OX Cloud Plugins Technical Documentation for
1.11.9

2022-02-04

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 1.11.9 Feature Delivery for Core 7.10.5 and 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/1.11.9/DebianBuster
https://software.open-xchange.com/components/cloud-plugins/stable/1.11.9/DebianStretch
https://software.open-xchange.com/components/cloud-plugins/stable/1.11.9/RHEL7

1.4 Build Dependencies

This delivery was build and tested with following dependencies:

```
AppSuite:node-10,plugins-1.7.0-rev2,frontend-7.10.6-rev5,
backend-7.10.6-rev7
```

1.5 Notice

ℹ️ **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:

```
1   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

```
1   {
2       "data": [
3           {
4               "announcement": "Ad astra per aspera",
5               "summary": "summary or title",
6               "discardable": true,
7               "id": 12345,
8               "version": 1
9           },
10          {
11              "announcement": "Carpe diem",
12              "summary": "summary or title",
13              "discardable": false,
14              "id": 12346,
15              "version": 2
16          }
17      ]
18  }
```

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

```
1  {
2      "id": 789,
3      "version": 2
4  }
```

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 is reloadable.

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:

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

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
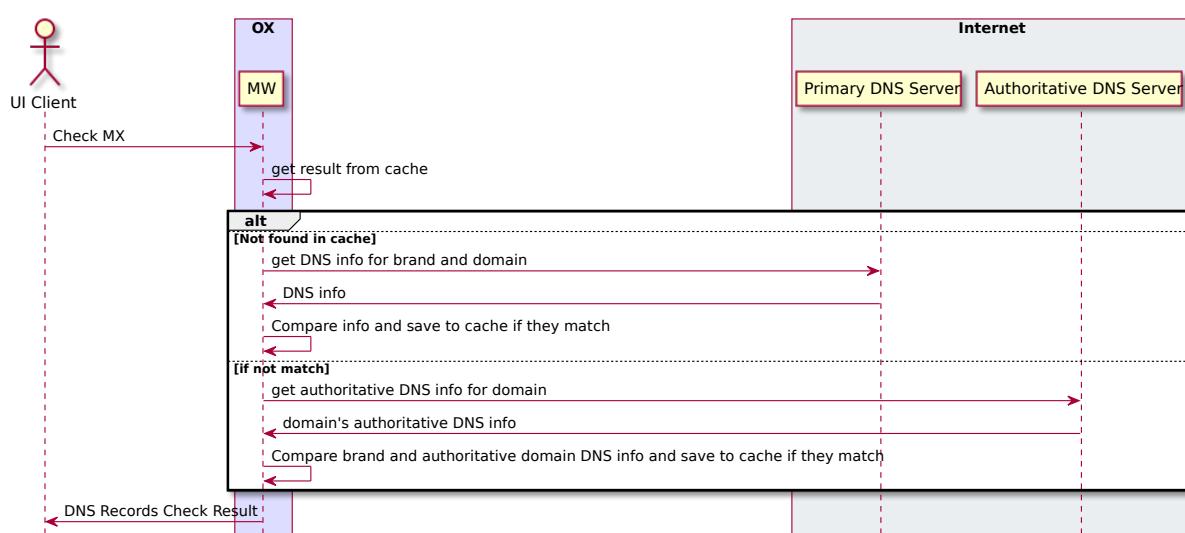
Figure 1: PlantUML 1

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 Packages and Version

4.1 Package open-xchange-cloudplugins

The Open-Xchange cloud plugin bundles
Version: 1.11.9-1

Type: OX Middleware Plugin
Depends on:

```
open-xchange-admin-reseller (<<7.10.7)
open-xchange-admin-soap-reseller (>=7.10.5)
open-xchange-cassandra (<<7.10.7)
open-xchange-cassandra (>=7.10.5)
open-xchange-mailfilter (<<7.10.7)
open-xchange-mailfilter (>=7.10.5)
open-xchange-rest (<<7.10.7)
open-xchange-rest (>=7.10.5)
```

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.

ℹ️ 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) );
```

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

```
1   Usage: createtkg112report
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    -b,--single-brand <single-brand>             * Brand to create report for. Identified
        by the brand admins sid.
9    -t,--terminate-report                         Terminates the currently processed
        report uuid.
10   -p,--pending-reports                          Get the status of the pending report.
11
12  Entries marked with an asterisk (*) are mandatory.
13  Entries marked with an question mark (?) are mandatory depending on your
14  configuration.
15  Entries marked with a pipe (|) are mandatory for one another which means that
16  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.

```
1   {
2       "uuid":"28b3573af6734877a448ab614698d115",
3       "reportType":"OXaaS-report",
4       "timestamps":{
5           "start":1498477236283,
6           "stop":1498477276490
7       },
8       "version":{
9           "buildDate":"01.01.2017",
10          "version":"7.8.3"
11      },
12      "configs":{
13          "options":{
14              "show-drive-metrics":false,
15              "timeframe-start":1466941236283,
16              "show-mail-metrics":false,
17              "single-brand":0,
18              "ignore-admin":true,
19              "timeframe-end":1498477236283
20          }
21      },
22      "errors":{
23          "Exception-ID":"ERROR-ID Categories=ERROR Message='Message' exceptionID=Exception-ID
                "
24      },
25      "oxaas":{
26          "capabilitySets":{
27              "283724704":"autologin,blacklist,...",...
28          },
29          "brandname":{
30              "totals":{
31                  "quota" : 2621440000,
32      "quotaUsage" : 2522466,
33      "mailQuota" : 45365,
```

```
34        "mailQuotaUsage" : 0
35             },
36             "1":{
37                 "3":{
38                     "capabilitySet":"283724704",
39                     "drive" : {
40         "mime-types" : {
41          "text/plain" : 5,
42           "application/zip" : 2
43         },
44         "file-count-all-versions" : 7,
45         "quota" : 104857600,
46         "used-quota" : 2511077,
47         "file-size-min" : 5,
48         "file-count-latest-version" : 7,
49         "file-size-avg" : 358725,
50         "file-size-max" : 1255526
51      },
52      "mail" : {
53         "mail-quota" : 2048,
54         "mail-quota-usage" : 0
55      },
56                 "imap-login":"3@1",
57                 "login-info":"adam@brandname",
58                 "email":"adam@brandname",
59                 "user-logins":{
60      "HTTP" : 1497364334092,
61         "open-xchange-appsuite" : 1497364334090
62                 },
63                 "unified-quota-enabled":true,
64                 "unified-quota-limit":104859648,
65                 "unified-quota":2511533
66         }
67        }
68       }
69     }
70 }
```

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.

```
1 Email_nameID, Email_Displayname, Email_Begin, Email_Address, Email_Create, Email_Remove,
     Email_isactive
2 ID1, TestUser1, 2016-12-06 14:11:00 +0000, testuser@brand.com, 2016-12-06 14:11:00 +0000,
     , true
3 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 26)
/opt/open-xchange/etc/cloudplugins-authentication.properties (page 27)
/opt/open-xchange/etc/cloudplugins-master-auth-servlet.properties (page 28)
/opt/open-xchange/etc/cloudplugins-cassandra.properties (page 28)
/opt/open-xchange/etc/cloudplugins.properties (page 35)

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

Implementation of VadeSecure antiphishing for cloudplugins within LDAP
Version: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
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 39)

4.3   Package open-xchange-cloudplugins-blackwhitelist-ldap

Implementation of blacklist whitelist for cloudplugins within LDAP
Version: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
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 39)

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: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
```

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 39)

4.5 Package open-xchange-cloudplugins-keycloak

Keycloak connector This package contains a keycloak connection handler to retrieve access and refresh tokens.
Version: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-oidc (<<7.10.7)
open-xchange-oidc (>=7.10.5)
```

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 end-points

    - 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 41)

4.6 Package open-xchange-cloudplugins-loginproxy-ws

Cloudplugins loginproxy REST API This package provides a restful API for the 2-step login.
Version: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
```

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-forward.yaml (page 42)
/opt/open-xchange/etc/cloudplugins-loginproxy-ws.properties (page 43)

4.7 Package open-xchange-cloudplugins-mailfilter

CloudPlugins MailFilter Utilities This package implements a mailfilter interceptor driver framework
and provides some useful drivers.
Version: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-mailfilter (<<7.10.7)
open-xchange-mailfilter (>=7.10.5)
open-xchange-rest (<<7.10.7)
open-xchange-rest (>=7.10.5)
```

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 Cloud-ManagementCassandraService the autoforward status and how many redirects exist.

