functionality

The plugin is available to everyone that has correctly setup configuration.
List of features implemented by this plugin:

- Configuration for templates are done on a config-cascade base
- `com.openexchange.oxaas.mail.quota.notify.prefix` with default value `notify.oxaas.over.quota`
- `com.openexchange.oxaas.mail.welcomemail.notify.prefix` with default value `notify.oxaas.welcome.mail`
- `com.openexchange.oxaas.mail.removed.sent.spam.notify.prefix` with default value `notify.oxaas.disable.sent.spam`
- The above prefix is used for the templates where each template must have `${prefix}.${quotavalue}.[html|subject|text].tmpl` files present, in the case of the over quota mails.For the others, it is `${prefix}.[html|subject|text].tmpl`
- Default files are provided for 90% and 100% with the prefix `notify.oxaas.over.quota`.
- `com.openexchange.noreply.address` must be set via config-cascade, otherwise this feature won't work.
- `com.openexchange.oxaas.mail.(quota|welcomemail|removed.sent.spam).ignoreFooterImage` can be set via config-cascade to disable footerImage added as attachment to the mail, or by using `com.openexchange.oxaas.mail.ignoreFooterImage` that applies to all types

4.16.2 REST API

This package implements the OXaaS mail notification generation servlet which will return several mails via a REST API:

```
1 /api/oxaas/notification/mail/quota/{usercontext}/ (JSON body: {"quota_threshold":"..."})
2 /api/oxaas/notification/mail/welcomemail/{usercontext}/
3 /api/oxaas/notification/mail/disable_sent_spam_notification/{usercontext}/
```

4.16.3 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-oxaas-mail-notify-ws
```

4.16.4 Configuration

For details, please see appendix A
/opt/open-xchange/etc/oxaas-mail-notification-templates.properties (page 58)
/opt/open-xchange/etc/oxaas-drive-quota-notification.properties (page 59)

4.16.5 Templates

/opt/open-xchange/templates/notify.oxaas.over.quota.100.text.tmpl
/opt/open-xchange/templates/notify.oxaas.over.quota.100.subject.tmpl
/opt/open-xchange/templates/notify.oxaas.over.quota.100.html.tmpl
/opt/open-xchange/templates/notify.oxaas.over.quota.90.subject.tmpl
/opt/open-xchange/templates/notify.oxaas.over.quota.90.text.tmpl
/opt/open-xchange/templates/notify.oxaas.over.quota.90.html.tmpl

4.17 Package open-xchange-oxaas-mail-unread-ws

OXaaS mail custom mail servlet bundle This package implements OXaaS mail servlet to gather information via rest api.
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-imap (<<7.10.7)
open-xchange-imap (>=7.10.6)
```

4.17.1 General Functionality

API to featch the user related unread count for INBOX
List of features implemented by this plugin:

- API is reachable at `http://localhost:8009/preliminary/api/oxaas/mail/unread/<useridentifier>`
- API is secured by oxaas-mail-unread.properties where it is possible to add configuration for each brand that should have this feature enabled
- Set com.openexchange.oxaas.mail.unread.ws.basic.usernames=hosterone
- Set com.openexchange.oxaas.mail.unread.ws.basic.hosterone.brand=internalBrandForhosterone
- Set com.openexchange.oxaas.mail.unread.ws.basic.hosterone.password=verySecretPassword

4.17.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-oxaas-mail-unread-ws
```

4.17.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/oxaas-mail-unread.properties (page 59)

4.18 Package open-xchange-oxaas-mail-ws

OXaaS mail custom mail servlet bundle This package implements OXaaS mail servlet to gather information via rest api.
Version: 1.11.11-5
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.12)
open-xchange-cloudplugins (>=1.11.11)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.6)
open-xchange-imap (<<7.10.7)
open-xchange-imap (>=7.10.6)
```

4.18.1 General Functionality

This plugin provides a middleware restfull API to retrieve details of customerdata.
List of features implemented by this plugin:

- Main entry point is **/api/oxaas/mail**
- secured by basic auth mapped to customer brand
- **/api/oxaas/mail/{uid}/recentmails** returns latest 5 mails in INBOX
- **/api/oxaas/mail/{uid}/quota** returns current mailbox quota
- **/api/oxaas/mail/{uid}/newmessages** returns the number of new mails since last login
- **/api/oxaas/mail/{uid}** all of the above combined

4.18.2 Installation

Install on OX middleware nodes with package installer **apt-get** or **yum**:

```
<package installer> install open-xchange-oxaas-mail-ws
```

4.18.3 Configuration

For details, please see appendix A
/opt/open-xchange/etc/oxaas-mail.properties (page 59)

A Configuration Files

File 1 /opt/open-xchange/etc/plugin/mailstore-cloudplugins.properties

```
1 MAILSTORE_CLOUD_STORAGE=com.openexchange.admin.cloudplugins.storage.mysqlStorage.
      MailstoreCloudMySQLStorage
```

File 2 /opt/open-xchange/etc/cloudplugins-authentication.properties

```
# Regex to validate host HTTP Header value
com.openexchange.authentication.cloudplugins.host.regex= [0-9a-zA-Z.]*

#Regex to validate user name
com.openexchange.authentication.cloudplugins.user.regex= [0-9a-zA-Z.@]*

# Setting to control the start of the own authenticationService which would be used as a fallback in the Tracker if started
# Default: true
com.openexchange.authentication.cloudplugins.enableauthentication=true

# Comma seperated blacklist of hostnames that should not be handled by the default authenticationService
# Default: <empty>
com.openexchange.authentication.cloudplugins.authentication.blacklist=

# Setting to control the start of the CloudAuthenticationDriverTracker
# Default: true
com.openexchange.authentication.cloudplugins.enable.authentication.tracker=true

# Configure the LDAP lookup method to find users using their logins.
# It is the method that is used by default when there is no brand specifi
# configuration setting).
#
# Optional, defaults to "uid".
#
# Possible values:
# uid
#   attempts to find users by matching their login against the uid attribute.
# email
#   attempts to find users by matching their login against the alias attribute.
# auto
#   when the login contains a "@", the "email" method is used and when not,
#   the "uid" method is used
# uid-or-email
#   attempts to find users by matching their login against the uid and the alias
#   attributes (either may match)
#
# Note that this only applies to the default authentication driver. If there
# is a custom implementation that is used for a given brand, its behavior is
# not influenced by this properties.
#
# Example:
# com.openexchange.authentication.cloudplugins.authentication.uid.mode=auto

# One may define any number of such settings per brand by setting properties
# with the following format for their name:
#
# com.openexchange.authentication.cloudplugins.authentication.uid.mode.<brand>=<uid|email|auto|uid-or-email>
#
# For the list of possible values, please consult the documentation for
# com.openexchange.authentication.cloudplugins.authentication.uid.mode
#
# Optional, defaults to falling back to the method configured in
# com.openexchange.authentication.cloudplugins.authentication.uid.mode
#
# Note that this only applies to the default authentication service driver.
# If there is a custom implementation that is used for a given brand, its
# behavior is not influenced by these properties.
#
# Example:
#
# com.openexchange.authentication.cloudplugins.authentication.uid.mode.brand1=auto
# com.openexchange.authentication.cloudplugins.authentication.uid.mode.brand2=uid
```

File 3 /opt/open-xchange/etc/cloudplugins-master-auth-servlet.properties

```
#
# login name of httpauth user to access the master auth api
#
com.openexchange.cloudplugins.master.auth.httpauth.login=

#
# password of httpauth user
#
com.openexchange.cloudplugins.master.auth.httpauth.password=
```

File 4 /opt/open-xchange/etc/cloudplugins-cassandra.properties

```
# turn on/off cassandra integration
# possible values: true/false
com.openexchange.cloudplugins.useCassandra = false

# keyspace to use in cassandra
com.openexchange.cloudplugins.cassandraKeyspace = ox

# comma separated list of brands which logins should be recorded to cassandra
com.openexchange.cloudplugins.cassandra.loginrecorder.remoteipfor.brands=

# comma separated list of brands where alias creation and deletion time should be recorded
    to cassandra
com.openexchange.cloudplugins.cassandra.loginrecorder.createdeletealias.brands=

# comma separated list of brands where mail autoforward active flag should be set
com.openexchange.cloudplugins.cassandra.autoforward.record.brands=

# Default TTL (in seconds) of change history entries in the permission_change_history
   table.
#
# Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
# Default value: 90 days
com.openexchange.cloudplugins.cassandra.changehistory.default.ttl=90d

# Per-brand TTL (in seconds) of change history entries in the permission_change_history
   table.
#
# Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
# Example:
# com.openexchange.cloudplugins.cassandra.changehistory.brand.ttl.brand1=30d
# com.openexchange.cloudplugins.cassandra.changehistory.brand.ttl.brand2=24h
# com.openexchange.cloudplugins.cassandra.changehistory.brand.ttl.brand3=0

# String of text to use as the "reason" when permission changes are performed using the
# SOAP or RMI API.
# Defaults to "SOAP API" when empty:
com.openexchange.cloudplugins.cassandra.changehistory.soap.reason=
# Enables the cassandra login recorder
com.openexchange.cloudplugins.cassandra.useLoginRecorder = true
```

File 5 /opt/open-xchange/etc/cloudplugins.properties

```

# ldap url for read access
com.openexchange.cloudplugins.read.ldapurl=ldap://localhost:389

# admin dn for read access
com.openexchange.cloudplugins.read.binddn=cn=oxadmin,o=oxcs

# admin dn password for read access
```

```
com.openexchange.cloudplugins.read.bindpassword=

# ldap url for write access
com.openexchange.cloudplugins.write.ldapurl=ldap://localhost:389

# admin dn for write
com.openexchange.cloudplugins.write.binddn=cn=oxadmin,o=oxcs

# admin dn password for write access
com.openexchange.cloudplugins.write.bindpassword=

# tree for brands
com.openexchange.cloudplugins.branddn=ou=brands,o=oxcs

# tree for other stuff like mailstores
com.openexchange.cloudplugins.configdn=ou=config,o=oxcs

# tree for contexts
com.openexchange.cloudplugins.contextdn=ou=contexts,o=oxcs

# tree for class of service definitions
com.openexchange.cloudplugins.classofservicedn=ou=services,o=oxcs

# tree for trusted identity keys
com.openexchange.cloudplugins.trustedidentitykeydn=ou=keys,o=oxcs

# password for doveadm service
com.openexchange.cloudplugins.doveadmPassword=

# This timeout only works since Java 6 SE to time out waiting for a response.
com.openexchange.cloudplugins.read.timeout=10000

# Specifies the connect timeout (in milliseconds) when establishing a connection to the
    LDAP end-point
#
# Default is 5000 (5 seconds)
com.openexchange.cloudplugins.connect.timeout=5000

# Setting if user password hashs should be updated if the algorithm doesn't match the
    configured hash
com.openexchange.cloudplugins.password.updateUserPassword=false

# Algorithm to use to encrypt user passwords
# valid values are:
#   MD5
#   SMD5
#   SHA
#   SHA256
#   SHA384
#   SHA512
#   SSHA
#   SSHA256
#   SSHA-384 (old name for SSHA384)
#   SSHA384
#   SSHA512
#   CRYPT (general CRYPT identifier, uses CRYPT-SHA512 for password generation)
#   CRYPT-BCRYPT
#   CRYPT-MD5
#   CRYPT-SHA256
#   CRYPT-SHA512
# Default: CRYPT-SHA512
com.openexchange.cloudplugins.password.algorithm=SSHA

# Rounds for supported algorithms
# As of now only CRYPT-BCRYPT, CRYPT-SHA256, CRYPT-SHA512 support rounds
# Bcrypt is on a log scale while SHA256 and SHA512 are just rounds.
# Unset will use the algorithms default
# Bcrypt: 10
# SHA256: 5000
# SHA512: 5000
com.openexchange.cloudplugins.password.algRounds=

