How-To Guides

Table of Contents

Active Directory Identity Management . . . 2
- Introduction . . . 2
- OpenIDM Configuration . . . 2
- Testing the Solution . . . 6

Can Your IDM Play Chess? . . . 11
- Introduction . . . 11
- Starting OpenIDM . . . 11
- Initial OpenIDM Setup . . . 12
- Playing Chess . . . 14
- Some Technical Details . . . 17

Configuring OpenIDM for Synchronization Between Active Directory and OpenDJ . . . 19
- Introduction . . . 19
- Configuring OpenIDM . . . 19
- Configuring the data sources . . . 19
- Configuring Active Directory → OpenDJ synchronization . . . 20
- Configure OpenDJ → Active Directory synchronization . . . 23
- Verifying the solution . . . 25

Configuring OpenIDM for Synchronization Between OpenLDAP and OpenDJ . . . 27
- Introduction . . . 27
- OpenIDM Setup . . . 27
- Data Sources Setup . . . 27
- Configuring OpenLDAP → OpenDJ Synchronization . . . 28
- Configuring OpenDJ → OpenLDAP Synchronization . . . 31
- Test the Solution . . . 33

How to Persist OpenIDM Configuration Data Between Restarts . . . 35

Practical tutorials for deploying OpenIDM, managing identities, *synchronizing directories, and using workflows. Examples may target particular product versions; check each tutorial’s prerequisites.*

Active Directory Identity Management

Introduction

In this article, we will configure Active Directory identity management from OpenIDM. Let's look at a typical scenario where HR hires a new employee, and enters their credentials into an account management system (IDM). From this system, the data must be imported into the main directory. Typically, this is Active Directory.

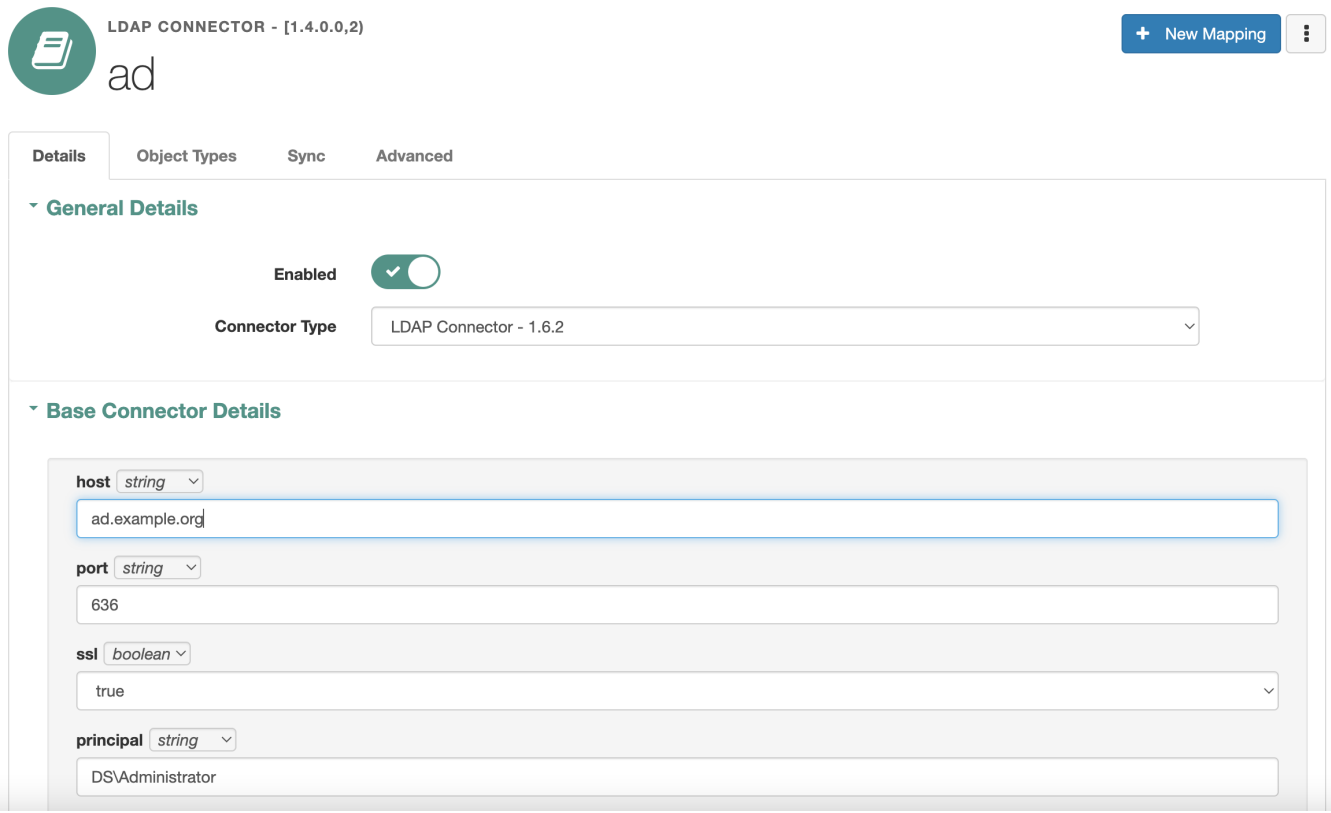
OpenIDM Configuration

How to quickly deploy OpenIDM was described in this article. Therefore, we will assume that you already have OpenIDM deployed.

Active Directory Connector Configuration

From the samples/provisioners directory of the OpenIDM distribution, copy the `provisioner.openicf-adldap.json` file into the `openidm/conf` directory. Change the configuration values to match your environment:

In the administrator console select Configure → Connectors. In the list select AD LDAP Connector and set the settings according to the following table:



Setting	Description	Example value
host	Active Directory hostname or IP	ad.example.org

Setting	Description	Example value
port	Active Directory port	636
ssl	Use SSL connection	true
principal	Active Directory account name	DOMAIN\Administrator
credentials	Account password (string, after saving, the value will be encrypted)	
baseContexts	One or more starting points in the LDAP tree that will be used when searching the tree.	CN=Users,DC=example,DC=org
baseContextsToSynchronize	One or more starting points in the LDAP tree that will be used to determine if a change should be synchronized.	CN=Users,DC=example,DC=org
accountSearchFilter	LDAP Filter for Retrieving Accounts	(&(!(userAccountControl:1.2.840.113556.1.4.803:=2))(!(objectClass=Computer)))
accountSynchronizationFilter	An optional LDAP filter for the objects to synchronize.	(&(!(userAccountControl:1.2.840.113556.1.4.803:=2))(!(objectClass=Computer)))
groupSearchFilter	LDAP Filter for Retrieving Groups	(objectClass=group)
groupSynchronizationFilter	An optional LDAP filter for the objects to synchronize.	(objectClass=group)

These account filters retain the sample configuration's exclusion of disabled accounts and computers.

To synchronize passwords from OpenIDM to Active Directory, the connection must be over a secure protocol. To set up a secure connection, save the certificate from the Active Directory with the command:

```
openssl s_client -showcerts -connect ad.example.org:636 </dev/null 2>/dev/null|openssl
x509 -outform PEM > ad_cert.pem
```

Upload the certificate to the OpenIDM trust store with the command:

```
keytool -import -alias openidm-ad -file ad_cert.pem -storetype JKS -keystore
truststore
```

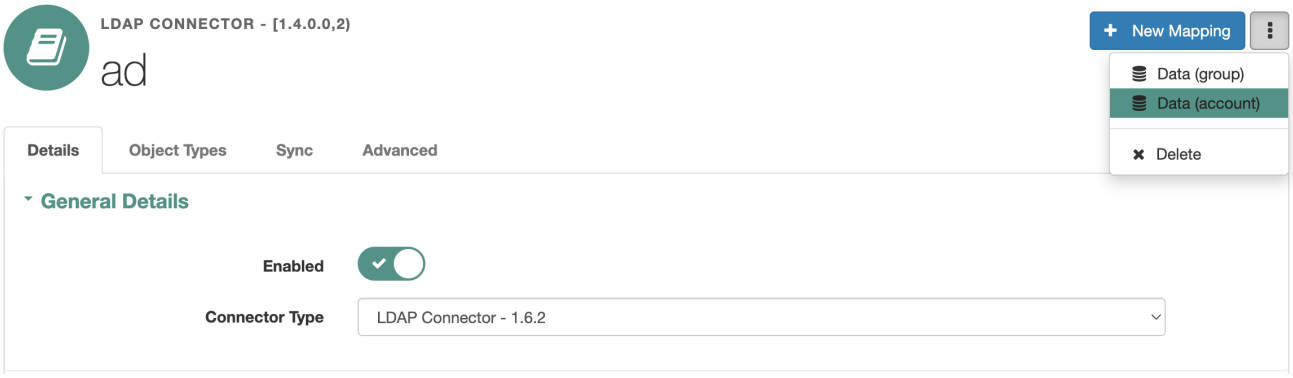
The trust store is located in the OpenIDM distribution directory in the openidm/security subdirectory. The default password for the trust store is `changeit`.