    - RedirectBlacklistDriver - supports Config Cascade. Blocks users from creating only auto-forward 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 44)

4.8 Package open-xchange-cloudplugins-master-auth

Provides a CloudPlugins CloudAuthenticationDriver for master authentication
Version: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
```

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 44)

4.9   Package open-xchange-cloudplugins-mx-checker

CloudPlugins MX Checker Connector This package implements an MX Checker Connector for OX Cloud
Version: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
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 44)

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: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-oidc (<<7.10.7)
open-xchange-oidc (>=7.10.5)
```

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 46)

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

Provides remote LDAP CloudPlugins CloudAuthenticationDriver and NginxAuthDriverExtended|DeclarativeNginxAuthDriver
Version: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
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 49)

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: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-saml-core (<<7.10.7)
open-xchange-saml-core (>=7.10.5)
```

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 53)

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: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
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 54)

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: 1.11.9-1
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: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-admin (<<7.10.7)
open-xchange-admin (>=7.10.5)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
```

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 55)

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

OXaaS notification mail servlet bundle
Version: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
open-xchange-imap (<<7.10.7)
open-xchange-imap (>=7.10.5)
open-xchange-smtp (<<7.10.7)
open-xchange-smtp (>=7.10.5)
```

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 55)
/opt/open-xchange/etc/oxaas-drive-quota-notification.properties (page 56)

4.16.5 Templates

/opt/open-xchange/templates/notify.oxaas.over.quota.90.subject.tmpl
/opt/open-xchange/templates/notify.oxaas.over.quota.100.text.tmpl
/opt/open-xchange/templates/notify.oxaas.over.quota.90.text.tmpl
/opt/open-xchange/templates/notify.oxaas.over.quota.90.html.tmpl
/opt/open-xchange/templates/notify.oxaas.over.quota.100.subject.tmpl
/opt/open-xchange/templates/notify.oxaas.over.quota.100.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: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
open-xchange-imap (<<7.10.7)
open-xchange-imap (>=7.10.5)
```

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 56)

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: 1.11.9-1
Type: OX Middleware Plugin
Depends on:

```
open-xchange-cloudplugins (<<1.11.10)
open-xchange-cloudplugins (>=1.11.9)
open-xchange-core (<<7.10.7)
open-xchange-core (>=7.10.5)
open-xchange-imap (<<7.10.7)
open-xchange-imap (>=7.10.5)
```

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 56)

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

```
1   # Regex to validate host HTTP Header value
2   com.openexchange.authentication.cloudplugins.host.regex= [0-9a-zA-Z.]*
3
4   #Regex to validate user name
5   com.openexchange.authentication.cloudplugins.user.regex= [0-9a-zA-Z.@]*
6
7   # Setting to control the start of the own authenticationService which would be used as a
        fallback in the Tracker if started
8   # Default: true
9   com.openexchange.authentication.cloudplugins.enableauthentication=true
10
11  # Comma seperated blacklist of hostnames that should not be handled by the default
        authenticationService
12  # Default: <empty>
13  com.openexchange.authentication.cloudplugins.authentication.blacklist=
14
15  # Setting to control the start of the CloudAuthenticationDriverTracker
16  # Default: true
17  com.openexchange.authentication.cloudplugins.enable.authentication.tracker=true
18
19  # Configure the LDAP lookup method to find users using their logins.
20  # It is the method that is used by default when there is no brand specifi
21  # configuration setting).
22  #
23  # Optional, defaults to "uid".
24  #
25  # Possible values:
26  # uid
27  #   attempts to find users by matching their login against the uid attribute.
28  # email
29  #   attempts to find users by matching their login against the alias attribute.
30  # auto
31  #   when the login contains a "@", the "email" method is used and when not,
32  #   the "uid" method is used
33  # uid-or-email
34  #   attempts to find users by matching their login against the uid and the alias
35  #   attributes (either may match)
36  #
37  # Note that this only applies to the default authentication driver. If there
38  # is a custom implementation that is used for a given brand, its behavior is
39  # not influenced by this properties.
40  #
41  # Example:
42  # com.openexchange.authentication.cloudplugins.authentication.uid.mode=auto
43
44  # One may define any number of such settings per brand by setting properties
45  # with the following format for their name:
46  #
47  # com.openexchange.authentication.cloudplugins.authentication.uid.mode.<brand>=<uid|email|
        auto|uid-or-email>
48  #
49  # For the list of possible values, please consult the documentation for
50  # com.openexchange.authentication.cloudplugins.authentication.uid.mode
51  #
52  # Optional, defaults to falling back to the method configured in
53  # com.openexchange.authentication.cloudplugins.authentication.uid.mode
54  #
55  # Note that this only applies to the default authentication service driver.
56  # If there is a custom implementation that is used for a given brand, its
57  # behavior is not influenced by these properties.
58  #
59  # Example:
60  #
61  # com.openexchange.authentication.cloudplugins.authentication.uid.mode.brand1=auto
62  # com.openexchange.authentication.cloudplugins.authentication.uid.mode.brand2=uid
```

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

```
1    #
2    # login name of httpauth user to access the master auth api
3    #
4    com.openexchange.cloudplugins.master.auth.httpauth.login=
5
6    #
7    # password of httpauth user
8    #
9    com.openexchange.cloudplugins.master.auth.httpauth.password=
```

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

```
1    # turn on/off cassandra integration
2    # possible values: true/false
3    com.openexchange.cloudplugins.useCassandra = false
4
5    # keyspace to use in cassandra
6    com.openexchange.cloudplugins.cassandraKeyspace = ox
7
8    # comma separated list of brands which logins should be recorded to cassandra
9    com.openexchange.cloudplugins.cassandra.loginrecorder.remoteipfor.brands=
10
11   # comma separated list of brands where alias creation and deletion time should be recorded
            to cassandra
12   com.openexchange.cloudplugins.cassandra.loginrecorder.createdeletealias.brands=
13
14   # comma separated list of brands where mail autoforward active flag should be set
15   com.openexchange.cloudplugins.cassandra.autoforward.record.brands=
16
17   # Default TTL (in seconds) of change history entries in the permission_change_history
            table.
18   #
19   # Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
20   # Default value: 90 days
21   com.openexchange.cloudplugins.cassandra.changehistory.default.ttl=90d
22
23   # Per-brand TTL (in seconds) of change history entries in the permission_change_history
            table.
24   #
25   # Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
26   # Example:
27   # com.openexchange.cloudplugins.cassandra.changehistory.brand.ttl.brand1=30d
28   # com.openexchange.cloudplugins.cassandra.changehistory.brand.ttl.brand2=24h
29   # com.openexchange.cloudplugins.cassandra.changehistory.brand.ttl.brand3=0
30
31   # String of text to use as the "reason" when permission changes are performed using the
32   # SOAP or RMI API.
33   # Defaults to "SOAP API" when empty:
34   com.openexchange.cloudplugins.cassandra.changehistory.soap.reason=
35   # Enables the cassandra login recorder
36   com.openexchange.cloudplugins.cassandra.useLoginRecorder = true
```