```

```
# Whether to enable timer metrics for password verifications,
# defaults to false
#
# When enabled, the service will record timer metrics for the
# duration of:
# - successful logins:
#   cloud-plugins-ldap/successfulLogins/<algorithm|rounds>
# - failed logins:
#   cloud-plugins-ldap/failedLogins/<algorithm|rounds>
# - user password hash updates:
#   cloud-plugins-ldap/passwordHashUpdates/<algorithm|rounds>
com.openexchange.cloudplugins.password.metrics=false

# how long should the random salts be
# only relevant for SMD5, SSHA, SSHA256, SSHA384, SSHA512
com.openexchange.cloudplugins.password.algorithm.saltlength=64

# HTTP Header from which the brand name is fetched
com.openexchange.cloudplugins.header=host

# can a user change the password without providing the old one?
# default is false
com.openexchange.capability.password_change_without_old_password=false

# Comma separated list of brands that where aliases should be moved into deleted tree
com.openexchange.cloudplugins.store.deleted.alias.brands=

# jndi ldap pool configuration
# see http://docs.oracle.com/javase/jndi/tutorial/ldap/connect/config.html for more
    information
com.openexchange.cloudplugins.pool=true
com.openexchange.cloudplugins.pool.initsize=1
com.openexchange.cloudplugins.pool.maxsize=20
com.openexchange.cloudplugins.pool.prefsize=10
com.openexchange.cloudplugins.pool.timeout=300000
com.openexchange.cloudplugins.pool.protocol=plain

# Allowed encryption methods for LDAP userPassword
# internal default: MD5,SHA,CRYPT,SSHA,SSHA384
com.openexchange.cloudplugins.setPasswordHash.allowedMethods=MD5,SHA,CRYPT,SSHA,SSHA384

# Comma separated list of brands that should be able to explicitly add maildomains to
    contexts
# This parameter is reloadable
com.openexchange.cloudplugins.explicitMailDomains.brands=

# Enable or disable the requirement to prefix context names with brandname_
# Default is false
# This parameter is reloadable
com.openexchange.cloudplugins.omit.contextname.prefix=false

# Enable Class of Service support.
#
# Note that this property is not reloadable, and changing it requires a restart.
#
# Defaults to: false
com.openexchange.cloudplugins.cos.enable=false

# Enable limiting the logging of invalid class of service properties
# that cannot be parsed.
#
# Since classes of service are loaded very often, their properties are also parsed
# often, and invalid property definitions can lead to excessive log pollution.
#
# Enabling this setting will limit the frequency of logging invalid properties,
# but each invalid property will be logged (as an error) at least once within the
# configured time period per class of service.
#
# The implementation uses a loading cache underneath, which causes some memory usage
# overhead, although limited by the maximum size configuration below.
#
# Note that it is not reloadable and requires a restart to change.
```

```
#
# Defaults to: false
com.openexchange.cloudplugins.cos.invalid.property.log.limiter.enable=false

# When enabled, configures the amount of time before a once logged combination of
# (class of service name, property) appears in the log again.
#
# Note that it is not reloadable and requires a restart to change.
#
# Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
# Defaults to 4h.
com.openexchange.cloudplugins.cos.invalid.property.log.limiter.quietPeriod=4h

# When enabled, the maximum amount of distinct combinations of (class of service name,
# property) that are kept track of to prevent excessive logging.
#
# This is to prevent excessive memory usage when keeping track.
#
# Note that it is not reloadable and requires a restart to change.
#
# Defaults to 256.
com.openexchange.cloudplugins.cos.invalid.property.log.limiter.maxSize=256

# Optional rate limiting for logging errors when failing to access LDAP to retrieve
# oxCloudClassOfService or oxCloudUser entries.
#
# Must be specified as a floating point rate value of allowed log entries
# per second (e.g. "1.0" means "log once per second", "0.01" means "log once per
# 100 seconds").
#
# Can also be set to "OFF", in which case no rate limiting is performed.
#
# Note that it is not reloadable and requires a restart to change.
#
# Example:
# com.openexchange.cloudplugins.cos.log.rate.limit=OFF
# com.openexchange.cloudplugins.cos.log.rate.limit=0.01
#
# Defaults to logging once per 5min (1 / 300 = 0.0033).
com.openexchange.cloudplugins.cos.log.rate.limit=0.0033

# When the oxCosDN attribute of UserEntity objects are changed through
    CloudManagementService,
# there are three options on how it is handled:
# - DISTRIBUTED: produces a distributed event that is broadcasted to all nodes and informs
#     them of injecting the new oxCosDN value into their respective cache (only if it
    already
#     exists in the cache); note that this setting also means that the implementation will
#     listen for remotely triggered events of that type
# - LOCAL: only updates the cache in the same App Suite node, locally in memory
# - OFF: does not update the cache at all
#
# Note that it is not reloadable and requires a restart to change.
#
# Examples:
# com.openexchange.cloudplugins.cos.cache.invalidation.mode=OFF
# com.openexchange.cloudplugins.cos.cache.invalidation.mode=LOCAL
#
# Defaults to DISTRIBUTED
com.openexchange.cloudplugins.cos.cache.invalidation.mode=DISTRIBUTED

###
### Class of Service Provisioning
###
### The following section configures the behavior of the provisioning APIs
### with regards to class of services.
### This applies to RMI, SOAP and command-line operations (createuser, changeuser) as
### well as to the Cloud-Plugins user management REST API under /oxaas/v1/admin/contexts
    /...
###

# Name of the user attribute that contains the class of service name.
```

```
#
# Class of services to apply to users must be specified as a user property with the
# namespace and name specified in the following configuration property.
# When the property is set to "OFF", class of service support is disabled in the provisioning
# APIs (RMI, SOAP and command-line -- but not in the Cloud-Plugins user management REST API).
#
# It must be specified in the form namespace//name
#
# Defaults to: cloud//service
#
# Note that the value of that user attribute may contain multiple names of classes of
# service, separated by ',' (whitespaces are stripped), e.g.:
# createuser ... --cloud/service=cloud_pim,cloud_security ...
#
# Examples:
# com.openexchange.cloudplugins.cos.provisioning.user.attribute=config//cos
# com.openexchange.cloudplugins.cos.provisioning.user.attribute=OFF
#
com.openexchange.cloudplugins.cos.provisioning.user.attribute=

# When com.openexchange.cloudplugins.cos.provisioning.user.attribute above is not "OFF",
# this property determines whether a class of service is mandatory when performing
# createuser operations.
# It is never mandatory for changeuser operations (the absence of the class of service
# information in a changeuser simply means that it is left untouched).
#
# Defaults to false.
com.openexchange.cloudplugins.cos.provisioning.mandatory=false

# When com.openexchange.cloudplugins.cos.provisioning.user.attribute above is not "OFF",
# this property determines whether a class of service may be empty in createuser and
# changeuser operations.
#
# Defaults to true.
com.openexchange.cloudplugins.cos.provisioning.allow.empty=true

# When com.openexchange.cloudplugins.cos.provisioning.user.attribute above is not "OFF",
# this property determines whether a missing class of service in createuser is to be
# understood as an empty list of classes of service.
# This only affects createuser, since the meaning of the absence of a list of classes of
# service in changeuser means leaving the classes of service value untouched.
# Note that enabling this property does not automatically mean that an empty list of
# classes of service is valid -- that is controlled separately using the property
# com.openexchange.cloudplugins.cos.provisioning.allow.empty.
#
# Defaults to true.
com.openexchange.cloudplugins.cos.provisioning.missing.means.empty=true

# When not empty, the following property defines an exhaustive list of class of service names that
# are valid and that must be matched.
# If the service names that are specified in provisioning operations are not part of this comma
# separated list, an error will abort the provisioning operation.
# Whitespaces are trimmed.
#
# To avoid using this list and perform LDAP lookups instead (or no validation at all),
# leave this property empty or commented out.
#
# Example:
# com.openexchange.cloudplugins.cos.provisioning.validate.list=cloud_pim,cloud_productivity
#
# Defaults to empty.
com.openexchange.cloudplugins.cos.provisioning.validate.list=

# When the validate.list property above is empty, the following property configures whether
# the validation of class of service names should be performed against LDAP, by querying the list
```

```
283 # of serviceName values that are defined below ou=services,o=oxcs
284 #
285 # Note that if both validate.list is empty and validate.ldap is false, service name
        validation
286 # will be disabled and any service name will be accepted.
287 #
288 # Defaults to true.
289 com.openexchange.cloudplugins.cos.provisioning.validate.ldap=true
290
291 ###
292 ### Class of Service Caching and Tuning
293 ###
294 ### The following section configures the behavior of the caching of service definition
295 ### and user service values.
296 ###
297
298 ##
299 ## Class of Service entries per user cache.
300 ##
301 ## Caches user class of service values that are queried from LDAP in the oxCosDN
302 ## attribute of oxCloudUser entries.
303 ##
304
305 # Fine-tune the concurrency level hint for the cache, which controls the amount of
306 # buckets and locks for concurrent thread access to the cache.
307 #
308 # Defaults to 32.
309 com.openexchange.cloudplugins.cos.user.cache.concurrencyLevel=32
310
311 # How long individual cached entries should be kept in the cache after having been
312 # queried.
313 #
314 # Since the config cascade performs multiple query operations to the service (multiple
315 # for each known capability), it is highly recommended to keep cache entries in memory
316 # for a few seconds at least.
317 #
318 # Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
319 # Defaults to 10m.
320 com.openexchange.cloudplugins.cos.user.cache.expireAfter=10m
321
322 # The maximum amount of oxCosDN attribute values of users that may be cached.
323 #
324 # When the maximum size is reached and entries need to be evicted, the last recently
325 # used entries will be removed.
326 #
327 # When left empty, no maximum cache size limit is applied, which is the default.
328 # Example:
329 # com.openexchange.cloudplugins.service.user.cache.maxSize=5000
330 com.openexchange.cloudplugins.cos.user.cache.maxSize=
331
332 # Whether to enable statistics and metrics for the oxCosDN attribute lookup cache.
333 #
334 # When enabled (true), the metrics endpoint will expose cache metrics for the
335 # cache named "cloud-plugins-ldap-cos-user".
336 #
337 # Defaults to false.
338 com.openexchange.cloudplugins.cos.user.cache.enableStats=false
339
340 # Whether the cache should be flushed when a reloadconfiguration operation is issued
341 # (regardless of whether another configuration parameter changed or not, through
342 # forced-reloadable).
343 #
344 # Defaults to false.
345 com.openexchange.cloudplugins.cos.user.cache.flushOnReload=false
346
347 ##
348 ## Class of Service definition cache.
349 ##
350 ## Caches class of service definitions (names to lists of properties) that are queried
351 ## from LDAP in oxCloudClassOfService entries below ou=services,o=oxcs.
352 ##
353
```

```
# Fine-tune the concurrency level hint for the cache, which controls the amount of
# buckets and locks for concurrent thread access to the cache.
#
# Defaults to 32.
com.openexchange.cloudplugins.cos.definition.cache.concurrencyLevel=32

# The maximum amount of oxCloudClassOfService values that may be cached.
#
# When the maximum size is reached and entries need to be evicted, the last recently
# used entries will be removed.
#
# When left empty, no maximum cache size limit is applied, which is the default.
# Example:
# com.openexchange.cloudplugins.cos.definition.cache.maxSize=100
com.openexchange.cloudplugins.cos.definition.cache.maxSize=

# How long individual cached entries should be kept in the cache after having been
# queried.
#
# Since the config cascade performs multiple query operations to the service (multiple
# for each known capability), it is highly recommended to keep cache entries in memory
# for a few seconds at least.
#
# Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
# Defaults to 10m.
com.openexchange.cloudplugins.cos.definition.cache.expireAfter=10m

# Whether to enable statistics and metrics for the oxCloudClassOfService lookup cache.
#
# When enabled (true), the metrics endpoint will expose cache metrics for the
# cache named "cloud-plugins-ldap-cos-definition".
#
# Defaults to false.
com.openexchange.cloudplugins.cos.definition.cache.enableStats=false

# Whether the cache should be flushed when a reloadconfiguration operation is issued
# (regardless of whether another configuration parameter changed or not, through
# forced-reloadable).
#
# Defaults to false.
com.openexchange.cloudplugins.cos.definition.cache.flushOnReload=false

# Fine-tune the concurrency level hint for the cache, which controls the amount of
# buckets and locks for concurrent thread access to the cache.
#
# Defaults to 32.
com.openexchange.cloudplugins.cos.listall.cache.concurrencyLevel=32

# How long individual cached entries should be kept in the cache after having been
# queried.
#
# Since the config cascade queries the list of all the known capabilities at
# every user login, it is highly recommended to keep cache entries in memory
# for a certain time, especially since class of service definitions are not likely
# to change often and, if so, don't need to be applied immediately but can wait
# until after the cache expires.
#
# Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
# Defaults to 10m.
com.openexchange.cloudplugins.cos.listall.cache.expireAfter=10m

# Whether to enable statistics and metrics for the list of all properties defined
# in oxCloudClassOfService properties attributes.
#
# When enabled (true), the metrics endpoint will expose cache metrics for the
# cache named "cloud-plugins-ldap-cos-listall".
#
# Defaults to false.
com.openexchange.cloudplugins.cos.listall.cache.enableStats=false

# Whether the cache should be flushed when a reloadconfiguration operation is issued
# (regardless of whether another configuration parameter changed or not, through
```

```
# forced-reloadable).
#
# Defaults to false.
com.openexchange.cloudplugins.cos.listall.cache.flushOnReload=false

# Whether to enable the announcements feature
# Not reloadable.
#
# Default: false
com.openexchange.cloudplugins.announcements.enabled=true

# The default group name to access the global database.
# See /opt/open-xchange/etc/globaldb.yml for more information and
# configuration possibilities
# Not reloadable.
#
# Default: default
com.openexchange.cloudplugins.announcements.globaldbContextGroup=default

# The list of available settings for spamlevels.  It defaults to "low,medium,high"
# Please note: this is mandatory and must be set
com.openexchange.cloudplugins.spamlevels=low,medium,high

# the default level to use from the above spamlevels settings.  It defaults to medium
# Please note: this is mandatory and must be set
com.openexchange.cloudplugins.spamlevels.default=medium

# The antivirus setting. it defaults to true
com.openexchange.cloudplugins.antivirus.default=true

# Whether to enable the Cloud-Plugins application password storage driver which uses
# LDAP and Cassandra.
#
# Config-cascade aware if
# com.openexchange.cloudplugins.authentication.application.storage.configCascadeAware=true
#
# To turn off Cloud-Plugins application password support altogether and in the most
# performant fashion, set both
# com.openexchange.cloudplugins.authentication.application.storage.enabled=false
# and
# com.openexchange.cloudplugins.authentication.application.storage.configCascadeAware=
    false
#
# Optional, defaults to true
com.openexchange.cloudplugins.authentication.application.storage.enabled=true

# Whether to use the Config Cascade to look up the configuration properties below
# for the application password support in Cloud-Plugins.
#
# When disabled, it allows for optimizations such as not performing an LDAP search
# when adding a password and skipping potential database queries when looking up
# properties through the context and user scopes of the config cascade.
#
# Unless you have a need to have different settings for the following properties
# per context sets, contexts or users, it is advised to keep the config cascade
# support turned off:
# - com.openexchange.cloudplugins.authentication.application.storage.enabled
# - com.openexchange.cloudplugins.authentication.application.storage.enabledApps
# - com.openexchange.cloudplugins.authentication.application.storage.storeUserPassword
#
# Optional, defaults to false
com.openexchange.cloudplugins.authentication.application.storage.configCascadeAware=false

# Comma-separated list of appTypes for which the Cloud-Plugins application password
# storage is enabled.
#
# Leave empty or Use '*' for all appTypes.
#
# Note that they are compared in a case-insensitive fashion.
#
# Config-cascade aware if
# com.openexchange.cloudplugins.authentication.application.storage.configCascadeAware=true
```

```
#
# Examples:
# com.openexchange.cloudplugins.authentication.application.storage.enabledApps=*
# com.openexchange.cloudplugins.authentication.application.storage.enabledApps=webdav,
    caldav
#
# Optional, defaults to all appTypes.
com.openexchange.cloudplugins.authentication.application.storage.enabledApps=

# When enabled, encrypts and stores the user's regular session password in the application
# password authentication.
# Required if applications accessing external systems like the mail server need their
    individual
# credentials rather than master- or OAuth-based authentication.
#
# Config-cascade aware if
# com.openexchange.cloudplugins.authentication.application.storage.configCascadeAware=true
#
# Optional, defaults to false.
com.openexchange.cloudplugins.authentication.application.storage.storeUserPassword=false
```