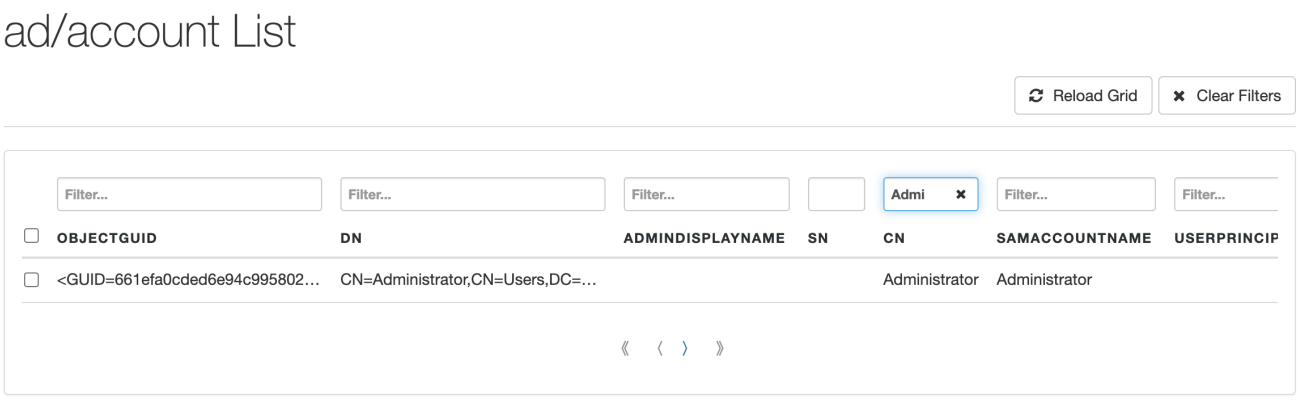
Also, to synchronize passwords from OpenIDM, edit `objectTypes.account.properties` in the Active Directory connector configuration file `provisioner.openicf-adldap.json`. Add a comma after the closing brace of the existing `whenCreated` field, then add the `password` field:

```
"password" : {
    "type" : "string",
    "nativeName" : "__PASSWORD__",
    "nativeType" : "JAVA_TYPE_GUARDEDSTRING",
    "flags" : [
        "NOT_READABLE",
        "NOT_RETURNED_BY_DEFAULT"
    ]
}
```

Active Directory Connector Test

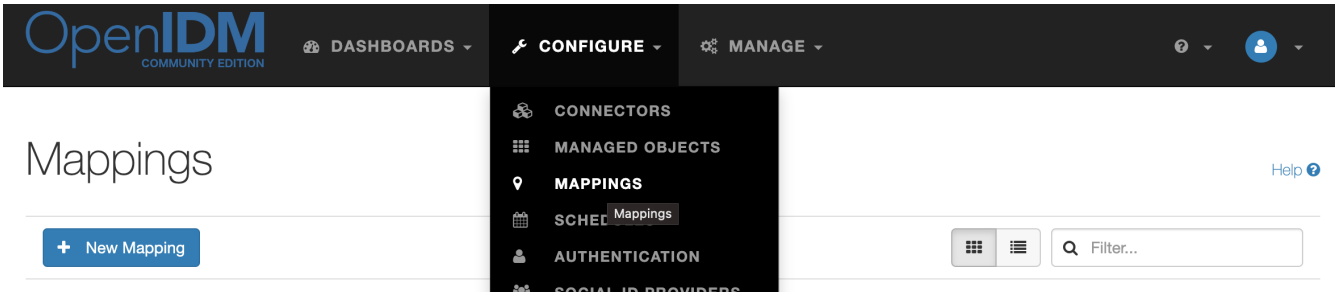
In the UI, in the AD Connector settings at the top right, click the three-dot button and select Data (account). A list of Active Directory accounts will appear



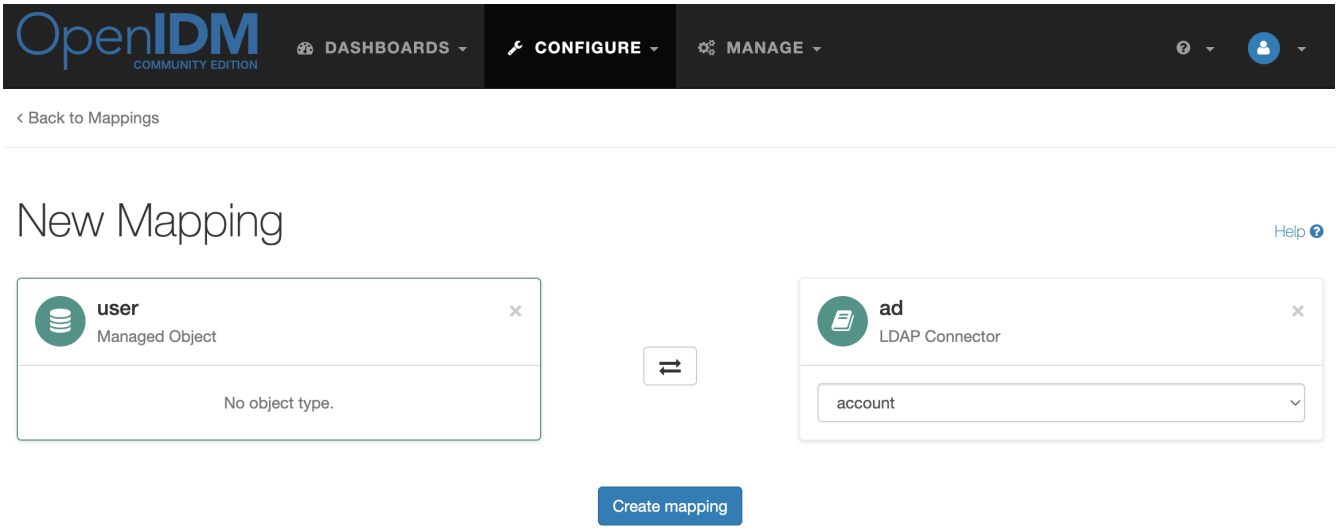


OpenIDM and Active Directory Accounts Synchronization Configuration

In the administrator console, select Configure → Mappings. Click the New Mapping button.



Into the source resource add Managed Object → user. Into the target resource add Connectors → ad.



And click the Create mapping button

In the administrator console, select Configure → Mappings → managedUser_systemAdAccount

In the Attributes Grid, configure the attribute mapping according to the table:

Source	Destination	Transformation script	Conditional updates
userName	dn	`'CN=' + source + ',CN=Users,DC=example,DC=org'`	
givenName	givenName		
sn	sn		
description	description		`!!object.description`
telephoneNumber	telephoneNumber		`!!object.telephoneNumber`
userName	sAMAccountName		
password	password		

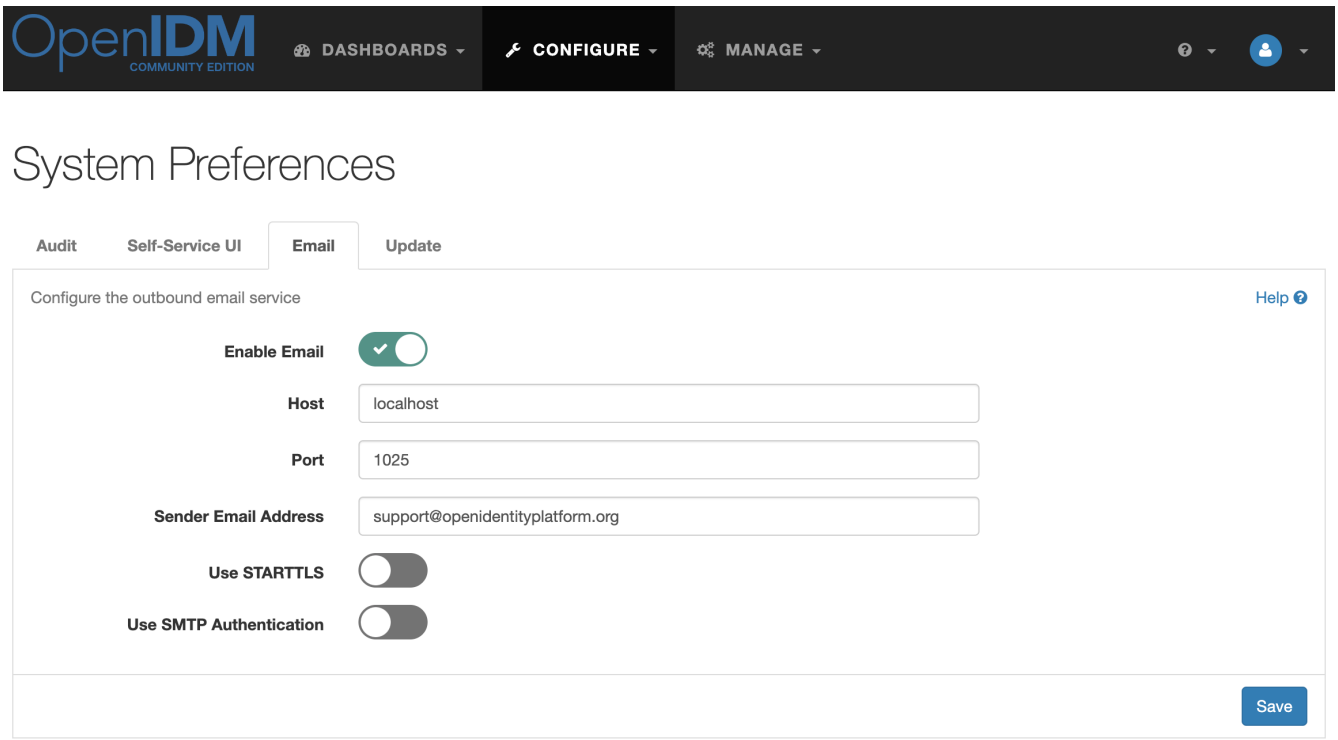
The sample connector marks `cn` as not creatable or updateable, so a separate `cn` mapping would be ignored. Active Directory uses the RDN from the `dn` mapping: for `userName` equal to `idmUser`, `cn` is `idmUser`.

On the Behaviors tab, under Policies, select Default Actions. Change the action for Source Missing (`SOURCE_MISSING`) from Exception to Delete, then click the Save button. This allows deleting a

managed user to delete its linked Active Directory account; Default Actions alone leaves that account in AD.