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

```
1
2    # ldap url for read access
3    com.openexchange.cloudplugins.read.ldapurl=ldap://localhost:389
4
5    # admin dn for read access
6    com.openexchange.cloudplugins.read.binddn=cn=oxadmin,o=oxcs
7
8    # admin dn password for read access
```

```
 9   com.openexchange.cloudplugins.read.bindpassword=
10
11   # ldap url for write access
12   com.openexchange.cloudplugins.write.ldapurl=ldap://localhost:389
13
14   # admin dn for write
15   com.openexchange.cloudplugins.write.binddn=cn=oxadmin,o=oxcs
16
17   # admin dn password for write access
18   com.openexchange.cloudplugins.write.bindpassword=
19
20   # tree for brands
21   com.openexchange.cloudplugins.branddn=ou=brands,o=oxcs
22
23   # tree for other stuff like mailstores
24   com.openexchange.cloudplugins.configdn=ou=config,o=oxcs
25
26   # tree for contexts
27   com.openexchange.cloudplugins.contextdn=ou=contexts,o=oxcs
28
29   # tree for class of service definitions
30   com.openexchange.cloudplugins.classofservicedn=ou=services,o=oxcs
31
32   # tree for trusted identity keys
33   com.openexchange.cloudplugins.trustedidentitykeydn=ou=keys,o=oxcs
34
35   # password for doveadm service
36   com.openexchange.cloudplugins.doveadmPassword=
37
38   # This timeout only works since Java 6 SE to time out waiting for a response.
39   com.openexchange.cloudplugins.read.timeout=10000
40
41   # Specifies the connect timeout (in milliseconds) when establishing a connection to the
         LDAP end-point
42   #
43   # Default is 5000 (5 seconds)
44   com.openexchange.cloudplugins.connect.timeout=5000
45
46   # Setting if user password hashs should be updated if the algorithm doesn't match the
         configured hash
47   com.openexchange.cloudplugins.password.updateUserPassword=false
48
49   # Algorithm to use to encrypt user passwords
50   # valid values are:
51   #    MD5
52   #    SMD5
53   #    SHA
54   #    SHA256
55   #    SHA384
56   #    SHA512
57   #    SSHA
58   #    SSHA256
59   #    SSHA-384 (old name for SSHA384)
60   #    SSHA384
61   #    SSHA512
62   #    CRYPT (general CRYPT identifier, uses CRYPT-SHA512 for password generation)
63   #    CRYPT-BCRYPT
64   #    CRYPT-MD5
65   #    CRYPT-SHA256
66   #    CRYPT-SHA512
67   # Default: CRYPT-SHA512
68   com.openexchange.cloudplugins.password.algorithm=SSHA
69
70   # Rounds for supported algorithms
71   # As of now only CRYPT-BCRYPT, CRYPT-SHA256, CRYPT-SHA512 support rounds
72   # Bcrypt is on a log scale while SHA256 and SHA512 are just rounds.
73   # Unset will use the algorithms default
74   # Bcrypt: 10
75   # SHA256: 5000
76   # SHA512: 5000
77   com.openexchange.cloudplugins.password.algRounds=
78
```

```
 79   # Whether to enable timer metrics for password verifications,
 80   # defaults to false
 81   #
 82   # When enabled, the service will record timer metrics for the
 83   # duration of:
 84   # - successful logins:
 85   #   cloud-plugins-ldap/successfulLogins/<algorithm|rounds>
 86   # - failed logins:
 87   #   cloud-plugins-ldap/failedLogins/<algorithm|rounds>
 88   # - user password hash updates:
 89   #   cloud-plugins-ldap/passwordHashUpdates/<algorithm|rounds>
 90   com.openexchange.cloudplugins.password.metrics=false
 91
 92   # how long should the random salts be
 93   # only relevant for SMD5, SSHA, SSHA256, SSHA384, SSHA512
 94   com.openexchange.cloudplugins.password.algorithm.saltlength=64
 95
 96   # HTTP Header from which the brand name is fetched
 97   com.openexchange.cloudplugins.header=host
 98
 99   # can a user change the password without providing the old one?
100   # default is false
101   com.openexchange.capability.password_change_without_old_password=false
102
103   # Comma separated list of brands that where aliases should be moved into deleted tree
104   com.openexchange.cloudplugins.store.deleted.alias.brands=
105
106   # jndi ldap pool configuration
107   # see http://docs.oracle.com/javase/jndi/tutorial/ldap/connect/config.html for more
        information
108   com.openexchange.cloudplugins.pool=true
109   com.openexchange.cloudplugins.pool.initsize=1
110   com.openexchange.cloudplugins.pool.maxsize=20
111   com.openexchange.cloudplugins.pool.prefsize=10
112   com.openexchange.cloudplugins.pool.timeout=300000
113   com.openexchange.cloudplugins.pool.protocol=plain
114
115   # Allowed encryption methods for LDAP userPassword
116   # internal default: MD5,SHA,CRYPT,SSHA,SSHA384
117   com.openexchange.cloudplugins.setPasswordHash.allowedMethods=MD5,SHA,CRYPT,SSHA,SSHA384
118
119   # Comma separated list of brands that should be able to explicitly add maildomains to
        contexts
120   # This parameter is reloadable
121   com.openexchange.cloudplugins.explicitMailDomains.brands=
122
123   # Enable or disable the requirement to prefix context names with brandname_
124   # Default is false
125   # This parameter is reloadable
126   com.openexchange.cloudplugins.omit.contextname.prefix=false
127
128   # Enable Class of Service support.
129   #
130   # Note that this property is not reloadable, and changing it requires a restart.
131   #
132   # Defaults to: false
133   com.openexchange.cloudplugins.cos.enable=false
134
135   # Enable limiting the logging of invalid class of service properties
136   # that cannot be parsed.
137   #
138   # Since classes of service are loaded very often, their properties are also parsed
139   # often, and invalid property definitions can lead to excessive log pollution.
140   #
141   # Enabling this setting will limit the frequency of logging invalid properties,
142   # but each invalid property will be logged (as an error) at least once within the
143   # configured time period per class of service.
144   #
145   # The implementation uses a loading cache underneath, which causes some memory usage
146   # overhead, although limited by the maximum size configuration below.
147   #
148   # Note that it is not reloadable and requires a restart to change.
```

```
149  #
150  # Defaults to: false
151  com.openexchange.cloudplugins.cos.invalid.property.log.limiter.enable=false
152
153  # When enabled, configures the amount of time before a once logged combination of
154  # (class of service name, property) appears in the log again.
155  #
156  # Note that it is not reloadable and requires a restart to change.
157  #
158  # Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
159  # Defaults to 4h.
160  com.openexchange.cloudplugins.cos.invalid.property.log.limiter.quietPeriod=4h
161
162  # When enabled, the maximum amount of distinct combinations of (class of service name,
163  # property) that are kept track of to prevent excessive logging.
164  #
165  # This is to prevent excessive memory usage when keeping track.
166  #
167  # Note that it is not reloadable and requires a restart to change.
168  #
169  # Defaults to 256.
170  com.openexchange.cloudplugins.cos.invalid.property.log.limiter.maxSize=256
171
172  # Optional rate limiting for logging errors when failing to access LDAP to retrieve
173  # oxCloudClassOfService or oxCloudUser entries.
174  #
175  # Must be specified as a floating point rate value of allowed log entries
176  # per second (e.g. "1.0" means "log once per second", "0.01" means "log once per
177  # 100 seconds").
178  #
179  # Can also be set to "OFF", in which case no rate limiting is performed.
180  #
181  # Note that it is not reloadable and requires a restart to change.
182  #
183  # Example:
184  # com.openexchange.cloudplugins.cos.log.rate.limit=OFF
185  # com.openexchange.cloudplugins.cos.log.rate.limit=0.01
186  #
187  # Defaults to logging once per 5min (1 / 300 = 0.0033).
188  com.openexchange.cloudplugins.cos.log.rate.limit=0.0033
189
190  # When the oxCosDN attribute of UserEntity objects are changed through
         CloudManagementService,
191  # there are three options on how it is handled:
192  # - DISTRIBUTED: produces a distributed event that is broadcasted to all nodes and informs
193  #     them of injecting the new oxCosDN value into their respective cache (only if it
         already
194  #     exists in the cache); note that this setting also means that the implementation will
195  #     listen for remotely triggered events of that type
196  # - LOCAL: only updates the cache in the same App Suite node, locally in memory
197  # - OFF: does not update the cache at all
198  #
199  # Note that it is not reloadable and requires a restart to change.
200  #
201  # Examples:
202  # com.openexchange.cloudplugins.cos.cache.invalidation.mode=OFF
203  # com.openexchange.cloudplugins.cos.cache.invalidation.mode=LOCAL
204  #
205  # Defaults to DISTRIBUTED
206  com.openexchange.cloudplugins.cos.cache.invalidation.mode=DISTRIBUTED
207
208  ###
209  ### Class of Service Provisioning
210  ###
211  ### The following section configures the behavior of the provisioning APIs
212  ### with regards to class of services.
213  ### This applies to RMI, SOAP and command-line operations (createuser, changeuser) as
214  ### well as to the Cloud-Plugins user management REST API under /oxaas/v1/admin/contexts
         /...
215  ###
216
217  # Name of the user attribute that contains the class of service name.
```

```
218  #
219  # Class of services to apply to users must be specified as a user property with the
220  # namespace and name specified in the following configuration property.
221  # When the property is set to "OFF", class of service support is disabled in the
         provisioning
222  # APIs (RMI, SOAP and command-line -- but not in the Cloud-Plugins user management REST
         API).
223  #
224  # It must be specified in the form namespace//name
225  #
226  # Defaults to: cloud//service
227  #
228  # Note that the value of that user attribute may contain multiple names of classes of
229  # service, separated by ',' (whitespaces are stripped), e.g.:
230  # createuser ... --cloud/service=cloud_pim,cloud_security ...
231  #
232  # Examples:
233  # com.openexchange.cloudplugins.cos.provisioning.user.attribute=config//cos
234  # com.openexchange.cloudplugins.cos.provisioning.user.attribute=OFF
235  #
236  com.openexchange.cloudplugins.cos.provisioning.user.attribute=
237
238  # When com.openexchange.cloudplugins.cos.provisioning.user.attribute above is not "OFF",
239  # this property determines whether a class of service is mandatory when performing
240  # createuser operations.
241  # It is never mandatory for changeuser operations (the absence of the class of service
242  # information in a changeuser simply means that it is left untouched).
243  #
244  # Defaults to false.
245  com.openexchange.cloudplugins.cos.provisioning.mandatory=false
246
247  # When com.openexchange.cloudplugins.cos.provisioning.user.attribute above is not "OFF",
248  # this property determines whether a class of service may be empty in createuser and
249  # changeuser operations.
250  #
251  # Defaults to true.
252  com.openexchange.cloudplugins.cos.provisioning.allow.empty=true
253
254  # When com.openexchange.cloudplugins.cos.provisioning.user.attribute above is not "OFF",
255  # this property determines whether a missing class of service in createuser is to be
256  # understood as an empty list of classes of service.
257  # This only affects createuser, since the meaning of the absence of a list of classes of
258  # service in changeuser means leaving the classes of service value untouched.
259  # Note that enabling this property does not automatically mean that an empty list of
260  # classes of service is valid -- that is controlled separately using the property
261  # com.openexchange.cloudplugins.cos.provisioning.allow.empty.
262  #
263  # Defaults to true.
264  com.openexchange.cloudplugins.cos.provisioning.missing.means.empty=true
265
266  # When not empty, the following property defines an exhaustive list of class of service
         names that
267  # are valid and that must be matched.
268  # If the service names that are specified in provisioning operations are not part of this
         comma
269  # separated list, an error will abort the provisioning operation.
270  # Whitespaces are trimmed.
271  #
272  # To avoid using this list and perform LDAP lookups instead (or no validation at all),
273  # leave this property empty or commented out.
274  #
275  # Example:
276  # com.openexchange.cloudplugins.cos.provisioning.validate.list=cloud_pim,
         cloud_productivity
277  #
278  # Defaults to empty.
279  com.openexchange.cloudplugins.cos.provisioning.validate.list=
280
281  # When the validate.list property above is empty, the following property configures
         whether
282  # 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
```

```
354   # Fine-tune the concurrency level hint for the cache, which controls the amount of
355   # buckets and locks for concurrent thread access to the cache.
356   #
357   # Defaults to 32.
358   com.openexchange.cloudplugins.cos.definition.cache.concurrencyLevel=32
359
360   # The maximum amount of oxCloudClassOfService values that may be cached.
361   #
362   # When the maximum size is reached and entries need to be evicted, the last recently
363   # used entries will be removed.
364   #
365   # When left empty, no maximum cache size limit is applied, which is the default.
366   # Example:
367   # com.openexchange.cloudplugins.cos.definition.cache.maxSize=100
368   com.openexchange.cloudplugins.cos.definition.cache.maxSize=
369
370   # How long individual cached entries should be kept in the cache after having been
371   # queried.
372   #
373   # Since the config cascade performs multiple query operations to the service (multiple
374   # for each known capability), it is highly recommended to keep cache entries in memory
375   # for a few seconds at least.
376   #
377   # Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
378   # Defaults to 10m.
379   com.openexchange.cloudplugins.cos.definition.cache.expireAfter=10m
380
381   # Whether to enable statistics and metrics for the oxCloudClassOfService lookup cache.
382   #
383   # When enabled (true), the metrics endpoint will expose cache metrics for the
384   # cache named "cloud-plugins-ldap-cos-definition".
385   #
386   # Defaults to false.
387   com.openexchange.cloudplugins.cos.definition.cache.enableStats=false
388
389   # Whether the cache should be flushed when a reloadconfiguration operation is issued
390   # (regardless of whether another configuration parameter changed or not, through
391   # forced-reloadable).
392   #
393   # Defaults to false.
394   com.openexchange.cloudplugins.cos.definition.cache.flushOnReload=false
395
396   # Fine-tune the concurrency level hint for the cache, which controls the amount of
397   # buckets and locks for concurrent thread access to the cache.
398   #
399   # Defaults to 32.
400   com.openexchange.cloudplugins.cos.listall.cache.concurrencyLevel=32
401
402   # How long individual cached entries should be kept in the cache after having been
403   # queried.
404   #
405   # Since the config cascade queries the list of all the known capabilities at
406   # every user login, it is highly recommended to keep cache entries in memory
407   # for a certain time, especially since class of service definitions are not likely
408   # to change often and, if so, don't need to be applied immediately but can wait
409   # until after the cache expires.
410   #
411   # Syntax: *h*m*s*ms, e.g. "10m", "1h", "300ms"
412   # Defaults to 10m.
413   com.openexchange.cloudplugins.cos.listall.cache.expireAfter=10m
414
415   # Whether to enable statistics and metrics for the list of all properties defined
416   # in oxCloudClassOfService properties attributes.
417   #
418   # When enabled (true), the metrics endpoint will expose cache metrics for the
419   # cache named "cloud-plugins-ldap-cos-listall".
420   #
421   # Defaults to false.
422   com.openexchange.cloudplugins.cos.listall.cache.enableStats=false
423
424   # Whether the cache should be flushed when a reloadconfiguration operation is issued
425   # (regardless of whether another configuration parameter changed or not, through
```

```
426  # forced-reloadable).
427  #
428  # Defaults to false.
429  com.openexchange.cloudplugins.cos.listall.cache.flushOnReload=false
430
431  # Whether to enable the announcements feature
432  # Not reloadable.
433  #
434  # Default: false
435  com.openexchange.cloudplugins.announcements.enabled=true
436
437  # The default group name to access the global database.
438  # See /opt/open-xchange/etc/globaldb.yml for more information and
439  # configuration possibilities
440  # Not reloadable.
441  #
442  # Default: default
443  com.openexchange.cloudplugins.announcements.globaldbContextGroup=default
444
445  # The list of available settings for spamlevels.  It defaults to "low,medium,high"
446  # Please note: this is mandatory and must be set
447  com.openexchange.cloudplugins.spamlevels=low,medium,high
448
449  # the default level to use from the above spamlevels settings.  It defaults to medium
450  # Please note: this is mandatory and must be set
451  com.openexchange.cloudplugins.spamlevels.default=medium
452
453  # The antivirus setting. it defaults to true
454  com.openexchange.cloudplugins.antivirus.default=true
```