File 6 /opt/open-xchange/etc/doveadm-config.properties

```
# Api secret
com.openexchange.cloudplugins.dovecot.apiSecret=

# Dovecot port
com.openexchange.cloudplugins.dovecot.port=8080

# Protocol to use
com.openexchange.cloudplugins.dovecot.protocol=http://

# Dovecot host
com.openexchange.cloudplugins.dovecot.host=localhost

# Path to dovecot commands
com.openexchange.cloudplugins.dovecot.path=/doveadm/v1

# How many contexts per request should be transmitted
com.openexchange.cloudplugins.dovecot.contextChunks=100

# Max number of http connections
com.openexchange.cloudplugins.dovecot.maxConnections=100

# Max number of http connections per host
com.openexchange.cloudplugins.dovecot.maxConnectionsPerHost=100

# The connection timeout in milliseconds
com.openexchange.cloudplugins.dovecot.connectionTimeout=5000

# The socket read timeout in milliseconds
com.openexchange.cloudplugins.dovecot.socketReadTimeout=15000
```

File 7 /opt/open-xchange/etc/nginx-auth-servlet.properties

```
# Regex to validate brandName value
com.openexchange.cloudplugins.nginx.auth.servlet.brandName.regex= [0-9a-zA-Z.]*

# Regex to validate user name
com.openexchange.cloudplugins.nginx.auth.servlet.uid.regex= [0-9a-zA-Z.@]*

# Configure the LDAP lookup method to find users using their logins.
# It is the method that is used by default when there is no brand specifi
```

```
# configuration setting).
#
# Optional, defaults to "uid".
#
# Possible values:
# uid
#   attempts to find users by matching their login against the uid attribute.
# email
#   attempts to find users by matching their login against the alias attribute.
# auto
#   when the login contains a "@", the "email" method is used and when not,
#   the "uid" method is used
# uid-or-email
#   attempts to find users by matching their login against the uid and the alias
#   attributes (either may match)
#
# Note that this only applies to the default authentication driver. If there
# is a custom implementation that is used for a given brand, its behavior is
# not influenced by this properties.
#
# Example:
# com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode=auto

# One may define any number of such settings per brand by setting properties
# with the following format for their name:
#
# com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode.<brand>=<uid|email|auto|uid-or
    -email>
#
# For the list of possible values, please consult the documentation for
# com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode
#
# Optional, defaults to falling back to the method configured in
# com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode
#
# Note that this only applies to the default authentication driver. If there
# is a custom implementation that is used for a given brand, its behavior is
# not influenced by these properties.
#
# Example:
#
# com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode.brand1=auto
# com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode.brand2=uid

# For Application Password authentication, define which password appTypes are acceptable
# per authentication protocol (imap, pop3, smtp).
#
# Values are a list of comma-separated appTypes.
# The appType values depend on the appTypes that are defined in .app-password-apps.yml
#
# When the Nginx servlet authenticates a user for a given protocol (imap, pop3 or smtp),
# it looks up the value of
# com.openexchange.cloudplugins.nginx.auth.servlet.apppassword.apptype.${protocol}
# When that value is empty or undefined, Application Password authentication is not
# attempted.
# When the list of values contains a value "*", it does not restrict the Application
# Passwords on the appType.
# When the list of values does not contain "*", then it restricts the search of
# potentially matching Application Passwords in LDAP to the specified appTypes.
#
# To block a protocol from using Application Passwords altogether, define an empty
# value for that protocol.
#
# Note that the protocol value is validated and must be one of "imap", "pop3" or "smtp";
# if not, it causes an error and a failure to read the configuration.
#
# Examples:
#
# com.openexchange.cloudplugins.nginx.auth.servlet.apppassword.apptype.imap=mail,imap
#   When authenticating an IMAP login, only Application Passwords having an appType
#   (applicationPasswordAppType LDAP attribute) of "mail" or "imap" are considered
#   for attempting authentication.
```

```
#
# com.openexchange.cloudplugins.nginx.auth.servlet.apppassword.apptype.smtp=
#   Since an empty value is defined for the protocol smtp, SMTP logins will never be
#   authenticated against Application Passwords.
#
com.openexchange.cloudplugins.nginx.auth.servlet.apppassword.apptype.imap=mail,imap
com.openexchange.cloudplugins.nginx.auth.servlet.apppassword.apptype.pop3=mail,pop3
com.openexchange.cloudplugins.nginx.auth.servlet.apppassword.apptype.smtp=mail,smtp
```