Sending Password by Email

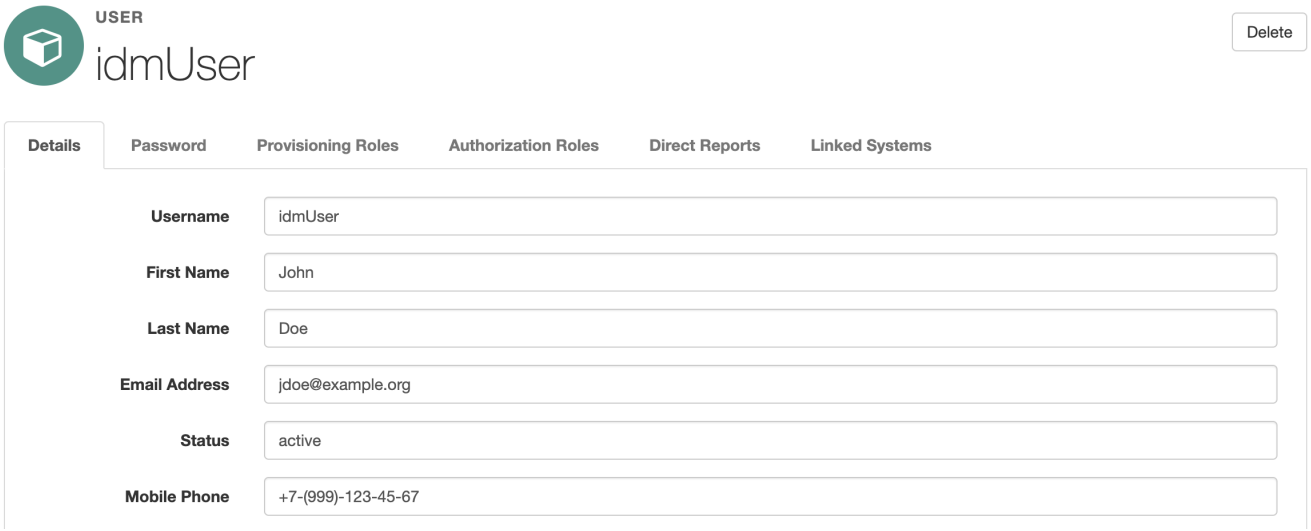
OpenIDM can generate a random password and send it to email. To configure this in the administrator console, go to Configure → System Preferences → Email. Enter the SMTP server settings and click Save



Testing the Solution

Account Creation

Now that we have configured the connection to the Active Directory and synchronization between OpenIDM and AD accounts, let's test how the solution works. In the administrator console, select Manage → User and create a new account by clicking the New User button. Fill in the attributes of the new user and click Save.



You can also create an account using OpenIDM REST API

```
curl 'http://localhost:8080/openidm/managed/user?_action=create' \
 -H 'X-OpenIDM-Username: openidm-admin' \
 -H 'X-OpenIDM-Password: openidm-admin' \
 -H 'Content-Type: application/json' \
 --data-raw
'{"mail":"jdoe@example.org","sn":"Doe","givenName":"John","telephoneNumber":"+7(999)12
3-45-67","userName":"idmUser"}' | json_pp

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   357    0   242  100   115   1340    637 --:--:-- --:--:-- --:--:--  2063
{
   "_id" : "a250eb09-28a1-4fab-b338-e729986f38ec",
   "_rev" : "2",
   "accountStatus" : "active",
   "effectiveAssignments" : [],
   "effectiveRoles" : [],
   "givenName" : "John",
   "mail" : "jdoe@example.org",
   "sn" : "Doe",
   "telephoneNumber" : "+7(999)123-45-67",
   "userName" : "idmUser"
}
```

You can also check if the account has been created using a GET request to the API

```
curl 'http://localhost:8080/openidm/managed/user?_queryFilter=true' \
 -H 'X-OpenIDM-Username: openidm-admin' \
 -H 'X-OpenIDM-Password: openidm-admin' | json_pp
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   380    0   380    0     0  25112      0 --:--:-- --:--:-- --:--:-- 47500
```

```
{
   "pagedResultsCookie" : null,
   "remainingPagedResults" : -1,
   "result" : [
      {
         "_id" : "a250eb09-28a1-4fab-b338-e729986f38ec",
         "_rev" : "2",
         "accountStatus" : "active",
         "effectiveAssignments" : [],
         "effectiveRoles" : [],
         "givenName" : "John",
         "mail" : "jdoe@example.org",
         "sn" : "Doe",
         "telephoneNumber" : "+7(999)123-45-67",
         "userName" : "idmUser"
      }
   ],
   "resultCount" : 1,
   "totalPagedResults" : -1,
   "totalPagedResultsPolicy" : "NONE"
}
```

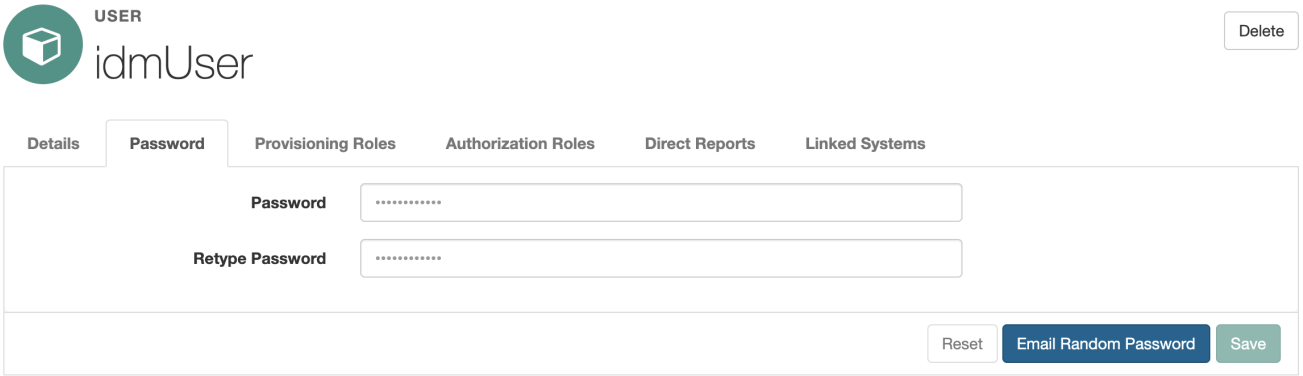

The OpenIDM account will be synchronized in the Active Directory automatically

Check its presence in AD with the command

```
ldapsearch -H ldap://ad.example.org -x -W -D "admin@example.org" -b
"dc=example,dc=org" "(sAMAccountName=idmUser)" | grep dn
Enter LDAP Password:
dn: CN=idmUser,CN=Users,DC=example,DC=org
```

Sending Password by Email

If you want to set a random password for Active Directory, go to the Password tab and click the Email Random Password button.



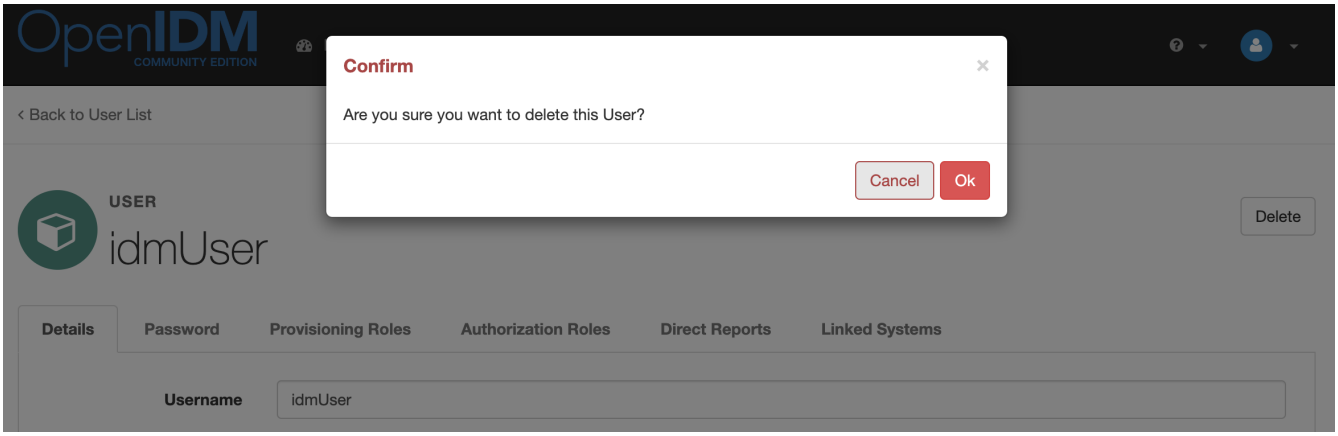
Or with a POST request to the API

```
curl 'http://localhost:8080/openidm/managed/user/a250eb09-28a1-4fab-b338-
e729986f38ec?_action=resetPassword' \
 -X 'POST' \
 -H 'X-OpenIDM-Username: openidm-admin' \
 -H 'X-OpenIDM-Password: openidm-admin' | json_pp
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   242    0   242    0     0    100      0 --:--:--  0:00:02 --:--:--   101
{
   "_id" : "a250eb09-28a1-4fab-b338-e729986f38ec",
   "_rev" : "4",
   "accountStatus" : "active",
   "effectiveAssignments" : [],
   "effectiveRoles" : [],
   "givenName" : "John",
   "mail" : "jdoe@example.org",
   "sn" : "Doe",
   "telephoneNumber" : "+7(999)123-45-67",
   "userName" : "idmUser"
}
```

A new password will be generated and sent to the user by Email. The same password will be set for the Active Directory login.

Account Deletion

In the Administrator Console, go to Manage → User. Select an account and click Delete.



In the pop-up dialog box, click the Ok button.

By using a DELETE request to the OpenIDM API:

```
curl 'http://localhost:8080/openidm/managed/user/a250eb09-28a1-4fab-b338-e729986f38ec'
\
 -X 'DELETE' \
 -H 'X-OpenIDM-Username: openidm-admin' \
 -H 'X-OpenIDM-Password: openidm-admin' | json_pp
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   242    0   242    0     0    756      0 --:--:-- --:--:-- --:--:--   773
{
   "_id" : "a250eb09-28a1-4fab-b338-e729986f38ec",
   "_rev" : "4",
   "accountStatus" : "active",
   "effectiveAssignments" : [],
   "effectiveRoles" : [],
   "givenName" : "John",
   "mail" : "jdoe@example.org",
   "sn" : "Doe",
   "telephoneNumber" : "+7(999)123-45-67",
   "userName" : "idmUser"
}
```

Let's check OpenIDM account deletion via REST API

```
curl 'http://localhost:8080/openidm/managed/user?_queryFilter=true' \
 -H 'X-OpenIDM-Username: openidm-admin' \
 -H 'X-OpenIDM-Password: openidm-admin' | json_pp
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138    0   138    0     0   7306      0 --:--:-- --:--:-- --:--:-- 11500
{
   "pagedResultsCookie" : null,
   "remainingPagedResults" : -1,
   "result" : [],
   "resultCount" : 0,
   "totalPagedResults" : -1,
   "totalPagedResultsPolicy" : "NONE"
}
```

Check if the account exists in Active Directory with the command:

```
ldapsearch -H ldap://ad.example.org -x -W -D "admin@example.org" -b
"dc=example,dc=org" "(sAMAccountName=idmUser)" | grep dn | wc -l
Enter LDAP Password:
0
```

As you can see from the output of the command, the account has been removed from the AD.

Can Your IDM Play Chess?