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

```
1   # Api secret
2   com.openexchange.cloudplugins.dovecot.apiSecret=
3
4   # Dovecot port
5   com.openexchange.cloudplugins.dovecot.port=8080
6
7   # Protocol to use
8   com.openexchange.cloudplugins.dovecot.protocol=http://
9
10  # Dovecot host
11  com.openexchange.cloudplugins.dovecot.host=localhost
12
13  # Path to dovecot commands
14  com.openexchange.cloudplugins.dovecot.path=/doveadm/v1
15
16  # How many contexts per request should be transmitted
17  com.openexchange.cloudplugins.dovecot.contextChunks=100
18
19  # Max number of http connections
20  com.openexchange.cloudplugins.dovecot.maxConnections=100
21
22  # Max number of http connections per host
23  com.openexchange.cloudplugins.dovecot.maxConnectionsPerHost=100
24
25  # The connection timeout in milliseconds
26  com.openexchange.cloudplugins.dovecot.connectionTimeout=5000
27
28  # The socket read timeout in milliseconds
29  com.openexchange.cloudplugins.dovecot.socketReadTimeout=15000
```

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

```
1   # Regex to validate brandName value
2   com.openexchange.cloudplugins.nginx.auth.servlet.brandName.regex= [0-9a-zA-Z.]*
3
4   # Regex to validate user name
5   com.openexchange.cloudplugins.nginx.auth.servlet.uid.regex= [0-9a-zA-Z.@]*
6
7   # Configure the LDAP lookup method to find users using their logins.
8   # It is the method that is used by default when there is no brand specifi
9   # configuration setting).
10  #
11  # Optional, defaults to "uid".
12  #
13  # Possible values:
14  # uid
15  #    attempts to find users by matching their login against the uid attribute.
16  # email
17  #    attempts to find users by matching their login against the alias attribute.
18  # auto
19  #    when the login contains a "@", the "email" method is used and when not,
20  #    the "uid" method is used
21  # uid-or-email
22  #    attempts to find users by matching their login against the uid and the alias
23  #    attributes (either may match)
24  #
25  # Note that this only applies to the default authentication driver. If there
26  # is a custom implementation that is used for a given brand, its behavior is
27  # not influenced by this properties.
28  #
29  # Example:
30  # com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode=auto
31
32  # One may define any number of such settings per brand by setting properties
33  # with the following format for their name:
34  #
35  # com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode.<brand>=<uid|email|auto|uid-or
         -email>
36  #
37  # For the list of possible values, please consult the documentation for
38  # com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode
39  #
40  # Optional, defaults to falling back to the method configured in
41  # com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode
42  #
43  # Note that this only applies to the default authentication driver. If there
44  # is a custom implementation that is used for a given brand, its behavior is
45  # not influenced by these properties.
46  #
47  # Example:
48  #
49  # com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode.brand1=auto
50  # com.openexchange.cloudplugins.nginx.auth.servlet.uid.mode.brand2=uid
```