File 8 /opt/open-xchange/etc/cloudquotaservice.properties

```
# Identifier of the default quota driver to use when not
# superseded by a per-brand configuration setting (see below).
#
# This configuration property is optional.
# When it is not set, it will attempt to use the "dovecot"
# driver, if available (installed).
# If the "dovecot" driver is not installed, it will use the
# highest ranked driver that is installed.
#
# For a deterministic approach, it is recommended to set
# a value for this configuration setting.
#
# Example:
# com.openexchange.cloudplugins.quota.default.driver=cassandra
com.openexchange.cloudplugins.quota.default.driver=

# Override the quota driver per brand.
#
# Use property names that start with
# "com.openexchange.cloudplugins.quota.driver."
# followed by the brand name (not the complete DN
# but just the brand name).
#
# These are optional and, if not defined, will always fall
# back to the driver configured in the property
# com.openexchange.cloudplugins.quota.default.driver
#
# Example:
# com.openexchange.cloudplugins.quota.driver.brand1=cassandra
# com.openexchange.cloudplugins.quota.driver.otherbrand=cassandra
#
```

File 9 /opt/open-xchange/etc/cloudquotaservice-cassandra.properties

```
# Use the legacy "type" value for lookups in the Cassandra
# quota table.
# In recent deployments, the "type" parameter is "dovecot_mail",
# but in previous installments, the "type" used to be null.
# Setting this property to true will use null for the "type"
# query parameter.
#
# This property is optional and when not set, defaults to false.
#
# Example:
# com.openexchange.cloudplugins.quota.use.legacy.type=true
#

com.openexchange.cloudplugins.quota.use.legacy.type=false

# Unified quota performance optimizations.
#
# The default behavior is to check whether every single user
```

```
# who's usage quota is retrieved has unified quota enabled or not,
# and return data accordingly.
#
# In most use cases, if not all, unified quota will be enabled or
# disabled uniformly
# - globally for a platform,
# - or globally for a brand,
# - or for a whole context,
# in which case the implementation can avoid or minimize the amount
# of queries it needs to perform in order to determine how to
# calculate the quota usage, depending on whether unified quota
# is enabled or disabled for a user.
#
# Two configuration settings govern this behavior:
# - the default behavior which is used for every brand on a platform
#   (com.openexchange.cloudplugins.quota.unified.quota)
# - per-brand behaviors which take precedence over the default
#   (com.openexchange.cloudplugins.quota.unified.quota.<brand name>)
#
# Each of those settings can have one of the following values:
#
# always
#   the implementation will assume that unified quota is enabled for
#   all contexts within the brand, or for all contexts within all
#   brands if applied to the default setting
#
# never
#   the implementation will assume that unified quota is disabled for
#   all contexts within the brand, or for all contexts within all
#   brands if applied to the default setting
#
# context
#   the implementation will only check whether unified quota has
#   been enabled for the context the user(s) are in, and not for
#   each individual user, assuming that all users within the same
#   context are always configured uniformly regarding unified quota,
#   be it enabled or disabled
#
# user
#   the implementation will make no assumptions and check whether
#   unified quota is enabled or not for every individual user -- this
#   is the safest setting, which is why it is the default, but also
#   the slowest and should be avoided if possible
#
# The default setting can be configured using the property
# com.openexchange.cloudplugins.quota.unified.quota=...
#
# It is optional and defaults to "user" (as explained above) if
# omitted, commented out or left empty.
#
# Example:
# com.openexchange.cloudplugins.quota.unified.quota=never
com.openexchange.cloudplugins.quota.unified.quota=

# Per-brand settings can be configured using the following
# prefix, followed by the name of the brand:
# com.openexchange.cloudplugins.quota.unified.quota.<brand name>=...
#
# Example:
# com.openexchange.cloudplugins.quota.unified.quota.my_brand=always
# com.openexchange.cloudplugins.quota.unified.quota.my_other_brand=context
#
# Those are obviously optional and default to using the default
# setting above (which, in turn, when omitted, defaults to "user").
#
```

File 10 /opt/open-xchange/etc/cloudplugins_report.properties

```
# Where the report and its parts should be stored
com.openexchange.cloudplugins.report.storagePath=/tmp

# How many contexts can be stored in memory before writing them on hdd
com.openexchange.cloudplugins.report.maxChunkSize=200

# How many parallel threads can work on the report
com.openexchange.cloudplugins.report.maxThreadPoolSize=20

# Report thread priority
com.openexchange.cloudplugins.report.threadPriority=1

# Max number of entities that will be included in an ldap search
com.openexchange.cloudplugins.report.maxLdapChunks=20000
```

File 11 /opt/open-xchange/etc/cloudplugins-antiphishing-vadesecure-ldap.properties

```

# Setting to change the VadeSecure connector identifier referenced in plugins-antiphishing
    .properties / com.openexchange.plugins.antiphishing.connector
# Default: "cloudplugins_antiphishing_vadesecure_ldap"
# Config-cascade aware: true
# Lean: true
com.openexchange.cloudplugins.antiphishing.vadesecure.ldap.identifier=
    cloudplugins_antiphishing_vadesecure_ldap
```

File 12 /opt/open-xchange/etc/cloudplugins-blackwhitelist-ldap.properties

```
# Identifier of this blackwhitelist connector: cloudplugins_blackwhitelist_ldap
# hostname of ldap server
com.openexchange.cloudplugins.blackwhitelist.connector.ldap.uri=ldap-fqhn.example.com

# ldap port
com.openexchange.cloudplugins.blackwhitelist.connector.ldap.port=389

# ldap user
com.openexchange.cloudplugins.blackwhitelist.connector.ldap.user=cn=oxadmin,o=oxcs

# ldap password
com.openexchange.cloudplugins.blackwhitelist.connector.ldap.passwd=

# ldap maximum pool size
com.openexchange.cloudplugins.blackwhitelist.connector.ldap.pool.size=10

# ldap max requests before connection is closed
# can be set to -1 to be disabled
com.openexchange.cloudplugins.blackwhitelist.connector.ldap.pool.max.requests=2000

# ldap max lifetime in seconds for each connection in the pool
com.openexchange.cloudplugins.blackwhitelist.connector.ldap.pool.max.lifetime=120

# config to enable LDAP SSL connection over ldaps
com.openexchange.cloudplugins.blackwhitelist.connector.ldap.useSSL=false

# Setting to check if memory backed test mock should be started
# This connector is identified by cloudplugins_blwl_test
# Default: false
com.openexchange.cloudplugins.blackwhitelist.connector.ldap.test=false
```

File 13 /opt/open-xchange/etc/cloudplugins-forwards.properties

```
# Defines whether the forward REST API should be enabled or not.
#
# This parameter is optional and defaults to "false" (disabled).
#
# Example:
# com.openexchange.cloudplugins.admin.forwards.ws.enabled=true
com.openexchange.cloudplugins.admin.forwards.ws.enabled=false
```