NOTE	Command output below was recorded with OpenIDM 6.2.3. Paths refer to the bundled `samples/workflow` project.

Introduction

OpenIDM manages the lifecycle of identities in an organization. Automates onboarding, provisioning, privilege management, and offboarding processes. Can synchronize account changes across multiple enterprise systems.

In OpenIDM it is possible to configure workflows in the BPMN2 notation, as well as to create an arbitrary user interfaces with the needs of the organization.

But it would be boring to customize some typical workflow, moreover, OpenIDM capabilities allow you to do it as flexibly as possible. In this article we will consider the workflow of playing chess as an example.

Starting OpenIDM

From the Binary Package

Execute the following commands:

Get the latest version of OpenIDM

```
$ export VERSION=$(curl -i -o - --silent
https://api.github.com/repos/OpenIdentityPlatform/OpenIDM/releases/latest | grep -m1
"\"name\"" | cut -d\" -f4); echo "Last version: $VERSION"
Last version: 6.2.3
```

Download the OpenIDM binary package

```
$ curl -L https://github.com/OpenIdentityPlatform/OpenIDM/releases/download/
$VERSION/openidm-$VERSION.zip --output openidm.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 92.1M  100 92.1M    0     0  9421k      0  0:00:10  0:00:10 --:--:-- 9926k
```

Unzip the OpenIDM binary package

```
$ unzip openidm.zip
Archive:  openidm.zip
   creating: openidm/
```

```
....
```

Run OpenIDM

```
$ openidm/startup.sh -p samples/workflow
Executing openidm/startup.sh...
Using OPENIDM_HOME:   /tmp/openidm
Using PROJECT_HOME:   /tmp/openidm/samples/workflow
Using OPENIDM_OPTS:   -Dlogback.configurationFile=conf/logging-config.groovy
Using LOGGING_CONFIG: -Djava.util.logging.config.file=/tmp/openidm/samples/getting-
started/conf/logging.properties
Using boot properties at /tmp/openidm/samples/getting-
started/conf/boot/boot.properties
-> OpenIDM version "6.2.3" (revision: 284a4) 2024-11-11T09:18:27Z master
OpenIDM ready
```

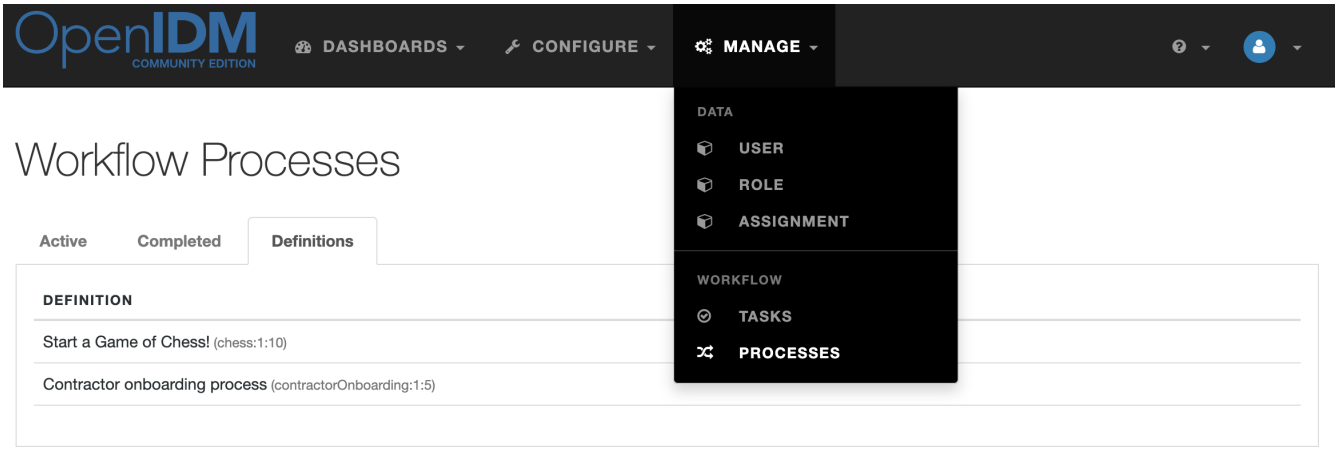
From the Docker Images

```
$ docker run -h idm-01.domain.com -p 8080:8080 -p 8443:8443 --name idm-01
openidentityplatform/openidm -p samples/workflow
Unable to find image 'openidentityplatform/openidm:latest' locally
latest: Pulling from openidentityplatform/openidm
74ac377868f8: Already exists
a182a611d05b: Already exists
e58ce1bd2f23: Already exists
e1b7fbdee987: Already exists
4f4fb700ef54: Already exists
26716adeef7f: Pull complete
Digest: sha256:6a6df88ca40116de4bba7ddef126a214feee04e7161b0d6f39ff9c9f448cda94
Status: Downloaded newer image for openidentityplatform/openidm:latest
Executing /opt/openidm/startup.sh...
Using OPENIDM_HOME:   /opt/openidm
Using PROJECT_HOME:   /opt/openidm/samples/workflow
Using OPENIDM_OPTS:   -server -XX:+UseContainerSupport -Dlogback.configurationFile
=conf/logging-config.groovy
Using LOGGING_CONFIG: -Djava.util.logging.config.file=/opt/openidm/samples/getting-
started/conf/logging.properties
Using boot properties at /opt/openidm/samples/getting-
started/conf/boot/boot.properties
ShellTUI: No standard input...exiting.
OpenIDM version "6.2.3" (revision: 284a4) 2024-11-11T09:18:27Z master
OpenIDM ready
```

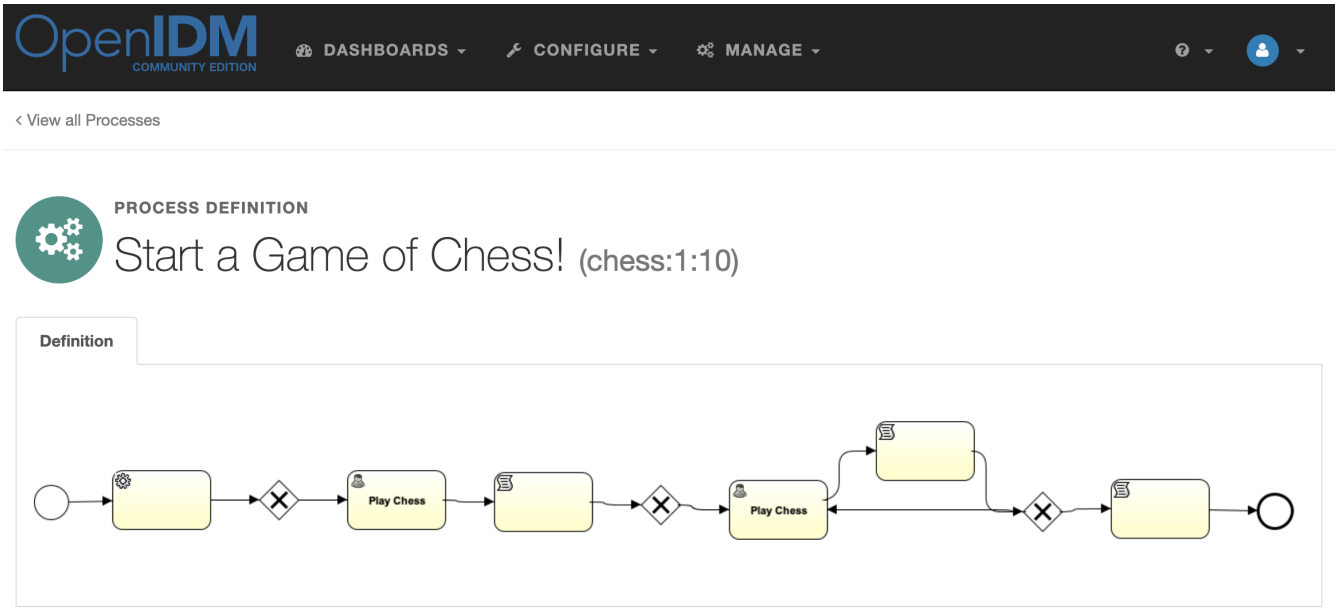
Initial OpenIDM Setup

Open the OpenIDM Administrator Console at http://localhost:8080/admin. Enter the administrator login and password. The default login is `openidm-admin`, password is `openidm-admin`.

In the Administrator Console go to Manage → Processes. Then go to the Definitions tab. And make sure the `Start a Game of Chess!` process is in the processes list.

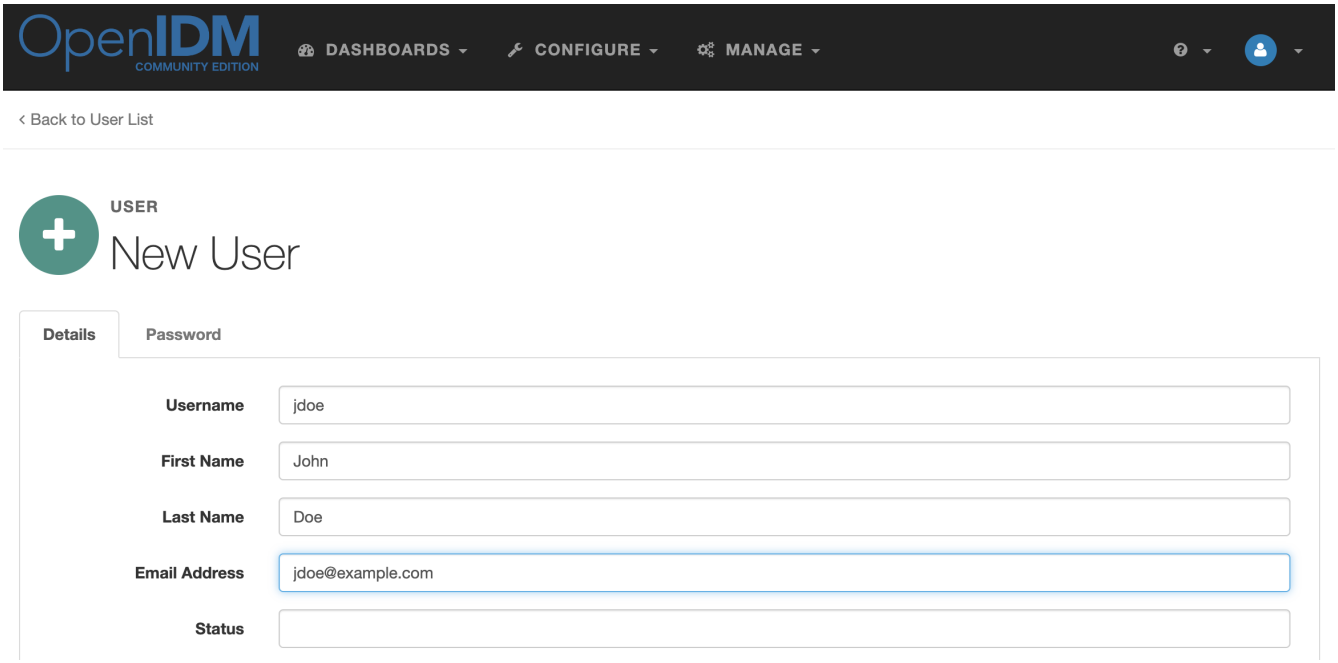


You can see what the workflow looks like by clicking on it.