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

```
1   # Identifier of the default quota driver to use when not
2   # superseded by a per-brand configuration setting (see below).
3   #
4   # This configuration property is optional.
5   # When it is not set, it will attempt to use the "dovecot"
6   # driver, if available (installed).
7   # If the "dovecot" driver is not installed, it will use the
8   # highest ranked driver that is installed.
9   #
10  # For a deterministic approach, it is recommended to set
11  # a value for this configuration setting.
12  #
13  # Example:
```

```
14  # com.openexchange.cloudplugins.quota.default.driver=cassandra
15  com.openexchange.cloudplugins.quota.default.driver=
16
17  # Override the quota driver per brand.
18  #
19  # Use property names that start with
20  # "com.openexchange.cloudplugins.quota.driver."
21  # followed by the brand name (not the complete DN
22  # but just the brand name).
23  #
24  # These are optional and, if not defined, will always fall
25  # back to the driver configured in the property
26  # com.openexchange.cloudplugins.quota.default.driver
27  #
28  # Example:
29  # com.openexchange.cloudplugins.quota.driver.brand1=cassandra
30  # com.openexchange.cloudplugins.quota.driver.otherbrand=cassandra
31  #
```

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

```
1   # Use the legacy "type" value for lookups in the Cassandra
2   # quota table.
3   # In recent deployments, the "type" parameter is "dovecot_mail",
4   # but in previous installments, the "type" used to be null.
5   # Setting this property to true will use null for the "type"
6   # query parameter.
7   #
8   # This property is optional and when not set, defaults to false.
9   #
10  # Example:
11  # com.openexchange.cloudplugins.quota.use.legacy.type=true
12  #
13
14  com.openexchange.cloudplugins.quota.use.legacy.type=false
15
16  # Unified quota performance optimizations.
17  #
18  # The default behavior is to check whether every single user
19  # who's usage quota is retrieved has unified quota enabled or not,
20  # and return data accordingly.
21  #
22  # In most use cases, if not all, unified quota will be enabled or
23  # disabled uniformly
24  # - globally for a platform,
25  # - or globally for a brand,
26  # - or for a whole context,
27  # in which case the implementation can avoid or minimize the amount
28  # of queries it needs to perform in order to determine how to
29  # calculate the quota usage, depending on whether unified quota
30  # is enabled or disabled for a user.
31  #
32  # Two configuration settings govern this behavior:
33  # - the default behavior which is used for every brand on a platform
34  #   (com.openexchange.cloudplugins.quota.unified.quota)
35  # - per-brand behaviors which take precedence over the default
36  #   (com.openexchange.cloudplugins.quota.unified.quota.<brand name>)
37  #
38  # Each of those settings can have one of the following values:
39  #
40  # always
41  #   the implementation will assume that unified quota is enabled for
42  #   all contexts within the brand, or for all contexts within all
43  #   brands if applied to the default setting
44  #
45  # never
46  #   the implementation will assume that unified quota is disabled for
47  #   all contexts within the brand, or for all contexts within all
```

```
48   #   brands if applied to the default setting
49   #
50   # context
51   #   the implementation will only check whether unified quota has
52   #   been enabled for the context the user(s) are in, and not for
53   #   each individual user, assuming that all users within the same
54   #   context are always configured uniformly regarding unified quota,
55   #   be it enabled or disabled
56   #
57   # user
58   #   the implementation will make no assumptions and check whether
59   #   unified quota is enabled or not for every individual user -- this
60   #   is the safest setting, which is why it is the default, but also
61   #   the slowest and should be avoided if possible
62   #
63   # The default setting can be configured using the property
64   # com.openexchange.cloudplugins.quota.unified.quota=...
65   #
66   # It is optional and defaults to "user" (as explained above) if
67   # omitted, commented out or left empty.
68   #
69   # Example:
70   # com.openexchange.cloudplugins.quota.unified.quota=never
71   com.openexchange.cloudplugins.quota.unified.quota=
72
73   # Per-brand settings can be configured using the following
74   # prefix, followed by the name of the brand:
75   # com.openexchange.cloudplugins.quota.unified.quota.<brand name>=...
76   #
77   # Example:
78   # com.openexchange.cloudplugins.quota.unified.quota.my_brand=always
79   # com.openexchange.cloudplugins.quota.unified.quota.my_other_brand=context
80   #
81   # Those are obviously optional and default to using the default
82   # setting above (which, in turn, when omitted, defaults to "user").
83   #
```

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

```
1    # Where the report and its parts should be stored
2    com.openexchange.cloudplugins.report.storagePath=/tmp
3
4    # How many contexts can be stored in memory before writing them on hdd
5    com.openexchange.cloudplugins.report.maxChunkSize=200
6
7    # How many parallel threads can work on the report
8    com.openexchange.cloudplugins.report.maxThreadPoolSize=20
9
10   # Report thread priority
11   com.openexchange.cloudplugins.report.threadPriority=1
12
13   # Max number of entities that will be included in an ldap search
14   com.openexchange.cloudplugins.report.maxLdapChunks=20000
```

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

```
1
2    # Setting to change the VadeSecure connector identifier referenced in plugins-antiphishing
        .properties / com.openexchange.plugins.antiphishing.connector
3    # Default: "cloudplugins_antiphishing_vadesecure_ldap"
4    # Config-cascade aware: true
5    # Lean: true
6    com.openexchange.cloudplugins.antiphishing.vadesecure.ldap.identifier=
```

```
    cloudplugins_antiphishing_vadesecure_ldap
```

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

```
1   # Identifier of this blackwhitelist connector: cloudplugins_blackwhitelist_ldap
2   # hostname of ldap server
3   com.openexchange.cloudplugins.blackwhitelist.connector.ldap.uri=ldap-fqhn.example.com
4
5   # ldap port
6   com.openexchange.cloudplugins.blackwhitelist.connector.ldap.port=389
7
8   # ldap user
9   com.openexchange.cloudplugins.blackwhitelist.connector.ldap.user=cn=oxadmin,o=oxcs
10
11  # ldap password
12  com.openexchange.cloudplugins.blackwhitelist.connector.ldap.passwd=
13
14  # ldap maximum pool size
15  com.openexchange.cloudplugins.blackwhitelist.connector.ldap.pool.size=10
16
17  # ldap max requests before connection is closed
18  # can be set to -1 to be disabled
19  com.openexchange.cloudplugins.blackwhitelist.connector.ldap.pool.max.requests=2000
20
21  # ldap max lifetime in seconds for each connection in the pool
22  com.openexchange.cloudplugins.blackwhitelist.connector.ldap.pool.max.lifetime=120
23
24  # config to enable LDAP SSL connection over ldaps
25  com.openexchange.cloudplugins.blackwhitelist.connector.ldap.useSSL=false
26
27  # Setting to check if memory backed test mock should be started
28  # This connector is identified by cloudplugins_blwl_test
29  # Default: false
30  com.openexchange.cloudplugins.blackwhitelist.connector.ldap.test=false
```

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

```
1   # Defines whether the forward REST API should be enabled or not.
2   #
3   # This parameter is optional and defaults to "false" (disabled).
4   #
5   # Example:
6   # com.openexchange.cloudplugins.admin.forwards.ws.enabled=true
7   com.openexchange.cloudplugins.admin.forwards.ws.enabled=false
```