File 14 /opt/open-xchange/etc/cloudplugins-keycloak.properties

```
# The token endpoint identified by the client
#
# Must be set for each client, default value: ""
#
# Example:
# com.openexchange.cloudplugins.keycloak.oauth.exampleClient.tokenEndpoint=http://
    localhost:8080/auth/realms/demo/protocol/openid-connect/token
com.openexchange.cloudplugins.keycloak.oauth.[client].tokenEndpoint=

# The clientId, if left empty, no clientId will be used
#
# Optional, default value: ""
#
# Example:
# com.openexchange.cloudplugins.keycloak.oauth.exampleClient.clientId=customerClient
com.openexchange.cloudplugins.keycloak.oauth.[client].clientId=

# The default-client used for the CloudAuthenticationDriver
com.openexchange.cloudplugins.keycloak.oauth.default-client.clientId=

# The client secret. Must be provided if clientId is set.
#
# Optional, default value: ""
#
# Example:
# com.openexchange.cloudplugins.keycloak.oauth.exampleClient.clientSecret=123123
com.openexchange.cloudplugins.keycloak.oauth.[client].clientSecret=

# The default-client secret used for the CloudAuthenticationDriver
com.openexchange.cloudplugins.keycloak.oauth.default-client.clientSecret=

# Max connections
#
# Optional, default value: 100
com.openexchange.cloudplugins.keycloak.oauth.maxConnections=100

# Max connections per host
#
# Optional, default value: 100
com.openexchange.cloudplugins.keycloak.oauth.maxConnectionsPerHost=100

# Connection timeout in ms
#
# Optional, default value in ms: 3000
com.openexchange.cloudplugins.keycloak.oauth.connectionTimeout=3000

# Socket read timeout in ms
#
# Optional, default value in ms: 6000
com.openexchange.cloudplugins.keycloak.oauth.socketReadTimeout=6000

# Refresh time in ms before expiry date
#
# Optional, default value is ms: 60000
com.openexchange.cloudplugins.keycloak.oauth.refreshTime=60000
```

```
55
56  # Enables the keycloak cloud-plugins CloudAuthenticationDriver.
57  # If either of the following properties is set, it is not required to enable this property
58  #     com.openexchange.mail.authType=xoauth2 or oauthbearer
59  #     com.openexchange.mail.filter.preferredSaslMech=OAUTHBEARER or XOAUTH2
60  #
61  # Default: false
62  com.openexchange.cloudplugins.keycloak.oauth.authentication.enabled=false
63
64  # Comma seperated blocklist of hostnames that should not be handled by the keycloak
        CloudAuthenticationDriver
65  # Default: <empty>
66  com.openexchange.cloudplugins.keycloak.oauth.authentication.blocklist=
67
68  # Sets the client identifier for the CloudAuthenticationDriver
69  # Internally will use the value "default-client" as a fallback
70  #
71  # Default: ""
72  com.openexchange.cloudplugins.keycloak.oauth.authentication.client=default-client
73
74  # One may set different clients on a brand base
75  #
76  # com.openexchange.cloudplugins.keycloak.oauth.authentication.client.<brand>=<client>
77  # Example:
78  #
79  # com.openexchange.cloudplugins.keycloak.oauth.authentication.client.brand1=cloudplugins-
        keycloak-custom-client
80  # com.openexchange.cloudplugins.keycloak.oauth.authentication.client.brand2=brand-specific
        -client
81
82  # Sets the response identifier for the CloudAuthenticationDriver
83  #
84  # Special case: oxUserId@oxContextId enables lookup for the two keys oxUserId and
        oxContextId
85  #               Afterwards they are again handled as oxUserId@oxContextId
86  #
87  # Default: "preferred_username"
88  com.openexchange.cloudplugins.keycloak.oauth.authentication.response.identifier=
        preferred_username
89
90  # One may set different response identifiers on a brand base
91  #
92  # com.openexchange.cloudplugins.keycloak.oauth.authentication.response.identifier.<brand
        >=<client>
93  # Example:
94  #
95  # com.openexchange.cloudplugins.keycloak.oauth.authentication.response.identifier.brand1=
        email
96  # com.openexchange.cloudplugins.keycloak.oauth.authentication.response.identifier.brand2=
        alias
97
98  # Configure the LDAP lookup method to find users using their logins.
99  # It is the method that is used by default when there is no brand specific
100 # configuration setting).
101 #
102 # Optional, defaults to "uid".
103 #
104 # Possible values:
105 # uid
106 #   attempts to find users by matching their login against the uid attribute.
107 # email
108 #   attempts to find users by matching their login against the alias attribute.
109 # auto
110 #   when the login contains a "@", the "email" method is used and when not,
111 #   the "uid" method is used
112 # uid-or-email
113 #   attempts to find users by matching their login against the uid and the alias
114 #   attributes (either may match)
115 #
116 # Note that this only applies to the keycloak authentication driver. If there
117 # is a custom implementation that is used for a given brand, its behavior is
118 # not influenced by this properties.
```

```
#
# Example:
# com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode=auto

# One may define any number of such settings per brand by setting properties
# with the following format for their name:
#
# com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode.<brand>=<uid|email|
    auto|uid-or-email|userid-contextid>
#
# For the list of possible values, please consult the documentation for
# com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode
#
# Optional, defaults to falling back to the method configured in
# com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode
#
# Note that this only applies to the default authentication service driver.
# If there is a custom implementation that is used for a given brand, its
# behavior is not influenced by these properties.
#
# Example:
#
# com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode.brand1=auto
# com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode.brand2=uid
```

File 15 /opt/open-xchange/etc/cloudplugins-loginproxy-ws.properties

```
# Maximum amount of login proxy lookup requests per second per source IP address.
# May be a decimal number.
#
# Optional, default value: 25.0
#
# Example:
# com.openexchange.cloudplugins.login.proxy.maxRequestsPerSecond=50.0
com.openexchange.cloudplugins.login.proxy.maxRequestsPerSecond=25.0

# Maximal time window, in milliseconds: after a given source IP address has not accessed
# the login proxy lookup API, its number of requests per second rate is reset.
#
# Optional, default value: 300000
#
# Example:
# com.openexchange.cloudplugins.login.proxy.maxRateTimeWindow=60000
com.openexchange.cloudplugins.login.proxy.maxRateTimeWindow=300000

# Strategy to use for reacting to the inability to access the API for a given source
# IP address due to surpassing the maxRequestsPerSecond rate.
#
# Format: it must be one of:
# * fail-fast
# * block
# * timeout:...
#
# fail-fast
#   if the rate limit is exceeded, the API will respond with a 403 Forbidden
# block
#   if the rate limit is exceeded, the API will block infinitely until the rate limit
#   allows for another request to be performed
# timeout:...
#   block until the specified timeout is reached, after which the API responds with a
#   403 Forbidden
#   The value after "timeout:" consists of a number followed by a time unit, examples:
#   - timeout:400s ---> 400 seconds
#   - timeout:1m ------>   1 minute
#   - timeout:2000ms -> 2000 milliseconds
#
# Optional, default value: timeout:5s
#
```

```
# Example:
# com.openexchange.cloudplugins.login.proxy.strategy=timeout:10s
com.openexchange.cloudplugins.login.proxy.strategy=timeout:5s

# The default URL to redirect to when the user is not marked as not migrated
# and the identifierof the user is not mapped in cloudplugins-loginproxy-forward.yaml
# and the brand does not have a default redirect in cloudplugins-loginproxy-forward.yaml
#
# Example:
# com.openexchange.cloudplugins.login.proxy.default.redirect=https://example.com/mail
# Default: not set
com.openexchange.cloudplugins.login.proxy.default.redirect=
```

File 16 /opt/open-xchange/etc/cloudplugins-loginproxy-forward.yaml

```
# This file contains mappings of brand, identifiers and redirect urls.
# It must be a YAML mapping, where
# * the key is the brand
# * the value is a list of properties with key, value
---
'brand1':
- identifier: https://loginpage1.example.com
- another_identifier: https://loginpage2.example.org
- default_redirect: https://default.example.com
'brand2':
- my_ident: https://loginpage1.example.com
- default_redirect: https://default.example.com
'brand3':
- some_other_identifier: https://loginpage1.example.com
```

File 17 /opt/open-xchange/etc/mailfilter-interceptor-drivers.properties

```
# This is the CloudPlugins MailFilterInterceptorDriver configuration
#
# Enable drivers by adding at least one brand in the brands property for that driver
# on the server level configuration. If no brand exists, the driver will not be registered
#
# Some drivers may also have additional configurations


##### Driver Brand Lists #####
# Comma delimited lists

# Brands that the RedirectStatusDriver should be enabled for
#
# Optional - default is no brands
com.openexchange.cloudplugins.mailfilter.intercept.drivers.redirect.status.driver.brands=

# Brands that the RedirectBlacklistDriver should be enabled for
#
# Optional - default is no brands
com.openexchange.cloudplugins.mailfilter.intercept.drivers.redirect.blacklist.driver.
    brands=

##### End Driver Brand Lists #####



##### Driver Specific configurations #####

### Redirect Blacklist Driver ###
#
# Set to true to enable config cascade for all properties of the Redirect Blacklist Driver
```

```
    .
# This should be used to set different configurations per brand or an even lower level.
# This property is NOT config cascade aware as it is used to control use of it.
# Even the driver brand list property can be config cascade if this is enabled which would be useful
# to enable it for a brand, but disable it for some users
#
# Optional - default is false
com.openexchange.cloudplugins.mailfilter.intercept.drivers.redirect.blacklist.driver.configcascade.enable=false
#
# The comma+space delimited list of regular expressions that are blacklisted for mail filter redirects.
# Java regular expressions are supported here, so non regex characters must be escaped.
# Example: abuse@.*, spam@domain\.com  <-- '.' has been escaped to match only '.' and not any char
# See for regex contructs: https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html
# WARNING: You must use ", " (comma and a space) as the separator to separate the regexs
#
# Optional - default is empty which should only be left if this driver is not used
com.openexchange.cloudplugins.mailfilter.intercept.drivers.redirect.blacklist.driver.blacklist=
#
# Set to true to only blacklist true autoforward rules which are defined by having the "autoforward" flag.
# If set to false, then all redirect rules will be checked for the blacklist addresses
#
# Optional - default is true
com.openexchange.cloudplugins.mailfilter.intercept.drivers.redirect.blacklist.driver.autoforward.only=true
#
### End Redirect Blacklist Driver ###

##### End Driver Specific configurations #####
```

File 18 /opt/open-xchange/etc/cloudplugins-masterauth.properties

```
# Configure the master password per brand.
#
# If the brand is configured here, master auth will be enabled and it will
# override normal authentication - meaning that users would not be able to
# login if normal authentication is used. This would not impact SSO.
#
# Example:
#
# com.openexchange.authentication.cloudplugins.brand.master.auth.brand1.password=secret
# com.openexchange.authentication.cloudplugins.brand.master.auth.brand2.password=secret
```

File 19 /opt/open-xchange/etc/cloudplugins-mx-checker.properties

```
# The max number of entries to cache.
# Caching is used for brand lookup and valid DNS results.
# Use 0 for no cache.
#
# Default: 10000
#
com.openexchange.cloudplugins.mx.checker.cache.max=10000

# The amount of time in seconds after a value is written to the cache
# until it expires.
# Caching is used for brand lookup and valid DNS results.
# Use 0 for no cache.
```

```
#
# Default: 3600
#
com.openexchange.cloudplugins.mx.checker.cache.expire.seconds=3600

# URL templates that are sent as the actionUrl depending on the
# MX/SPF checks.
# All of those are config-cascade aware and are looked up with the
# userId and contextId.
# If the property value is empty, it is considered undefined.
# The URL templates may contain the following placeholders that will
# be resolved before sending them as the actionUrl in the response:
# ${brand}: the toplevel brand
# ${domain}: the domain of the email of the user
# ${userId}: the numeric user ID
# ${contextId}: the numeric context ID
# ${login}: the login that was used during authentication
# ${locale}: the user's locale (e.g. "de-DE"), or ""
# ${language}: the user's preferred language (e.g. "fr"), or ""
# Those can be used anywhere in the URL since it is a simple
# text replacement.

# URL template that is sent as the actionUrl when both the MX check
# and the SPF check are BAD, or as a fallback when only MX or SPF are
# BAD but their more specific properties .invalidmx and .invalidspf are
# not defined:
com.openexchange.cloudplugins.mx.checker.url.invalid=

# URL template that is sent as the actionUrl for the MX check result
# when the MX check is BAD but the SPF check is not:
com.openexchange.cloudplugins.mx.checker.url.invalidmx=

# URL template that is sent as the actionUrl for the SPF check result
# when the SPF check is BAD but the MX check is not:
com.openexchange.cloudplugins.mx.checker.url.invalidspf=

# URL template that is sent as the actionUrl for either check when it
# is in propagation status:
com.openexchange.cloudplugins.mx.checker.url.propagating=
```