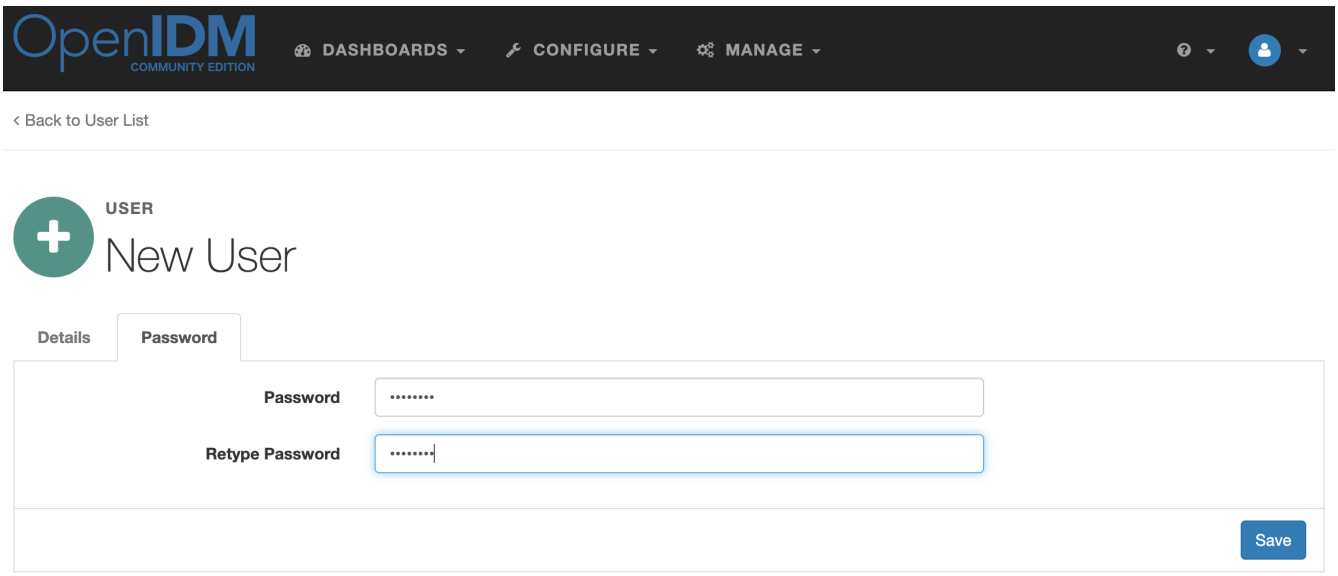


Now let's create a user for the administrator to play chess with.

In the Administrator Console in the main menu go to Manage → User. Create an account. Let it be an account with the login `jdoe`, for example.

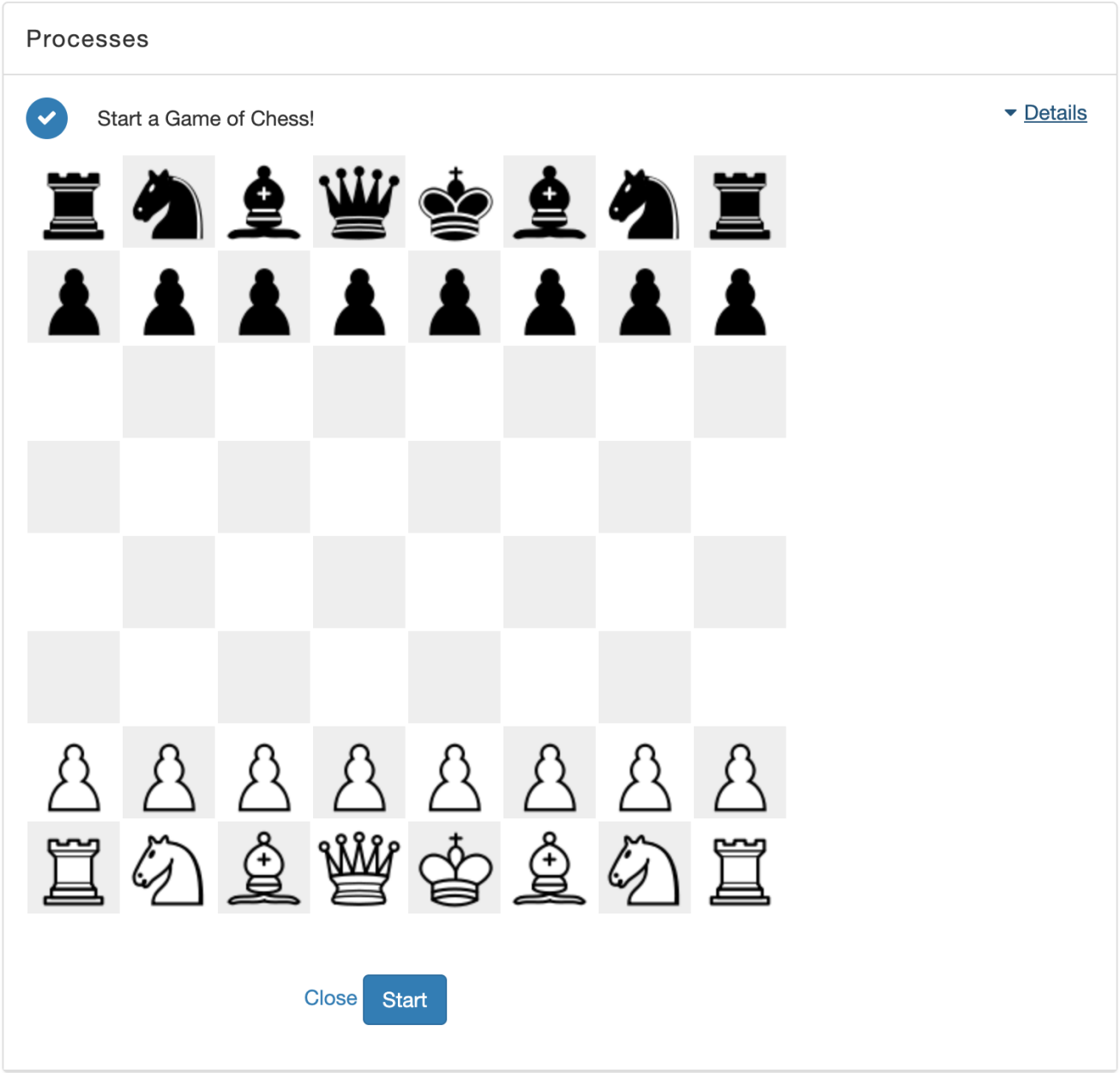


Click the Password tab and enter the password for the new user. Click the `Save` button.



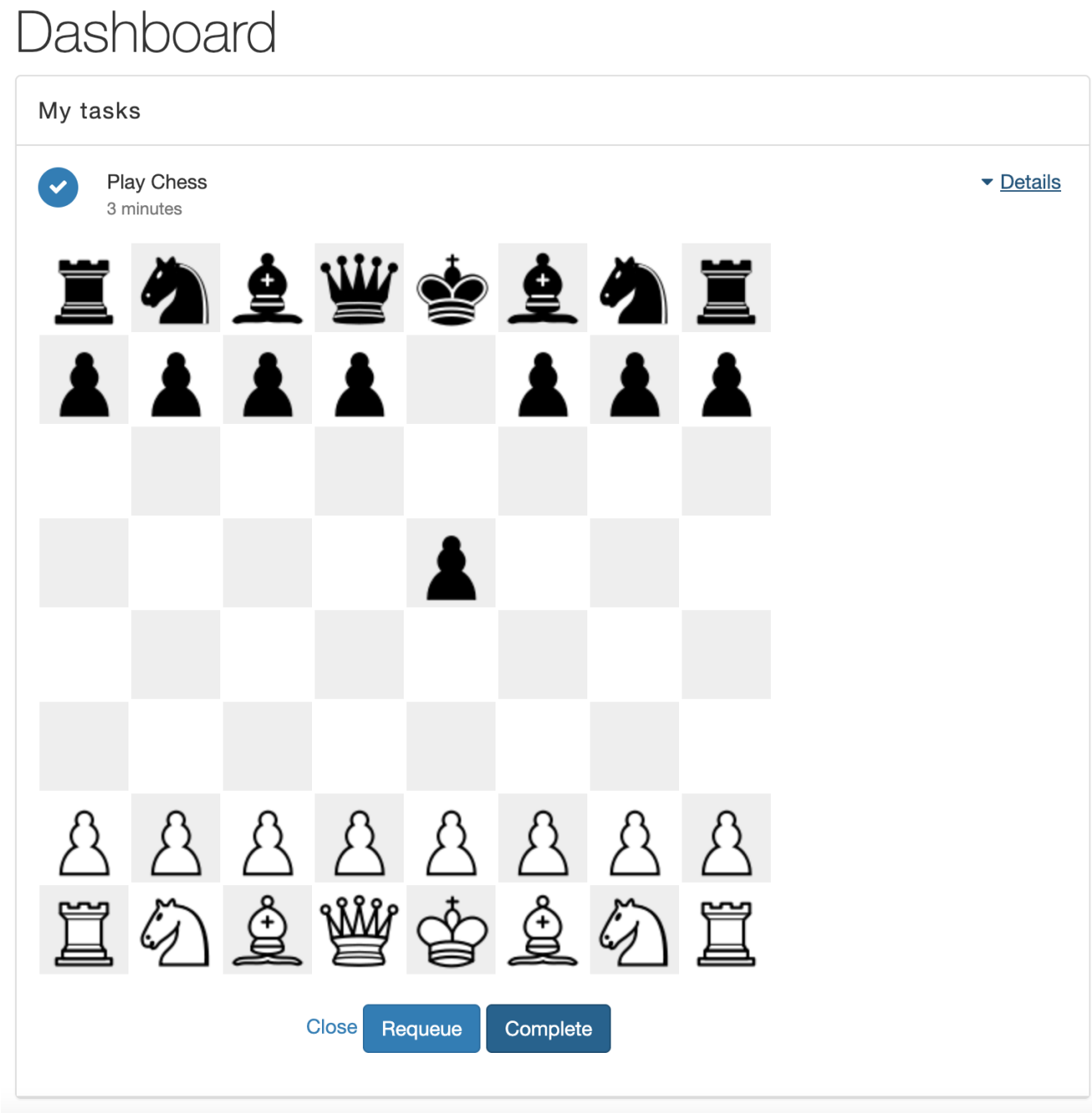
Playing Chess

Login to the OpenIDM console with the `openidm-admin` account at http://localhost:8080. On the Dashboard screen, under Processes, there will be an item `Start a Game of Chess!`. Click on the `Details` link A chess board will appear. Click the `Start` button. A new game will be created. You can assign the game to yourself and play with yourself or wait for one of the other users to accept your invitation to play.