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

```
1   # The token endpoint identified by the client
2   #
3   # Must be set for each client, default value: ""
4   #
5   # Example:
6   # com.openexchange.cloudplugins.keycloak.oauth.exampleClient.tokenEndpoint=http://
        localhost:8080/auth/realms/demo/protocol/openid-connect/token
7   com.openexchange.cloudplugins.keycloak.oauth.[client].tokenEndpoint=
8
9   # The clientId, if left empty, no clientId will be used
10  #
11  # Optional, default value: ""
```

```
12  #
13  # Example:
14  # com.openexchange.cloudplugins.keycloak.oauth.exampleClient.clientId=customerClient
15  com.openexchange.cloudplugins.keycloak.oauth.[client].clientId=
16
17  # The default-client used for the CloudAuthenticationDriver
18  com.openexchange.cloudplugins.keycloak.oauth.default-client.clientId=
19
20  # The client secret. Must be provided if clientId is set.
21  #
22  # Optional, default value: ""
23  #
24  # Example:
25  # com.openexchange.cloudplugins.keycloak.oauth.exampleClient.clientSecret=123123
26  com.openexchange.cloudplugins.keycloak.oauth.[client].clientSecret=
27
28  # The default-client secret used for the CloudAuthenticationDriver
29  com.openexchange.cloudplugins.keycloak.oauth.default-client.clientSecret=
30
31  # Max connections
32  #
33  # Optional, default value: 100
34  com.openexchange.cloudplugins.keycloak.oauth.maxConnections=100
35
36  # Max connections per host
37  #
38  # Optional, default value: 100
39  com.openexchange.cloudplugins.keycloak.oauth.maxConnectionsPerHost=100
40
41  # Connection timeout in ms
42  #
43  # Optional, default value in ms: 3000
44  com.openexchange.cloudplugins.keycloak.oauth.connectionTimeout=3000
45
46  # Socket read timeout in ms
47  #
48  # Optional, default value in ms: 6000
49  com.openexchange.cloudplugins.keycloak.oauth.socketReadTimeout=6000
50
51  # Refresh time in ms before expiry date
52  #
53  # Optional, default value is ms: 60000
54  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.
119  #
120  # Example:
121  # com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode=auto
122
123  # One may define any number of such settings per brand by setting properties
124  # with the following format for their name:
125  #
126  # com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode.<brand>=<uid|email|
         auto|uid-or-email|userid-contextid>
127  #
128  # For the list of possible values, please consult the documentation for
129  # com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode
130  #
131  # Optional, defaults to falling back to the method configured in
132  # com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode
133  #
134  # Note that this only applies to the default authentication service driver.
135  # If there is a custom implementation that is used for a given brand, its
136  # behavior is not influenced by these properties.
137  #
138  # Example:
139  #
140  # com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode.brand1=auto
141  # com.openexchange.cloudplugins.keycloak.oauth.authentication.uid.mode.brand2=uid
```

File 15   /opt/open-xchange/etc/cloudplugins-loginproxy-forward.yaml

```
1   # This file contains mappings of brand, identifiers and redirect urls.
2   # It must be a YAML mapping, where
3   # * the key is the brand
4   # * the value is a list of properties with key, value
5   ---
6   'brand1':
7   - identifier: https://loginpage1.example.com
8   - another_identifier: https://loginpage2.example.org
9   - default_redirect: https://default.example.com
10  'brand2':
11  - my_ident: https://loginpage1.example.com
12  - default_redirect: https://default.example.com
13  'brand3':
14  - some_other_identifier: https://loginpage1.example.com
```

File 16   /opt/open-xchange/etc/cloudplugins-loginproxy-ws.properties

```
1   # Maximum amount of login proxy lookup requests per second per source IP address.
2   # May be a decimal number.
3   #
4   # Optional, default value: 25.0
5   #
6   # Example:
7   # com.openexchange.cloudplugins.login.proxy.maxRequestsPerSecond=50.0
8   com.openexchange.cloudplugins.login.proxy.maxRequestsPerSecond=25.0
9
10  # Maximal time window, in milliseconds: after a given source IP address has not accessed
11  # the login proxy lookup API, its number of requests per second rate is reset.
12  #
13  # Optional, default value: 300000
14  #
15  # Example:
16  # com.openexchange.cloudplugins.login.proxy.maxRateTimeWindow=60000
17  com.openexchange.cloudplugins.login.proxy.maxRateTimeWindow=300000
18
19  # Strategy to use for reacting to the inability to access the API for a given source
20  # IP address due to surpassing the maxRequestsPerSecond rate.
21  #
22  # Format: it must be one of:
23  # * fail-fast
24  # * block
25  # * timeout:...
26  #
27  # fail-fast
28  #   if the rate limit is exceeded, the API will respond with a 403 Forbidden
29  # block
30  #   if the rate limit is exceeded, the API will block infinitely until the rate limit
31  #   allows for another request to be performed
32  # timeout:...
33  #   block until the specified timeout is reached, after which the API responds with a
34  #   403 Forbidden
35  #   The value after "timeout:" consists of a number followed by a time unit, examples:
36  #   - timeout:400s ---> 400 seconds
37  #   - timeout:1m ------>   1 minute
38  #   - timeout:2000ms -> 2000 milliseconds
39  #
40  # Optional, default value: timeout:5s
41  #
42  # Example:
43  # com.openexchange.cloudplugins.login.proxy.strategy=timeout:10s
44  com.openexchange.cloudplugins.login.proxy.strategy=timeout:5s
45
46  # The default URL to redirect to when the user is not marked as not migrated
47  # and the identifierof the user is not mapped in cloudplugins-loginproxy-forward.yaml
48  # and the brand does not have a default redirect in cloudplugins-loginproxy-forward.yaml
49  #
50  # Example:
```

```
51   # com.openexchange.cloudplugins.login.proxy.default.redirect=https://example.com/mail
52   # Default: not set
53   com.openexchange.cloudplugins.login.proxy.default.redirect=
```

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

```
1    # This is the CloudPlugins MailFilterInterceptorDriver configuration
2    #
3    # Enable drivers by adding at least one brand in the brands property for that driver
4    # on the server level configuration. If no brand exists, the driver will not be registered
5    #
6    # Some drivers may also have additional configurations
7
8
9    ##### Driver Brand Lists #####
10   # Comma delimited lists
11
12   # Brands that the RedirectStatusDriver should be enabled for
13   #
14   # Optional - default is no brands
15   com.openexchange.cloudplugins.mailfilter.intercept.drivers.redirect.status.driver.brands=
16
17   # Brands that the RedirectBlacklistDriver should be enabled for
18   #
19   # Optional - default is no brands
20   com.openexchange.cloudplugins.mailfilter.intercept.drivers.redirect.blacklist.driver.
         brands=
21
22   ##### End Driver Brand Lists #####
23
24
25
26   ##### Driver Specific configurations #####
27
28   ### Redirect Blacklist Driver ###
29   #
30   # Set to true to enable config cascade for all properties of the Redirect Blacklist Driver
         .
31   # This should be used to set different configurations per brand or an even lower level.
32   # This property is NOT config cascade aware as it is used to control use of it.
33   # Even the driver brand list property can be config cascade if this is enabled which would
          be useful
34   # to enable it for a brand, but disable it for some users
35   #
36   # Optional - default is false
37   com.openexchange.cloudplugins.mailfilter.intercept.drivers.redirect.blacklist.driver.
         configcascade.enable=false
38   #
39   # The comma+space delimited list of regular expressions that are blacklisted for mail
         filter redirects.
40   # Java regular expressions are supported here, so non regex characters must be escaped.
41   # Example: abuse@.*, spam@domain\.com  <-- '.' has been escaped to match only '.' and not
         any char
42   # See for regex contructs: https://docs.oracle.com/javase/7/docs/api/java/util/regex/
         Pattern.html
43   # WARNING: You must use ", " (comma and a space) as the separator to separate the regexs
44   #
45   # Optional - default is empty which should only be left if this driver is not used
46   com.openexchange.cloudplugins.mailfilter.intercept.drivers.redirect.blacklist.driver.
         blacklist=
47   #
48   # Set to true to only blacklist true autoforward rules which are defined by having the "
         autoforward" flag.
49   # If set to false, then all redirect rules will be checked for the blacklist addresses
50   #
51   # Optional - default is true
52   com.openexchange.cloudplugins.mailfilter.intercept.drivers.redirect.blacklist.driver.
         autoforward.only=true
```

```
53  #
54  ### End Redirect Blacklist Driver ###
55
56  ##### End Driver Specific configurations #####
```

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

```
1   # Configure the master password per brand.
2   #
3   # If the brand is configured here, master auth will be enabled and it will
4   # override normal authentication - meaning that users would not be able to
5   # login if normal authentication is used. This would not impact SSO.
6   #
7   # Example:
8   #
9   # com.openexchange.authentication.cloudplugins.brand.master.auth.brand1.password=secret
10  # com.openexchange.authentication.cloudplugins.brand.master.auth.brand2.password=secret
```

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

```
1   # The max number of entries to cache.
2   # Caching is used for brand lookup and valid DNS results.
3   # Use 0 for no cache.
4   #
5   # Default: 10000
6   #
7   com.openexchange.cloudplugins.mx.checker.cache.max=10000
8
9   # The amount of time in seconds after a value is written to the cache
10  # until it expires.
11  # Caching is used for brand lookup and valid DNS results.
12  # Use 0 for no cache.
13  #
14  # Default: 3600
15  #
16  com.openexchange.cloudplugins.mx.checker.cache.expire.seconds=3600
```