File 20 /opt/open-xchange/etc/cloudplugins-oidc.properties

```
# The properties for cloud setup OIDC bundle

# The general oidc property to enable or disable the core oidc registry
com.openexchange.oidc.enabled=true

### General settings
##############################################################################################
# Regex to validate user name
com.openexchange.cloudplugins.oidc.user.regex=[0-9a-zA-Z.@]*

# All properties below can also be assigned to an identifier by configuring them as
# com.openexchange.cloudplugins.oidc.<identifier>.key
#
# If a property is not set for an identifier, the key without the identifier is used as a
    default
#
# Example:
#    com.openexchange.cloudplugins.oidc.example.id=someValue
#    com.openexchange.cloudplugins.oidc.brand=someBrand
#    com.openexchange.cloudplugins.oidc.example.brand=
#    example brand => someBrand
#
# !! Note !!
#    The list of possible OIDCBackends is identified by the property
#    com.openexchange.cloudplugins.oidc.enabled.<identifier>
#
```

```
# Example:
#    com.openexchange.cloudplugins.oidc.enabled=true
#    com.openexchange.cloudplugins.oidc.enabled.myIdentifier=true
#    com.openexchange.cloudplugins.oidc.enabled.moreIdentifier=true
#
#    It is possible to disable certain OIDCBackends by configuration
#
#    com.openexchange.cloudplugins.oidc.enabled is also valid and will be used as an empty identifier
#    An empty identifier will use all properties set in this config file.
#
# com.openexchange.cloudplugins.oidc.enabled.<identifier>=true
com.openexchange.cloudplugins.oidc.enabled=false

### OIDC Specific configuration
##########################################################################################
# All properties mentioned at https://documentation.open-xchange.com/components/middleware/config/7.10.2/#mode=features&feature=OpenID
# can be assigned to each OIDCbackend.
#
# !! NOTE !!
#     The default prefix is not 'com.openexchange.oidc'
#     'com.openexchange.cloudplugins.oidc'
#     is used instead meaning that there is an additional
#     'cloudplugins' in between to differentiate from the general oidc configuration.
#
# Example:
#    com.openexchange.oidc.clientId will not be evaluated
#    com.openexchange.oidc.cloudplugins.clientId is the correct default key
#
# The id inside the jwt token response which holds the userinformation
#
# The search for the id is done in the attributeStatement if nothing is configured
# possible configuration values are:
#    <not_set>      // the subject is used
#    claim:key      // a claim with the identifier <key> is used
#    key            // a claim with the identifier <key> is used
#
# In any case, the id must match the uid used to provision the user
# com.openexchange.cloudplugins.oidc.<identifier>.id=
com.openexchange.cloudplugins.oidc.id=

# The ldap lookup used by this OIDCBackend
#
# Configure the LDAP lookup method to find users using their identifiers.
#
#
# Possible values:
# uid
#   attempts to find users by matching their login against the uid attribute.
# email
#   attempts to find users by matching their login against the alias attribute.
# auto
#   when the login contains a "@", the "email" method is used and when not,
#   the "uid" method is used
# uid-or-email
#   attempts to find users by matching their login against the uid and the alias
#   attributes (either may match)
#
# This property is optional.
# Default: uid
# com.openexchange.cloudplugins.oidc.<identifier>.ldapLookup=
com.openexchange.cloudplugins.oidc.ldapLookup=


# The brand to use for OXaaS LDAP authentication operations.
# Uses the host name when empty or not set.
#
# This property is mandatory.
# Default: <empty>
# com.openexchange.cloudplugins.oidc.<identifier>.brand=
com.openexchange.cloudplugins.oidc.brand=
```

```
96
97  # The authentication method used for the token endpoint.
98  # Can be a selection of:
99  #    basic
100 #    post
101 #
102 # This property is optional.
103 # Default: basic if not set
104 # com.openexchange.cloudplugins.oidc.<identifier>.tokenAuth=
105 com.openexchange.cloudplugins.oidc.tokenAuth=
106
107 # This backends servlet path, which is appended to the default /oidc/ path.
108 #
109 # This property is optional.
110 # Default: <empty>
111 # com.openexchange.cloudplugins.oidc.<identifier>.backendPath=
112 com.openexchange.cloudplugins.oidc.backendPath=
113
114 # List of hosts where that this OIDCBackend is responsible for
115 # if all is present, this is responsible for all hosts
116 #
117 # It is possible to control the backendPath with this property.
118 # Another way would be to set the backendPath within the as-config.yml.
119 # If set in as-config.yml, it must be set as oidcPath
120 # Default: <empty>
121 # com.openexchange.cloudplugins.oidc.<identifier>.hosts=
122 com.openexchange.cloudplugins.oidc.hosts=
123
124 # Set the redirect location for a failed authentcation reuqest if the request could not be
        identified or took too long
125 com.openexchange.cloudplugins.oidc.failureRedirect=
126
127 # Set the redirect location for all other authentication exceptions that may occure. This
       mainly targets issues with the token validation
128 com.openexchange.cloudplugins.oidc.authenticationFailedExceptionRedirect=
129
130 # Set the redirect location for logout exceptions.
131 com.openexchange.cloudplugins.oidc.logoutFailedExceptionRedirect=
132
133 # Set the redirect location for general exceptions in the middleware that could not be
       handled by either
134 #  - failureRedirect
135 #  - authenticationFailedExceptionRedirect
136 #  - logoutFailedExceptionRedirect
137 # If only one endpoint should be defined for redirect, it is save to only set
       responseExceptionRedirect
138 com.openexchange.cloudplugins.oidc.responseExceptionRedirect=
```

File 21 /opt/open-xchange/etc/cloudplugins-remote-ldap.properties

```
1  # All properties below must be assigned to an identifier by configuring them as
2  # com.openexchange.cloudplugins.remote.ldap.key.<type>.<brand>
3  #
4  # Possible values for type:
5  # nginx
6  #   if the nginx version should be configured
7  # auth
8  #   if the AuthenticationService version should be configured
9  #
10 # Config layout:
11 #   <base>.<type>.<brand>=value
12
13 # Property to enable a service either for nginx or as an AuthenticationService
14 #
15 # Example:
16 # com.openexchange.cloudplugins.remote.ldap.enabled.nginx.brand1=true
17 # com.openexchange.cloudplugins.remote.ldap.enabled.auth.brand1=true
18 com.openexchange.cloudplugins.remote.ldap.enabled=false
```

```
19
20  # Configure the LDAP lookup method to find users using their logins.
21  # It is the method that is used by default when there is no brand specific
22  # configuration setting).
23  #
24  # Optional, defaults to "uid".
25  #
26  # Possible values:
27  #   * uid
28  #       attempts to find users by matching their login against the uid attribute.
29  #   * email
30  #        attempts to find users by matching their login against the alias attribute.
31  #   * auto
32  #       when the login contains a "@", the "email" method is used and when not,
33  #       the "uid" method is used
34  #   * uid-or-email
35  #       attempts to find users by matching their login against the uid and the alias
36  #       attributes (either may match)
37  #   * userid-contextid
38  #       splits the login by @ and assumes that the layout is userId@contextId
39  #
40  #
41  # Example:
42  # com.openexchange.cloudplugins.remote.ldap.uid.mode.nginx=auto
43  # com.openexchange.cloudplugins.remote.ldap.uid.mode.auth=auto
44
45  # One may define any number of such settings per brand by setting properties
46  # with the following format for their name:
47  #
48  # com.openexchange.cloudplugins.remote.ldap.uid.mode.nginx.<brand>=<uid|email|auto|uid-or-
        email|userid-contextid>
49  # com.openexchange.cloudplugins.remote.ldap.uid.mode.auth.<brand>=<uid|email|auto|uid-or-
        email|userid-contextid>
50  #
51  # For the list of possible values, please consult the documentation for
52  # com.openexchange.cloudplugins.remote.ldap.uid.mode
53  #
54  # Optional, defaults to falling back to the method configured in
55  # com.openexchange.cloudplugins.remote.ldap.uid.mode.nginx
56  # com.openexchange.cloudplugins.remote.ldap.uid.mode.auth
57  #
58  #
59  # Example:
60  # com.openexchange.cloudplugins.remote.ldap.uid.mode.nginx.brand1=auto
61  # com.openexchange.cloudplugins.remote.ldap.uid.mode.auth.brand2=uid
62
63  # The name of the external LDAP pool to be used to bind the user
64  # Mandatory, there is no default value.
65  #
66  # Example:
67  # com.openexchange.cloudplugins.remote.ldap.authLdapIdentifier.nginx.brand1=
        brand1_nginx_auth_pool
68  # com.openexchange.cloudplugins.remote.ldap.authLdapIdentifier.auth.brand2=
        brand2_normal_auth_pool
69  com.openexchange.cloudplugins.remote.ldap.authLdapIdentifier=
70
71  # The auth strategy to use against the remote LDAP for bind Requests
72  # Possible values:
73  #   * bind
74  #       do a single bind
75  #   * bind-and-revert
76  #       do and bind and revert to initial config
77  #
78  # Optional, will use bindAndRevert by default.
79  #
80  # WARNING: when 'bind' is selected, the auth pool must not be used for searching as the
        connection can
81  # be in a wrong state afterwards. 'bind' is faster than 'bindAndRevert' as only 1 call
        must be done
82  #
83  # Example:
84  # com.openexchange.cloudplugins.remote.ldap.authStrategy.brand1=bind-and-revert
```

```
com.openexchange.cloudplugins.remote.ldap.authStrategy=

# The name of the external LDAP pool to be used to search the user if searching is enabled via bindDnStrategy
# Optional, will use the authLdapIdentifier if not set
#
# Example:
# com.openexchange.cloudplugins.remote.ldap.searchLdapIdentifier.nginx.brand1=brand1_search_pool
com.openexchange.cloudplugins.remote.ldap.searchLdapIdentifier=

# Optionally change the uid lookup in our own LDAP
#
# Possible values:
#  * keep
#      do not change the uid
#  * add-default-domain:example.com
#      will add the default domain to an existing uid if it doesn't contain a @
#
# will use 'keep' if not set
#
# Example:
# com.openexchange.cloudplugins.remote.ldap.localUid.nginx.brand1=add-default-domain:example.com
com.openexchange.cloudplugins.remote.ldap.localUid=

# Optionally change the uid lookup in the external LDAP
#
# Possible values:
#   * keep
#       do not change the uid
#   * uid
#       use the uid of the UserEntity
#   * add-default-domain:example.com
#       will add the default domain to an existing uid if it doesn't contain a @
#   * alias | any-alias
#       uses any of the existing aliases from the userEntity object that isn't the catch-all alias
#   * mail | primary-mail
#       uses the primary mail of the user object which requires a database lookup via userId and contextId
#
# will use 'keep' if not set
#
# Example:
# com.openexchange.cloudplugins.remote.ldap.remoteUid.nginx.brand1=mail
com.openexchange.cloudplugins.remote.ldap.remoteUid=

# LDAP search filter for finding mailboxDN in remote LDAP
# Possible values:
#   * search:filter
#       search in the remote LDAP with the baseDN and the searchLdapIdentifier ldap pool
#   * parse:filter
#       parses existing data into a usable DN

#   Can use the following data:
#    * {{local}} if input contains an @
#    * {{domain}} if input contains an @
#    * {{uid}} as full input string
#    * {{{baseDN}}} the configured baseDN property, can be empty.
#        Note that the baseDN should be escaped with 3 { } as it would otherwise be escaped
#        This is also true for other values. Use 3 { } if those should not be escaped
#
# Mandatory, there is no default value.
#
# Example:
# com.openexchange.cloudplugins.remote.ldap.bindDnStrategy.nginx.brand1=search:uid={{uid}}
# com.openexchange.cloudplugins.remote.ldap.bindDnStrategy.nginx.brand1=parse:cn={{local}},ou={{domain}},{{{baseDN}}}
com.openexchange.cloudplugins.remote.ldap.bindDnStrategy=