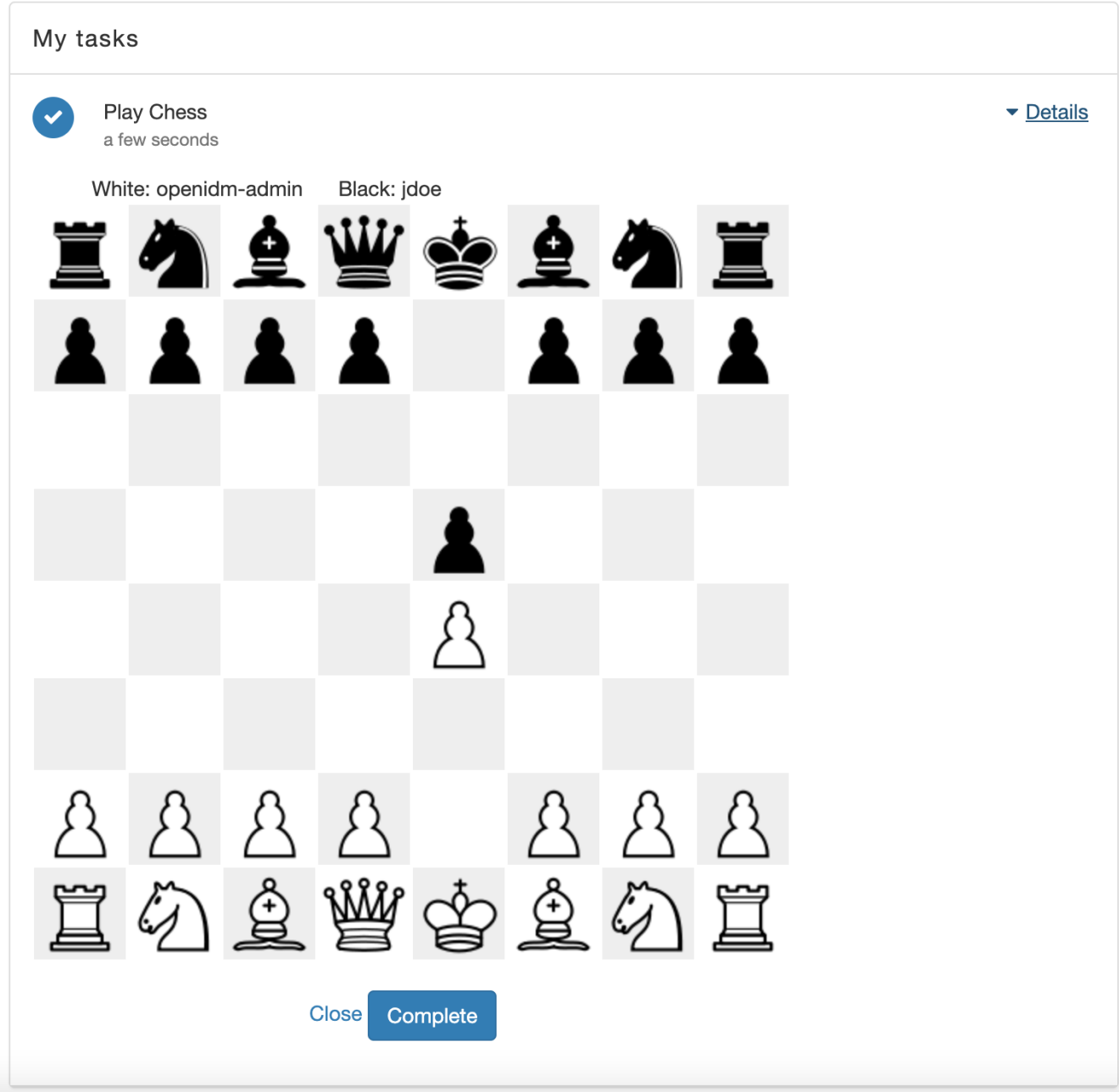


Open another browser or a new window in incognito mode and log in with the `jdoe` account.

On the Dashboard page, you will see an invitation to play. Assign the game to yourself, take the first turn, and press the `Complete` button.



The task will disappear from the list. Now go to the window with the authenticated user `openidm-admin` and refresh the task list. You will be prompted to make a return move. Make the move and click the `Complete` button. Now go again to the window with the user `jdoe`. A new task will appear for the user, prompting you to make your move. In this way, OpenIDM users can play chess among themselves.



Some Technical Details

Now let's understand how it works. Workflow in BPMN format is stored in the `samples/workflow/workflow` directory in the `chess.bar` archive You can view it using the command

```
$ unzip -p samples/workflow/workflow/chess.bar chess.bpmn20.xml | less
```

The user interface, the chessboard, is also stored in the same archive.

```
unzip -p samples/workflow/workflow/chess.bar chessboard.xhtml | less
```

You can create your workflow and user interface by analogy. Afterward, create a bar archive with the command

```
$ jar cvf chess.bar your-bpmn.xml your-ui.xhtml
```

And place it in the OpenIDM `workflow` directory.

You can read more about customizing workflows in the OpenIDM documentation here and here.

Configuring OpenIDM for Synchronization Between Active Directory and OpenDJ

Introduction

In this article we will configure two-way synchronization between Active Directory and OpenDJ. Changes made in Active Directory will be synchronized to OpenDJ and vice versa.

Configuring OpenIDM

A deployment of OpenIDM is described in the documentation. It is assumed that you have already deployed OpenIDM.

Configuring the data sources

Configuring Active Directory

Download the Active Directory provisioner.openicf-adldap.json Active Directory connection file from GitHub and copy it to the OpenIDM `conf` directory.

Modify the properties to match your Active Directory server settings:

Setting	Description
host	Hostname/IP address of the AD server
port	Connection port (default 389)
ssl	SSL is not used by default
principal	DN of the account connecting to AD, e.g. `"CN=Administrator,CN=Users,DC=example,DC=com"`.
credentials	Account password
baseContexts	A list of DNs containing accounts to synchronize, e.g. `["CN=Users,DC=Example,DC=com"].`
baseContextsToSynchronize	Identical to `baseContexts`
accountSearchFilter	A filter to search for accounts
accountSynchronizationFilter	Filter for account synchronization.

Configuring OpenDJ

If you do not have OpenDJ installed, install it as described in the documentation.

Download the Example.ldif test data file from GitHub

Perform the initial configuration of OpenDJ and import the data using the following command:

```
 cd /path/to/opendj

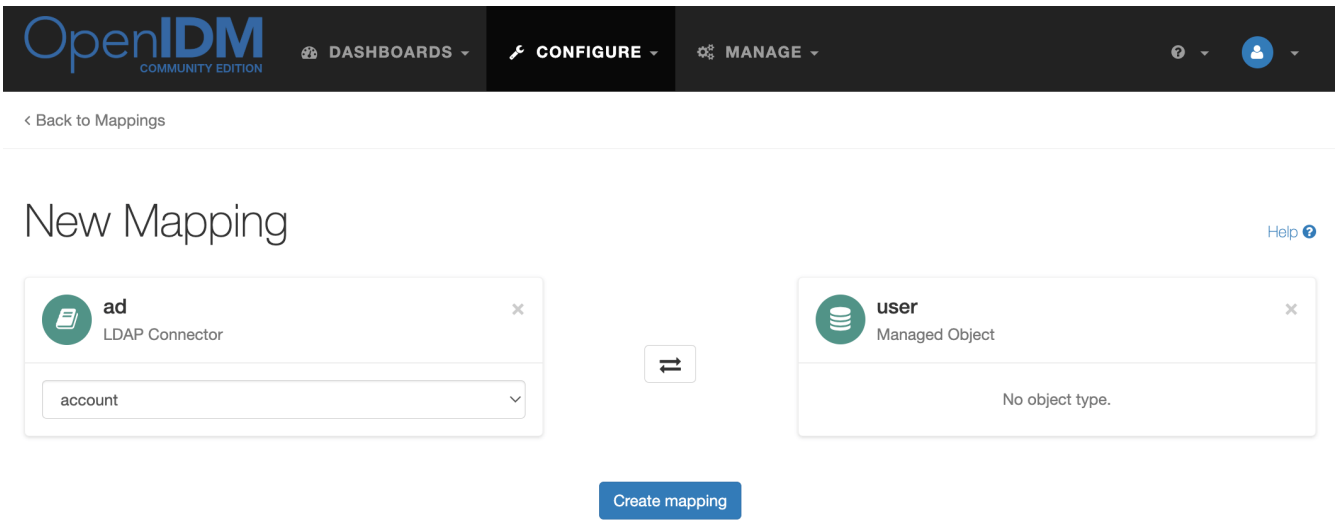
 ./setup --cli \
--hostname localhost \
--ldapPort 1389 \
--rootUserDN "cn=Directory Manager" \
--rootUserPassword password \
--adminConnectorPort 4444 \
--baseDN dc=com \
--ldifFile /path/to/Example.ldif \
--acceptLicense \
--no-prompt
...
Configuring Directory Server ..... Done.
Creating Base Entry dc=com ..... Done.
Starting Directory Server ....... Done.
...
```

Download the OpenDJ provisioner.openicf-ldap.json connection configuration file from GitHub and copy it to the OpenIDM `conf` directory.