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

```
1   # The properties for cloud setup OIDC bundle
2
3   # The general oidc property to enable or disable the core oidc registry
4   com.openexchange.oidc.enabled=true
5
6   ### General settings
7   ################################################################################
8   # Regex to validate user name
9   com.openexchange.cloudplugins.oidc.user.regex=[0-9a-zA-Z.@]*
10
11  # All properties below can also be assigned to an identifier by configuring them as
12  # com.openexchange.cloudplugins.oidc.<identifier>.key
13  #
14  # If a property is not set for an identifier, the key without the identifier is used as a
       default
15  #
16  # Example:
17  #    com.openexchange.cloudplugins.oidc.example.id=someValue
18  #    com.openexchange.cloudplugins.oidc.brand=someBrand
19  #    com.openexchange.cloudplugins.oidc.example.brand=
```

```
20  #     example brand => someBrand
21  #
22  # !! Note !!
23  #     The list of possible OIDCBackends is identified by the property
24  #     com.openexchange.cloudplugins.oidc.enabled.<identifier>
25  #
26  # Example:
27  #     com.openexchange.cloudplugins.oidc.enabled=true
28  #     com.openexchange.cloudplugins.oidc.enabled.myIdentifier=true
29  #     com.openexchange.cloudplugins.oidc.enabled.moreIdentifier=true
30  #
31  #     It is possible to disable certain OIDCBackends by configuration
32  #
33  #     com.openexchange.cloudplugins.oidc.enabled is also valid and will be used as an empty
        identifier
34  #     An empty identifier will use all properties set in this config file.
35  #
36  # com.openexchange.cloudplugins.oidc.enabled.<identifier>=true
37  com.openexchange.cloudplugins.oidc.enabled=false
38
39  ### OIDC Specific configuration
40  ###################################################################################
41  # All properties mentioned at https://documentation.open-xchange.com/components/middleware
        /config/7.10.2/#mode=features&feature=OpenID
42  # can be assigned to each OIDCbackend.
43  #
44  # !! NOTE !!
45  #     The default prefix is not 'com.openexchange.oidc'
46  #     'com.openexchange.cloudplugins.oidc'
47  #     is used instead meaning that there is an additional
48  #     'cloudplugins' in between to differentiate from the general oidc configuration.
49  #
50  # Example:
51  #     com.openexchange.oidc.clientId will not be evaluated
52  #     com.openexchange.oidc.cloudplugins.clientId is the correct default key
53  #
54  # The id inside the jwt token response which holds the userinformation
55  #
56  # The search for the id is done in the attributeStatement if nothing is configured
57  # possible configuration values are:
58  #     <not_set>       // the subject is used
59  #     claim:key       // a claim with the identifier <key> is used
60  #     key             // a claim with the identifier <key> is used
61  #
62  # In any case, the id must match the uid used to provision the user
63  # com.openexchange.cloudplugins.oidc.<identifier>.id=
64  com.openexchange.cloudplugins.oidc.id=
65
66  # The ldap lookup used by this OIDCBackend
67  #
68  # Configure the LDAP lookup method to find users using their identifiers.
69  #
70  #
71  # Possible values:
72  # uid
73  #   attempts to find users by matching their login against the uid attribute.
74  # email
75  #   attempts to find users by matching their login against the alias attribute.
76  # auto
77  #   when the login contains a "@", the "email" method is used and when not,
78  #   the "uid" method is used
79  # uid-or-email
80  #   attempts to find users by matching their login against the uid and the alias
81  #   attributes (either may match)
82  #
83  # This property is optional.
84  # Default: uid
85  # com.openexchange.cloudplugins.oidc.<identifier>.ldapLookup=
86  com.openexchange.cloudplugins.oidc.ldapLookup=
87
88
89  # The brand to use for OXaaS LDAP authentication operations.
```

```
 90    # Uses the host name when empty or not set.
 91    #
 92    # This property is mandatory.
 93    # Default: <empty>
 94    # com.openexchange.cloudplugins.oidc.<identifier>.brand=
 95    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
 85  com.openexchange.cloudplugins.remote.ldap.authStrategy=
 86
 87  # The name of the external LDAP pool to be used to search the user if searching is enabled
          via bindDnStrategy
 88  # Optional, will use the authLdapIdentifier if not set
 89  #
 90  # Example:
 91  # com.openexchange.cloudplugins.remote.ldap.searchLdapIdentifier.nginx.brand1=
        brand1_search_pool
 92  com.openexchange.cloudplugins.remote.ldap.searchLdapIdentifier=
 93
 94  # Optionally change the uid lookup in our own LDAP
 95  #
 96  # Possible values:
 97  #  * keep
 98  #      do not change the uid
 99  #  * add-default-domain:example.com
100  #      will add the default domain to an existing uid if it doesn't contain a @
101  #
102  # will use 'keep' if not set
103  #
104  # Example:
105  # com.openexchange.cloudplugins.remote.ldap.localUid.nginx.brand1=add-default-domain:
        example.com
106  com.openexchange.cloudplugins.remote.ldap.localUid=
107
108  # Optionally change the uid lookup in the external LDAP
109  #
110  # Possible values:
111  #   * keep
112  #       do not change the uid
113  #   * uid
114  #       use the uid of the UserEntity
115  #   * add-default-domain:example.com
116  #       will add the default domain to an existing uid if it doesn't contain a @
117  #   * alias | any-alias
118  #       uses any of the existing aliases from the userEntity object that isn't the catch-
        all alias
119  #   * mail | primary-mail
120  #       uses the primary mail of the user object which requires a database lookup via
        userId and contextId
121  #
122  # will use 'keep' if not set
123  #
124  # Example:
125  # com.openexchange.cloudplugins.remote.ldap.remoteUid.nginx.brand1=mail
126  com.openexchange.cloudplugins.remote.ldap.remoteUid=
127
128  # LDAP search filter for finding mailboxDN in remote LDAP
129  # Possible values:
130  #   * search:filter
131  #       search in the remote LDAP with the baseDN and the searchLdapIdentifier ldap pool
132  #   * parse:filter
133  #       parses existing data into a usable DN
134
135  #   Can use the following data:
136  #    * {{local}} if input contains an @
137  #    * {{domain}} if input contains an @
138  #    * {{uid}} as full input string
139  #    * {{{baseDN}}} the configured baseDN property, can be empty.
140  #       Note that the baseDN should be escaped with 3 { } as it would otherwise be
        escaped
141  #       This is also true for other values. Use 3 { } if those should not be escaped
142  #
143  # Mandatory, there is no default value.
144  #
```

```
145   # Example :
146   # com.openexchange.cloudplugins.remote.ldap.bindDnStrategy.nginx.brand1=search:uid={{uid}}
147   # com.openexchange.cloudplugins.remote.ldap.bindDnStrategy.nginx.brand1=parse:cn={{local
          }},ou={{domain}},{{{baseDN}}}
148   com.openexchange.cloudplugins.remote.ldap.bindDnStrategy=
149
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
196    # attributes in IdP responses and will be contained in the service providers metadata XML.
197    # The according endpoint is always registered with '{prefix}/saml/{samlPath}/sls' as
           servlet alias.
198    #
199    # This property is mandatory if 'com.openexchange.cloudplugins.saml.enableSingleLogout' is
           'true'.
200    # Default: <empty>
201    # Example: https://appsuite.example.com/appsuite/api/saml/{samlPath}/sls
202    com.openexchange.cloudplugins.saml.slsURL=
203
204    # The binding via which logout responses shall be sent to the IdP on IdP-initiated single
205    # logout flows. Must be 'http-redirect' or 'http-post'.
206    #
207    # This property is mandatory if 'com.openexchange.cloudplugins.saml.enableSingleLogout' is
           'true'.
208    # Default: http-redirect
209    com.openexchange.cloudplugins.saml.logoutResponseBinding=http-redirect
210
211    # The HTML template to use when logout responses are sent to the IdP via HTTP POST.
212    # The template must be located in '/opt/open-xchange/templates'.
213    #
214    # This property is mandatory if 'com.openexchange.cloudplugins.saml.enableSingleLogout' is
           'true'
215    # and 'com.openexchange.cloudplugins.saml.logoutResponseBinding' is set to 'http-post'.
216    # Default: saml.logout.response.html.tmpl
217    com.openexchange.cloudplugins.saml.logoutResponseTemplate=saml.logout.response.html.tmpl
218
219    # The entity ID of the IdP. It will be used to validate the 'Issuer' elements of SAML
           responses.
220    #
221    # This property is mandatory.
222    # Default: <empty>
223    com.openexchange.cloudplugins.saml.idpEntityID=
224
225    # The URL of the IdP endpoint where authentication requests are to be sent to.
226    #
227    # This property is mandatory.
228    # Default: <empty>
229    com.openexchange.cloudplugins.saml.idpAuthnURL=
230
231    # The URL of the IdP endpoint where logout requests are to be sent to.
232    #
233    # This property is mandatory if 'com.openexchange.cloudplugins.saml.enableSingleLogout' is
           'true'.
234    # Default: <empty>
235    com.openexchange.cloudplugins.saml.idpLogoutURL=
236
237    # It is possible to enable a special kind of auto login mechanism that allows user agents
           to
238    # re-use an existing OX session if it was created during the same browser session. If
           enabled,
239    # a special cookie will be set, which is linked to the OX session and bound to the browser
           sessions
240    # life time. The advantage of this mechanism is, that sessions are simply re-entered if
           the user
241    # refreshes his browser window. He is then also able to open more than one tab of OX App
           Suite
242    # at the same time. This mechanism can only re-use sticky sessions, i.e. it is mandatory
           that the
243    # requests are always routed to the same backend for a certain session.
244    #
245    # --- SECURITY WARNING ---
246    # Enabling this setting is not compliant to the SAML specification as it bypasses the IdP
           in
247    # certain cases. Additionally in scenarios where a public device is used, a foreign user
```

```
         might
248  # take over a formerly authenticated users session if that user forgets to log out and
         doesn't
249  # close his web browser (even if he closes the App Suite tab). As no login screen is
         displayed
250  # by OX in SAML environments, the user is even not able to decide, whether the application
         shall
251  # remember him or not.
252  #
253  # Default: false
254  com.openexchange.cloudplugins.saml.enableAutoLogin=false
255
256  # Whether unsolicited responses will be accepted or not.
257  #
258  # Default: true
259  com.openexchange.cloudplugins.saml.allowUnsolicitedResponses=true
260
261  # Whether SAML-specific auto-login is enabled, that uses the SessionIndex of the
         AuthnResponse
262  #
263  # Default: false
264  com.openexchange.cloudplugins.saml.enableSessionIndexAutoLogin=false
265
266  # List of hosts where that this SAMLBackend is responsible for
267  # if all is present, this SAMLBackend responsible for all hosts
268  #
269  # It is possible to control the samlPath with this property.
270  # Another way would be to set the samlPath within the as-config.yml.
271  # Default: <empty>
272  com.openexchange.cloudplugins.saml.hosts=
```