```

```
150 # Remote LDAP baseDN used for the mailboxDN search
151 #
152 # Optional, the default value is empty string
153 #
154 # Example:
155 # com.openexchange.cloudplugins.remote.ldap.baseDN.nginx.brand1=o=domains,dc=email,dc=
        example,dc=net
156 com.openexchange.cloudplugins.remote.ldap.baseDN=
```

File 22 /opt/open-xchange/etc/cloudplugins-saml.properties

```
1  # The properties for cloud setup SAML bundle
2
3  ### General settings
4  ##########################################################################################
5  # Regex to validate host HTTP Header value
6  com.openexchange.cloudplugins.saml.host.regex=[0-9a-zA-Z.]*
7
8  # Regex to validate user name
9  com.openexchange.cloudplugins.saml.user.regex=[0-9a-zA-Z.@]*
10
11 # Base folder for the saml keystore files
12 # This property must be set if Java Security Manager is enabled
13 # It can't be set for inidividual identifiers
14 com.openexchange.cloudplugins.saml.keyStoreBasePath=
15
16 ### Key-store/certificate settings
17 ##########################################################################################
18 # All properties below can also be assigned to an identifier by configuring them as
19 # com.openexchange.cloudplugins.saml.<identifier>.key
20 #
21 # If a property is not set for an identifier, the key without the identifier is used as a
       default
22 # !!Note !!
23 #    If a property is marked as optional, the default value for the optional case is used
       and not the general
24 #    optional value if that is set
25 #
26 # Example:
27 #    com.openexchange.cloudplugins.saml.example.id=someValue
28 #    com.openexchange.cloudplugins.saml.keyStore=testStore
29 #    com.openexchange.cloudplugins.saml.example.keyStore=
30 #    example keyStore => testStore
31 #
32 # Example2:
33 #    com.openexchange.cloudplugins.saml.example.id=someValue
34 #    com.openexchange.cloudplugins.saml.brand=defaultBrand
35 #    com.openexchange.cloudplugins.saml.example.brand=
36 #    example brand (for ldap selection) => <empty>, will use the domainName of the request
       .
37 #
38 # !! Note !!
39 #    The list of possible SAMLBackends is identified by the property
40 #    com.openexchange.cloudplugins.saml.<identifier>.id
41 #    If there is no property for the id(s) set, no SAMLBackend will be started.
42 #    com.openexchange.cloudplugins.saml.id is also valid and will be used as an empty
       identifier
43 #    An empty identifier will use all properties set in this config file.
44 #
45 # The full path to a Java keystore containing the IdPs certificate.
46 #
47 # Default: <empty>
48 com.openexchange.cloudplugins.saml.keyStore=
49
50 # Password to open the keystore.
51 #
52 # Default: <empty>
53 com.openexchange.cloudplugins.saml.keyStorePass=
```

```
54
55  # The aliases of the IdP certificate entry within the above specified
56  # keystore. Split by ','. Multiple certs can be used in a rolling upgrade
57  # case at the customers location without the need to coordinate an upgrade.
58  #
59  # The support for multiple certs is only enabled on 7.10.1+ systems.
60  # A 7.10.0 system will only use the first certAlias
61  #
62  # Default: <empty>
63  com.openexchange.cloudplugins.saml.certAlias=
64
65  # The alias of the signingKey entry within the above specified
66  # keystore.
67  #
68  # Default: <empty>
69  com.openexchange.cloudplugins.saml.signingKeyAlias=
70
71  # The password of the signingKey entry within the above specified
72  # keystore.
73  #
74  # Default: <empty>
75  com.openexchange.cloudplugins.saml.signingKeyPassword=
76
77  # The alias of the decryptionKey entry within the above specified
78  # keystore.
79  #
80  # Default: <empty>
81  com.openexchange.cloudplugins.saml.decryptionKeyAlias=
82
83  # The password of the decryptionKey entry within the above specified
84  # keystore.
85  #
86  # Default: <empty>
87  com.openexchange.cloudplugins.saml.decryptionKeyPassword=
88
89  ### SAML Specific configuration
90  ##########################################################################################
91  # The id inside the saml response which holds the userinformation
92  #
93  # The search for the id is done in the attributeStatement if nothing is configured
94  # possible configuration values are:
95  #    key            // search is done in the attributeStatement with the key
96  #    attribute:key  // search is done in the attributeStatement with the key
97  #    subject:nameID // the subject:NameId is used
98  #
99  # In any case, the id must match the uid used to provision the user
100 com.openexchange.cloudplugins.saml.id=
101
102 # The ldap lookup used by this SAMLBackend
103 #
104 # Configure the LDAP lookup method to find users using their identifiers.
105 #
106 #
107 # Possible values:
108 # uid
109 #   attempts to find users by matching their login against the uid attribute.
110 # email
111 #   attempts to find users by matching their login against the alias attribute.
112 # auto
113 #   when the login contains a "@", the "email" method is used and when not,
114 #   the "uid" method is used
115 # uid-or-email
116 #   attempts to find users by matching their login against the uid and the alias
117 #   attributes (either may match)
118 #
119 # This property is optional.
120 # Default: uid
121 com.openexchange.cloudplugins.saml.ldapLookup=
122
123 # URL of where the users are redirected after logout if single_logout is active
124 # Must only be set if enableSingleLogout is enabled
125 #
```

```
126 # Default: <empty>
127 com.openexchange.cloudplugins.saml.logout.redirect.url=
128
129 # The brand to use for OXaaS LDAP authentication operations.
130 # Uses the host name when empty or not set.
131 #
132 # This property is optional.
133 # Default: <empty>
134 com.openexchange.cloudplugins.saml.brand=
135
136 # The URL to redirect to in case the SAML back-end fails to look up the authenticated user
        .
137 # When left empty or not set, an HTTP 500 error page is sent instead.
138 #
139 # This property is optional.
140 # Default: <empty>
141 com.openexchange.cloudplugins.saml.failure.redirect=
142
143 # The URL to redirect to in case the SAML back-end has an error, when the user logs out
144 # When left empty or not set, the value of com.openexchange.cloudplugins.saml.failure.
        redirect is used.
145 #
146 # Default: <empty>
147 com.openexchange.cloudplugins.saml.logout.failure.redirect=
148
149 # The samlPath value required for the servlet alias
150 # registered with '{prefix}/saml/{samlPath}/..' as servlet alias.
151 #
152 # This property is optional.
153 # Default: <empty>
154 com.openexchange.cloudplugins.saml.samlPath=
155
156 # Static redirect upon login or relogin
157 #
158 # Ths property is optional.
159 # Default: <empty>
160 com.openexchange.cloudplugins.saml.staticRedirect=
161
162 # Whether the SPs metadata XML shall be made available via HTTP. The according
163 # servlet will then be available under 'http(s)://{hostname}/{prefix}/saml/metadata'.
164 #
165 # Default: false
166 com.openexchange.cloudplugins.saml.enableMetadataService=false
167
168 # Whether the single logout profile is enabled.
169 #
170 # Default: false
171 com.openexchange.cloudplugins.saml.enableSingleLogout=false
172
173 # Sets the entity ID of the service provider.
174 #
175 # This property is mandatory.
176 # Default: <empty>
177 com.openexchange.cloudplugins.saml.entityID=
178
179 # Sets the human-readable name of the service provider.
180 #
181 # This property is mandatory.
182 # Default: <empty>
183 com.openexchange.cloudplugins.saml.providerName=
184
185 # Sets the URL of the local assertion consumer service (ACS). This value is used within
186 # authentication requests, compared against Destination attributes in IdP responses
187 # and will be contained in the service providers metadata XML. The according endpoint
188 # is always registered with '{prefix}/saml/{samlPath}/acs' as servlet alias.
189 #
190 # This property is mandatory.
191 # Default: <empty>
192 # Example: https://appsuite.example.com/appsuite/api/saml/{samlPath}/acs
193 com.openexchange.cloudplugins.saml.acsURL=
194
195 # Sets the URL of the local single logout service. This value is compared against
```

```
    Destination
# attributes in IdP responses and will be contained in the service providers metadata XML.
# The according endpoint is always registered with '{prefix}/saml/{samlPath}/sls' as servlet alias.
#
# This property is mandatory if 'com.openexchange.cloudplugins.saml.enableSingleLogout' is 'true'.
# Default: <empty>
# Example: https://appsuite.example.com/appsuite/api/saml/{samlPath}/sls
com.openexchange.cloudplugins.saml.slsURL=

# The binding via which logout responses shall be sent to the IdP on IdP-initiated single
# logout flows. Must be 'http-redirect' or 'http-post'.
#
# This property is mandatory if 'com.openexchange.cloudplugins.saml.enableSingleLogout' is 'true'.
# Default: http-redirect
com.openexchange.cloudplugins.saml.logoutResponseBinding=http-redirect

# The HTML template to use when logout responses are sent to the IdP via HTTP POST.
# The template must be located in '/opt/open-xchange/templates'.
#
# This property is mandatory if 'com.openexchange.cloudplugins.saml.enableSingleLogout' is 'true'
# and 'com.openexchange.cloudplugins.saml.logoutResponseBinding' is set to 'http-post'.
# Default: saml.logout.response.html.tmpl
com.openexchange.cloudplugins.saml.logoutResponseTemplate=saml.logout.response.html.tmpl

# The entity ID of the IdP. It will be used to validate the 'Issuer' elements of SAML responses.
#
# This property is mandatory.
# Default: <empty>
com.openexchange.cloudplugins.saml.idpEntityID=

# The URL of the IdP endpoint where authentication requests are to be sent to.
#
# This property is mandatory.
# Default: <empty>
com.openexchange.cloudplugins.saml.idpAuthnURL=

# The URL of the IdP endpoint where logout requests are to be sent to.
#
# This property is mandatory if 'com.openexchange.cloudplugins.saml.enableSingleLogout' is 'true'.
# Default: <empty>
com.openexchange.cloudplugins.saml.idpLogoutURL=

# It is possible to enable a special kind of auto login mechanism that allows user agents to
# re-use an existing OX session if it was created during the same browser session. If enabled,
# a special cookie will be set, which is linked to the OX session and bound to the browser sessions
# life time. The advantage of this mechanism is, that sessions are simply re-entered if the user
# refreshes his browser window. He is then also able to open more than one tab of OX App Suite
# at the same time. This mechanism can only re-use sticky sessions, i.e. it is mandatory that the
# requests are always routed to the same backend for a certain session.
#
# --- SECURITY WARNING ---
# Enabling this setting is not compliant to the SAML specification as it bypasses the IdP in
# certain cases. Additionally in scenarios where a public device is used, a foreign user might
# take over a formerly authenticated users session if that user forgets to log out and doesn't
# close his web browser (even if he closes the App Suite tab). As no login screen is displayed
# by OX in SAML environments, the user is even not able to decide, whether the application
```

```
     shall
# remember him or not.
#
# Default: false
com.openexchange.cloudplugins.saml.enableAutoLogin=false

# Whether unsolicited responses will be accepted or not.
#
# Default: true
com.openexchange.cloudplugins.saml.allowUnsolicitedResponses=true

# Whether SAML-specific auto-login is enabled, that uses the SessionIndex of the
    AuthnResponse
#
# Default: false
com.openexchange.cloudplugins.saml.enableSessionIndexAutoLogin=false

# List of hosts where that this SAMLBackend is responsible for
# if all is present, this SAMLBackend responsible for all hosts
#
# It is possible to control the samlPath with this property.
# Another way would be to set the samlPath within the as-config.yml.
# Default: <empty>
com.openexchange.cloudplugins.saml.hosts=
```