The file can remain unchanged, as it is already configured for the default OpenDJ connection settings.

Configuring Active Directory → OpenDJ synchronization

Open the OpenIDM administrator console at http://localhost:8080/admin. Login with the username `openidm-admin` and password `openidm-admin`. In the top menu, navigate Configure → Mappings and create Mapping ad → user as shown in the figure below.



Open the created mapping **systemAdAccounts_managedUser** and on the Properties tab configure the field mappings as shown in the table below.

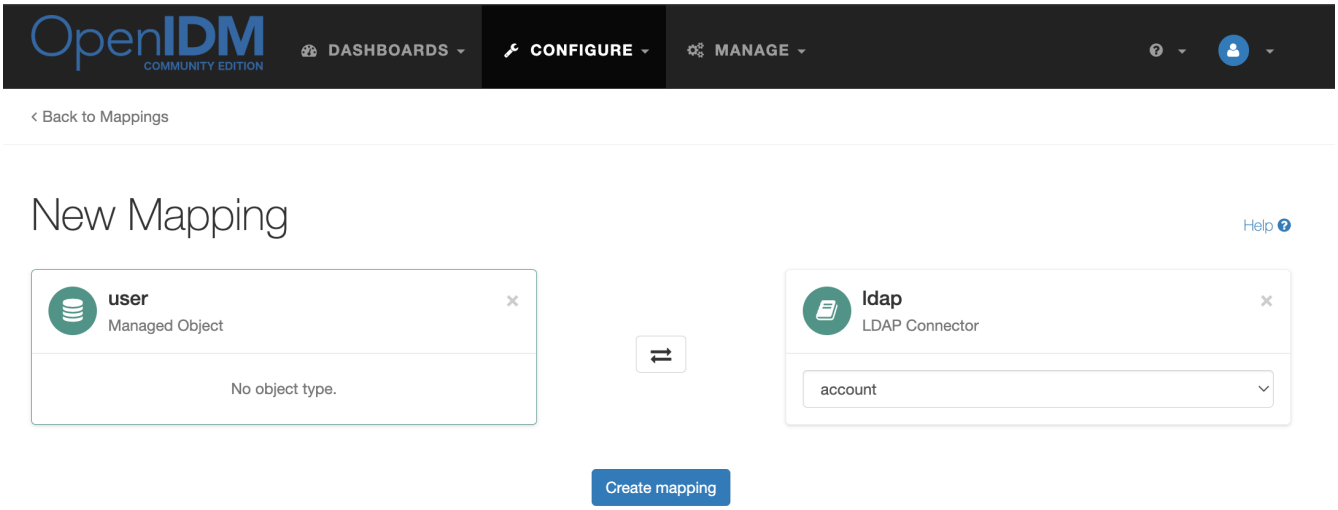
Source	Target
cn	cn
description	description
givenName	givenName
mail	mail
sn	sn
telephoneNumber	telephoneNumber
sAMAccountName	userName

On the Behaviors tab, configure the behavior for different synchronization situations.

Situation	Action
Ambiguous	Ignore
Source Missing	Delete
Missing	Ignore
Found Already Linked	Exception
Unqualified	Delete
Unassigned	Ignore
Link Only	Exception
Target Ignored	Ignore
Source Ignored	Ignored
All Gone	Ignore
Confirmed	Update
Found	Ignore
Absent	Create

Save the changes.

On the Mappings tab, create another mapping as shown in the figure below



Open the settings of the created mapping **managedUser_systemLdapAccounts** and on the Properties tab customize the field mappings as shown in the table below:

Source	Target	Transformation script	Conditional updates
userName	uid		
sn	sn		
	cn	`source.cn || (source.givenName + ' ' + source.sn)`	
givenName	givenName		
mail	mail		
description	description		
telephoneNumber	telephoneNumber		`object.telephoneNumber !== undefined && object.telephoneNumber !== null && object.telephoneNumber !== ''`

For the `description` field, specify the default value `Created in OpenIDM`

On the Behaviors tab, configure the behavior for different synchronization situations:

Situation	Action
Ambiguous	Ignore
Source Missing	Delete
Missing	Ignore
Found Already Linked	Exception
Unqualified	Delete
Unassigned	Ignore
Link Only	Exception

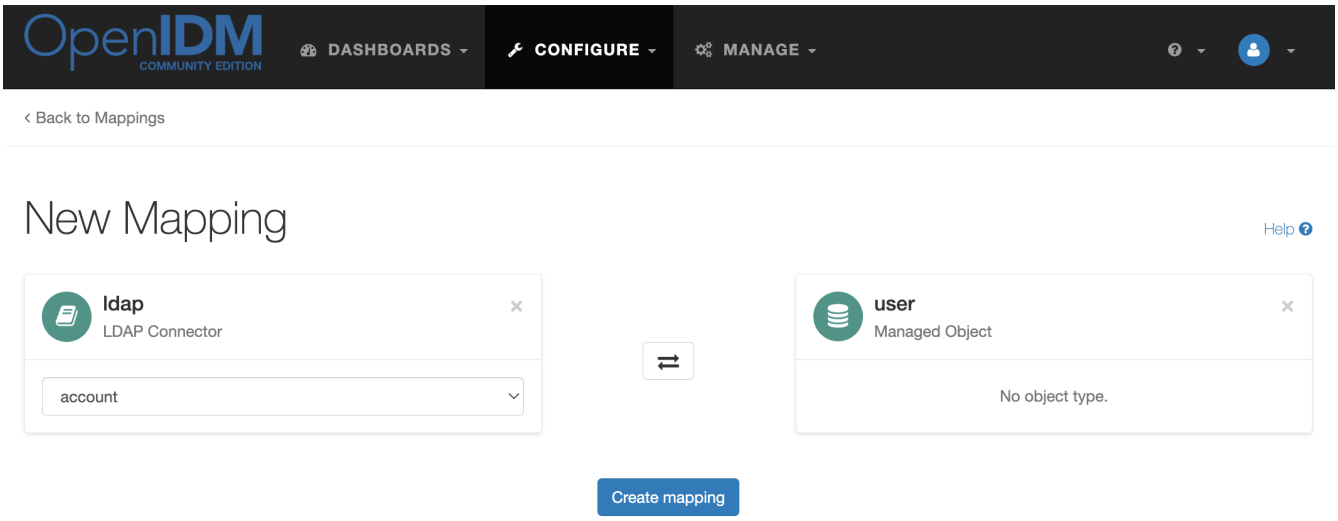
Situation	Action
Target Ignored	Ignore
Source Ignored	Ignore
All Gone	Ignore
Confirmed	Update
Found	Update
Absent	Create

On the same tab in the **Situational Event Scripts** section, add a script for the `onCreate` event.

```
target.dn = 'uid=' + source.userName + ',ou=People,dc=example,dc=com';
```

Configure OpenDJ → Active Directory synchronization

Create OpenDJ → Managed User synchronization



In the Linked Mapping field, select **managedUser_systemLdapAccounts.**

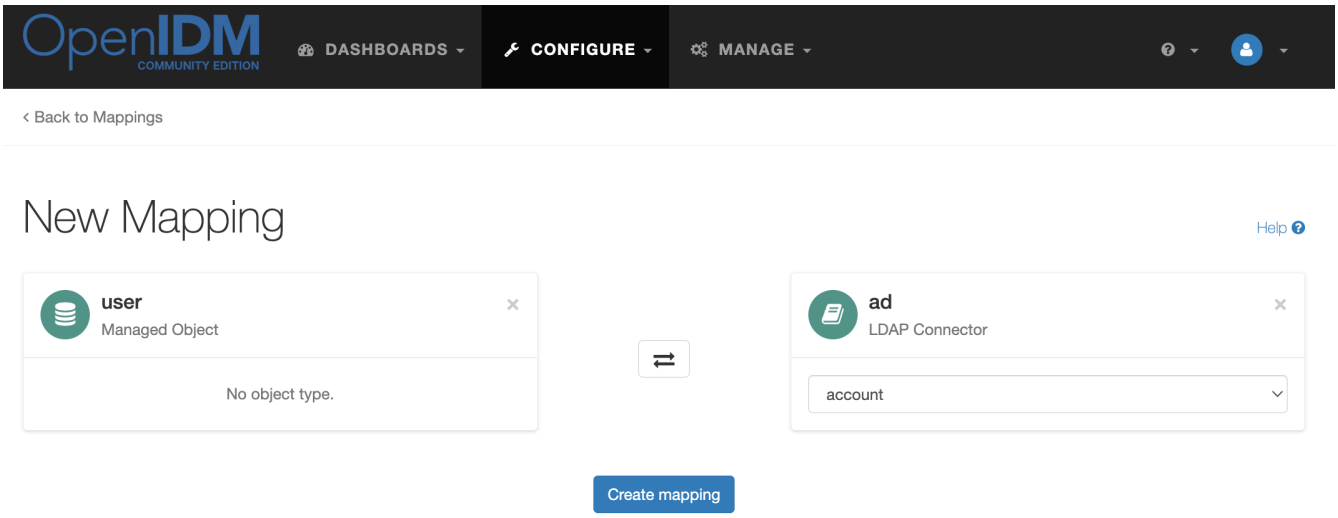
Open the created mapping **systemLdapAccount_managedUser** and on the Properties tab customize the field mappings as shown in the table.

Source	Target
mail	mail
sn	sn
givenName	givenName
uid	userName
telephoneNumber	telephoneNumber

On the Behaviors tab, configure the behavior.