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

```
 1   ### Configuration for LDAP support for Trusted Identity.
 2
 3   # Storage Keys are used to decrypt private ECDSA keys that are stored in LDAP.
 4   # LDAP oxCloudTrustedIdentityKeyPair entities contain optional (but
 5   # strongly encouraged) references to storage keys by name.
 6   # Storage keys are symmetric/secret keys (AES is recommended).
 7   # Those storage keys are defined here by configuration, with multiple parameters,
 8   # that are defined using different prefixes but the same storageKeyName part in
 9   # each property name:
10   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.{storageKeyName}=...
11   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.keyType.{storageKeyName
         }=...
12   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.cipher.{storageKeyName
         }=...
13   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.alias.{storageKeyName}=...
14   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.secret.{storageKeyName
         }=...
15   #
16   # For example, using "sk1" as the {storageKeyName}:
17   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.sk1=keystore:/opt/
         open-xchange/etc/sk1.jks
18   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.keyType.sk1=AES
19   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.cipher.sk1=AES/GCM/
         NoPadding
20   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.alias.sk1=storageKey
21   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.secret.sk1=secret
22
23   # The storage key file location is a fully qualified path to either a plain
24   # file that contains the encoded bytes of the symmetric key, or a Java KeyStore
25   # file.
26   # When using a keystore file, one may also want to configure the key alias and secret
27   # (see next properties below.)
28   #
29   # Property format:
30   # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.{storageKeyName}=...
```

```
31  # When the file is a KeyStore file, it must be prepended with "keystore:":
32  # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.{storageKeyName}=
        keystore:...
33  #
34  # Example of a plain file:
35  # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.sk1=/opt/open-xchange
        /etc/sk1.key
36  #
37  # Example of a KeyStore file:
38  # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.sk1=keystore:/opt/
        open-xchange/etc/sk1.jks
39  #
40  # The value is mandatory and has no default.
41  #
42  # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.file.storageKeyName=
43
44  # KeyStore key alias: when using a KeyStore file, defines the alias of the key entry to
        use
45  # as the symmetric/secret key.
46  #
47  # Example:
48  # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.alias.sk1=storageKey
49  #
50  # The value is optional when the KeyStore file contains a single entry.
51  #
52  # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.alias.storageKeyName=
53
54  # KeyStore secret: when using a KeyStore file, defines the password to use to
55  # decrypt the KeyStore as well as the key inside of it.
56  #
57  # Example:
58  # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.secret.sk1=my_secret
59  #
60  # The value is optional and defaults to an empty string ("").
61  #
62  # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.secret.storageKeyName=
63
64  # The cipher algorithm defines which symmetric decryption algorithm to use when
65  # unwrapping the private key from LDAP, and must be the same as the cipher used
66  # when encrypting it in the first place.
67  #
68  # Example:
69  # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.cipher.sk1=AES/CBC/
        PKCS5Padding
70  #
71  # The value is optional and defaults to AES/GCM/NoPadding
72  #
73  # com.openexchange.cloudplugins.trustedidentity.ldap.storageKey.cipher.storageKeyName=
74
75  # Storage Key caching: keys that are looked up in LDAP are cached in memory for
        performance
76  # reasons.
77  # The following configuration property determines how long they are kept in cache
78  # before being fetched from LDAP again:
79  # Format: duration[d|h|m|s|ms]
80  #
81  # Example:
82  # com.openexchange.cloudplugins.trustedidentity.ldap.cache.ttl=4h
83  #
84  # Optional, the default value is 1h (one hour).
85  com.openexchange.cloudplugins.trustedidentity.ldap.cache.ttl=
```

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

```
1  # Setting to control the uri of the tarent adapter
2  com.openexchange.oxaas.alias.tarent.uri=http://localhost
3
4  # Setting to control allowed domains
```

```
 5    com.openexchange.oxaas.alias.allowed.domains=
 6
 7    # Setting to control if alias adapter should be in test mode (this means a mock is used)
 8    com.openexchange.oxaas.alias.test=false
 9
10    # Loglevel for the internal OkHttp3 client
11    # Allowed values are: NONE, BASIC, HEADERS, BODY
12    com.openexchange.oxaas.alias.tarent.loglevel=NONE
13
14    # Setting to control if Unsecured Https should be allowed or not
15    # Default: false
16    com.openexchange.oxaas.alias.tarent.allowUnsecuredHttps=false
17
18    # Master user needed to delete alias
19    com.openexchange.oxaas.alias.master.user=
20
21    # Master user password needed to delete alias
22    com.openexchange.oxaas.alias.master.password=
23
24    # Setting to control if a client cert shoul be loaded, must be in PKCS 12 format
25    # Can be mixed with com.openexchange.oxaas.alias.tarent.allowUnsecuredHttps
26    # allowUnsecuredHttps=true and clientcert.path=set
27    #    a client cert is used but the hostname is not verified and all server certs are
         trusted
28    # allowUnsecuredHttps=false and clientcert.path=set
29    #    a client cert is used, but hostname is verified and server certs must be trustable
30    # Default: empty
31    com.openexchange.oxaas.alias.tarent.ssl.clientcert.path=
32
33    # Setting that holds the password for the PKCS 12 container
34    # Default: empty
35    com.openexchange.oxaas.alias.tarent.ssl.clientcert.password=
36
37    # Default number of aliases to be configured by each userr
38    # Default: 15
39    com.openexchange.oxaas.aliasquota=15
```

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

```
 1    # Config cascade-aware property to control the prefix of the users templates
 2    # Used to show the overquota warning mails
 3    #
 4    # Format: prefix[.language|].quota.[html|text|subject].tmpl
 5    com.openexchange.oxaas.mail.quota.notify.prefix=notify.oxaas.over.quota
 6
 7    # Config cascade-aware property to control the prefix of the users templates
 8    # Used to show the welcome mail
 9    #
10    # Format: prefix[.language|].[html|text|subject].tmpl
11    com.openexchange.oxaas.mail.welcome.mail.notify.prefix=notify.oxaas.welcome.mail
12
13    # Config cascade-aware property to control the prefix of the users templates
14    # Used to show disabled sent spam mail
15    #
16    # Format: prefix[.language|].[html|text|subject].tmpl
17    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

```
 1    # Config-cascade aware setting to control the quotas that should be monitored
 2    com.openexchange.oxaas.mail.quota.drive.quotas=90,100
 3
 4    # Setting to control if the quota notification should be ignored for context-wide
```

```
      filestores
5   # This makes sense, if the users are having user quota and there is a context-quota in
      place too
6   # Config-Cascade only on context level
7   com.openexchange.oxaas.mail.quota.drive.ignoreContextQuota=false
8
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=
```