File 23 /opt/open-xchange/etc/trustedidentity-ldap.properties

```
### Configuration for LDAP support for Trusted Identity.

# Storage Keys are used to decrypt private ECDSA keys that are stored in LDAP.
# LDAP oxCloudTrustedIdentityKeyPair entities contain optional (but
# strongly encouraged) references to storage keys by name.
# Storage keys are symmetric/secret keys (AES is recommended).
# Those storage keys are defined here by configuration, with multiple parameters,
# that are defined using different prefixes but the same storageKeyName part in
# each property name:
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.{storageKeyName}=...
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.keyType.{storageKeyName
    }=...
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.cipher.{storageKeyName
    }=...
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.alias.{storageKeyName}=...
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.secret.{storageKeyName
    }=...
#
# For example, using "sk1" as the {storageKeyName}:
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.sk1=keystore:/opt/
    open-xchange/etc/sk1.jks
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.keyType.sk1=AES
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.cipher.sk1=AES/GCM/
    NoPadding
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.alias.sk1=storageKey
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.secret.sk1=secret

# The storage key file location is a fully qualified path to either a plain
# file that contains the encoded bytes of the symmetric key, or a Java KeyStore
# file.
# When using a keystore file, one may also want to configure the key alias and secret
# (see next properties below.)
#
# Property format:
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.{storageKeyName}=...
# When the file is a KeyStore file, it must be prepended with "keystore:":
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.{storageKeyName}=
    keystore:...
#
# Example of a plain file:
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.sk1=/opt/open-xchange
```

```
    /etc/sk1.key
#
# Example of a KeyStore file:
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.sk1=keystore:/opt/
    open-xchange/etc/sk1.jks
#
# The value is mandatory and has no default.
#
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.storageKeyName=

# KeyStore key alias: when using a KeyStore file, defines the alias of the key entry to
    use
# as the symmetric/secret key.
#
# Example:
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.alias.sk1=storageKey
#
# The value is optional when the KeyStore file contains a single entry.
#
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.alias.storageKeyName=

# KeyStore secret: when using a KeyStore file, defines the password to use to
# decrypt the KeyStore as well as the key inside of it.
#
# Example:
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.secret.sk1=my_secret
#
# The value is optional and defaults to an empty string ("").
#
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.secret.storageKeyName=

# The cipher algorithm defines which symmetric decryption algorithm to use when
# unwrapping the private key from LDAP, and must be the same as the cipher used
# when encrypting it in the first place.
#
# Example:
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.cipher.sk1=AES/CBC/
    PKCS5Padding
#
# The value is optional and defaults to AES/GCM/NoPadding
#
# com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.cipher.storageKeyName=

# Storage Key caching: keys that are looked up in LDAP are cached in memory for
    performance
# reasons.
# The following configuration property determines how long they are kept in cache
# before being fetched from LDAP again:
# Format: duration[d|h|m|s|ms]
#
# Example:
# com.openexchange.cloudplugins.trustedidentity.ldap.cache.ttl=4h
#
# Optional, the default value is 1h (one hour).
com.openexchange.cloudplugins.trustedidentity.ldap.cache.ttl=
```

File 24 /opt/open-xchange/etc/oxaas-alias.properties

```
# Setting to control the uri of the tarent adapter
com.openexchange.oxaas.alias.tarent.uri=http://localhost

# Setting to control allowed domains
com.openexchange.oxaas.alias.allowed.domains=

# Setting to control if alias adapter should be in test mode (this means a mock is used)
com.openexchange.oxaas.alias.test=false

# Loglevel for the internal OkHttp3 client
```

```
# Allowed values are: NONE, BASIC, HEADERS, BODY
com.openexchange.oxaas.alias.tarent.loglevel=NONE

# Setting to control if Unsecured Https should be allowed or not
# Default: false
com.openexchange.oxaas.alias.tarent.allowUnsecuredHttps=false

# Master user needed to delete alias
com.openexchange.oxaas.alias.master.user=

# Master user password needed to delete alias
com.openexchange.oxaas.alias.master.password=

# Setting to control if a client cert shoul be loaded, must be in PKCS 12 format
# Can be mixed with com.openexchange.oxaas.alias.tarent.allowUnsecuredHttps
# allowUnsecuredHttps=true and clientcert.path=set
#     a client cert is used but the hostname is not verified and all server certs are trusted
# allowUnsecuredHttps=false and clientcert.path=set
#     a client cert is used, but hostname is verified and server certs must be trustable
# Default: empty
com.openexchange.oxaas.alias.tarent.ssl.clientcert.path=

# Setting that holds the password for the PKCS 12 container
# Default: empty
com.openexchange.oxaas.alias.tarent.ssl.clientcert.password=

# Default number of aliases to be configured by each userr
# Default: 15
com.openexchange.oxaas.aliasquota=15
```

File 25 /opt/open-xchange/etc/oxaas-mail-notification-templates.properties

```
# Config cascade-aware property to control the prefix of the users templates
# Used to show the overquota warning mails
#
# Format: prefix[.language|].quota.[html|text|subject].tmpl
com.openexchange.oxaas.mail.quota.notify.prefix=notify.oxaas.over.quota

# Config cascade-aware property to control the prefix of the users templates
# Used to show the welcome mail
#
# Format: prefix[.language|].[html|text|subject].tmpl
com.openexchange.oxaas.mail.welcome.mail.notify.prefix=notify.oxaas.welcome.mail

# Config cascade-aware property to control the prefix of the users templates
# Used to show disabled sent spam mail
#
# Format: prefix[.language|].[html|text|subject].tmpl
com.openexchange.oxaas.mail.removed.sent.spam.notify.prefix=notify.oxaas.disable.sent.spam
```

File 26 /opt/open-xchange/etc/oxaas-drive-quota-notification.properties

```
# Config-cascade aware setting to control the quotas that should be monitored
com.openexchange.oxaas.mail.quota.drive.quotas=90,100

# Setting to control if the quota notification should be ignored for context-wide filestores
# This makes sense, if the users are having user quota and there is a context-quota in place too
# Config-Cascade only on context level
com.openexchange.oxaas.mail.quota.drive.ignoreContextQuota=false

```

```
 9  # Config-cascade aware setting to control if the admin should also receive a mail, in case
       the filestore is context-wide
10  com.openexchange.oxaas.mail.quota.drive.updateAdmin=false
11
12  # Config-cascade aware setting to control how often a mail should be sent
13  # Default is 86400 (1 day)
14  # Set to 0 to ignore that and always send a new mail
15  com.openexchange.oxaas.mail.quota.drive.mail.seconds=86400
```

File 27 /opt/open-xchange/etc/oxaas-mail-unread.properties

```
 1  # Value holding the usernames for basic authentication
 2  # must be the username for basic auth split by ,
 3  # e.g hosterone,hostertwo
 4  com.openexchange.oxaas.mail.unread.ws.basic.usernames=
 5
 6  # Setting to control basic auth username
 7  # example would be com.openexchange.oxaas.mail.unread.ws.basic.hosterone.brand=
       internalBrandForhosterone
 8  #com.openexchange.oxaas.mail.unread.ws.basic.[username].brand=
 9
10  # Setting to control basic auth password
11  # example would be com.openexchange.oxaas.mail.unread.ws.basic.hosterone.password=
       verySecretPassword
12  #com.openexchange.oxaas.mail.unread.ws.basic.[username].password=
```

File 28 /opt/open-xchange/etc/oxaas-mail.properties

```
 1  # Value holding the usernames for basic authentication
 2  # must be the username for basic auth split by ,
 3  # e.g hosterone,hostertwo
 4  com.openexchange.oxaas.mail.ws.basic.usernames=
 5
 6  # Setting to optimize the fetching of recentMessages
 7  # If set to true, the virtual/all folder will be quried
 8  # If set to false, the calculation is done in the middleware
 9  # config-cascade aware
10  com.openexchange.oxaas.mail.ws.recentMessagesFromVirtualAll=false
11
12  # Setting to control basic auth username
13  # example would be com.openexchange.oxaas.mail.basic.hosterone.password=verySecretPassword
14  #com.openexchange.oxaas.mail.ws.basic.[username].password=
15
16  # Setting to control basic auth password
17  # example would be com.openexchange.oxaas.mail.basic.hosterone.brand=
       internalBrandForhosterone
18  #com.openexchange.oxaas.mail.ws.basic.[username].brand=
```