Situation	Action
Ambiguous	Ignore
Source Missing	Delete
Missing	Ignore
Found Already Linked	Exception
Unqualified	Delete
Unassigned	Ignore
Link Only	Exception
Target Ignored	Ignore
Source Ignored	Ignore
All Gone	Ignore
Confirmed	Update
Found	Ignore
Absent	Create

Create mapping Managed User → Active Directory



In the Linked mapping field, select **systemAdAccounts_managedUser.**

In the created mapping **managedUser_systemAdAccounts** , configure the mapping:

Source	Target	Transformation Script
userName	dn	`'CN=' + source + ',CN=Users,DC=example,DC=org'`
givenName	givenName	
sn	sn	
	cn	`source.displayName || (source.givenName + ' ' + source.sn)`

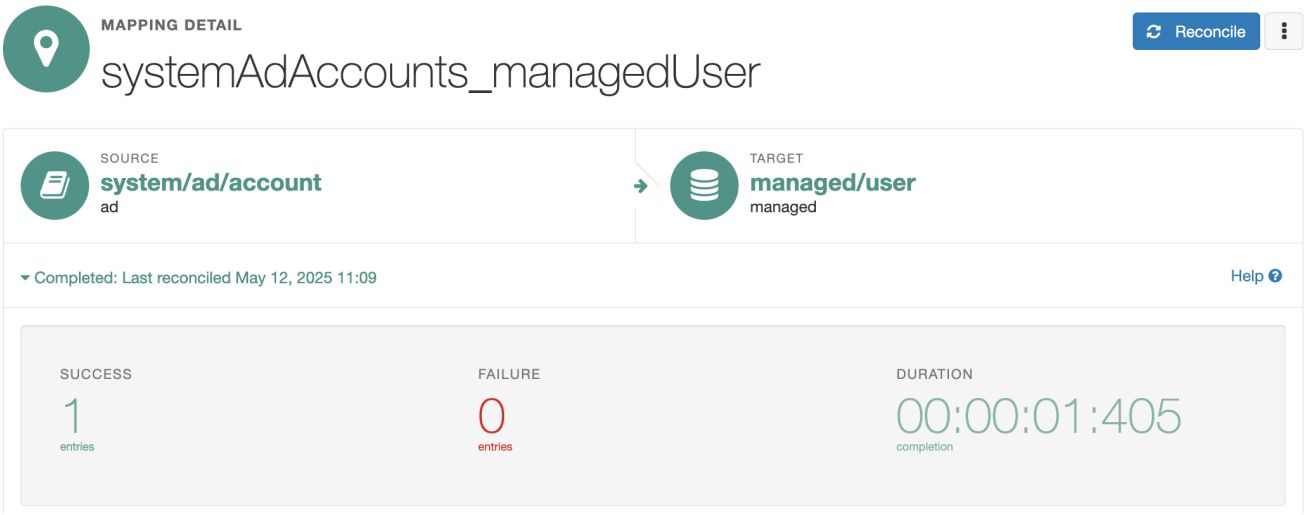
Source	Target	Transformation Script
description	description	
telephoneNumber	telephoneNumber	
userName	sAMAccountName	

On the Behaviors tab, configure the behavior similar to the step above.

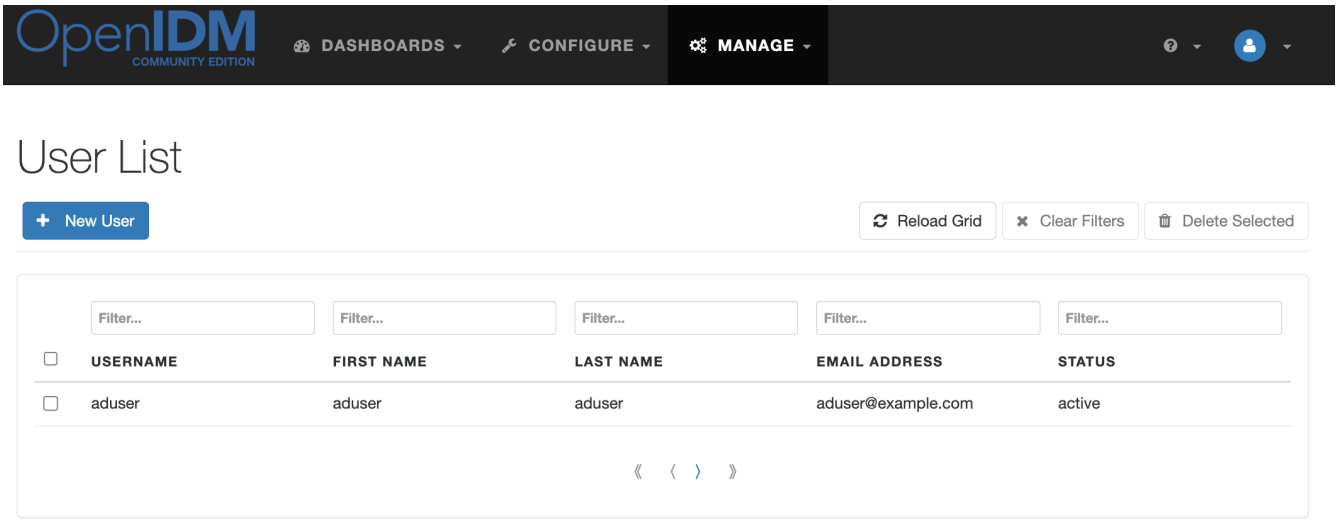
Verifying the solution

Active Directory Synchronization → OpenDJ

In the administrator console, select the **systemAdAccounts_managedUser** mapping and click Reconcile.



In the admin console, navigate to the Manage → User list. Accounts from Active Directory will appear in the user list



In the admin console, select the **managedUser_systemLdapAccounts** mapping and click Reconcile. After successful synchronization, the Active Directory records created in Managed Users will

appear in OpenDJ.

Check if the account exists with the command

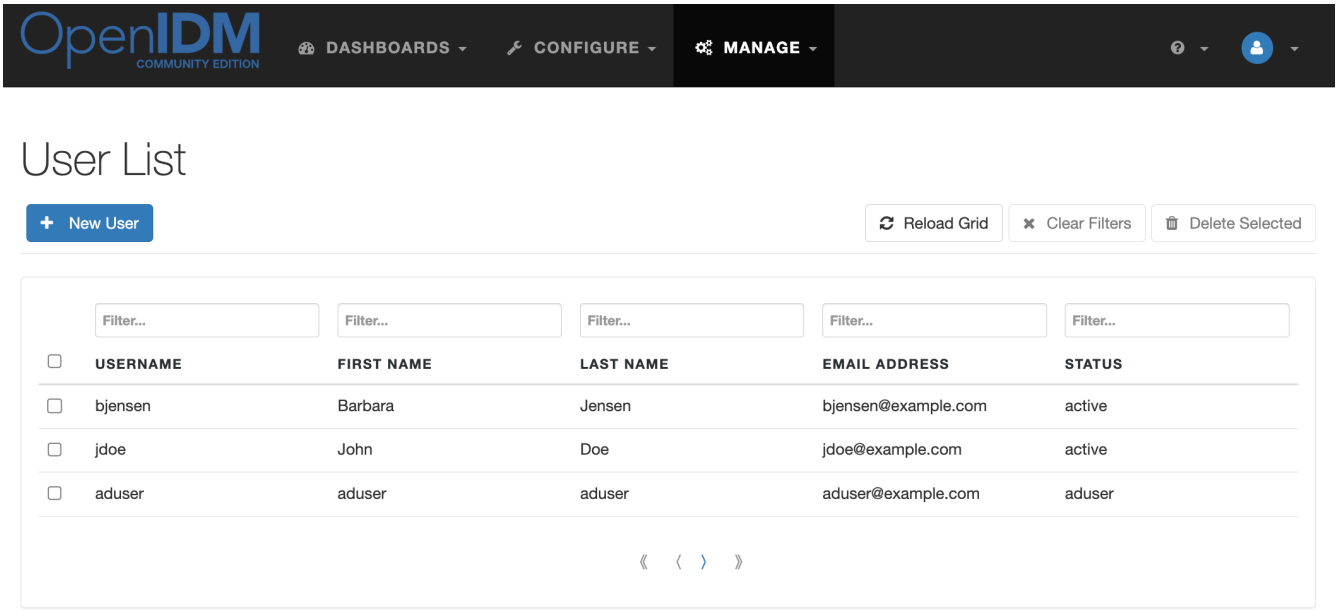
```
./opendj/bin/ldapsearch -p 1389 -b dc=example,dc=com  "(uid=aduser)" uid
dn: uid=aduser,ou=People,dc=example,dc=com
uid: aduser
```

OpenDJ Synchronization → Active Directory

In the admin console, under Configure → Mappings, select the **systemLdapAccount_managedUser** mapping and click Reconcile.

Navigate to Manage → User.

Accounts from OpenDJ will appear in the user list



Next, select the **managedUser_systemAdAccount** mapping and click Reconcile.

After successful synchronization, accounts from OpenDJ will appear in Active Directory.

Verify their existence with the following command

```
ldapsearch -H ldap://ad.example.com -x -W -D "admin@example.com" -b
"dc=example,dc=com" "(sAMAccountName=bjensen)" | grep dn
Enter LDAP Password:
dn: CN=bjensen,CN=Users,DC=example,DC=org
```

Configuring OpenIDM for Synchronization Between OpenLDAP and OpenDJ

NOTE	This example uses the Bitnami OpenLDAP image and its original configuration. Check image availability and compatibility before using it.

Introduction

This guide describes how to set up two-way synchronization between OpenLDAP and OpenDJ. Changes made in OpenLDAP will be reflected in OpenDJ, and vice versa.

OpenIDM Setup

OpenIDM deployment is described in documentation. It is assumed that OpenIDM is already installed and ready to be configured.

Data Sources Setup

OpenLDAP Setup

If you do not already have OpenLDAP installed, install it as described in documentation. Or you can use the Docker image. This is the approach we will use in this article.

Download the test data file openldap-data.ldif from GitHub and place it in the `./ldifs` directory that will be mounted into the OpenLDAP container.

```
docker run --name openldap \
 -p 2389:1389 -p 2636:1636 \
 --env LDAP_ADMIN_USERNAME=admin \
 --env LDAP_ADMIN_PASSWORD=adminpassword \
 --env LDAP_ROOT=dc=example,dc=com \
 --env LDAP_ADMIN_DN=cn=admin,dc=example,dc=com \
 -v ./ldifs:/ldifs \
 bitnami/openldap:latest
```

After OpenLDAP starts, verify that the data is loaded by running the command

```
ldapsearch -x -H ldap://localhost:2389 -D "cn=admin,dc=example,dc=com" \
 -w adminpassword -b "dc=example,dc=com" "(objectClass=inetOrgPerson)" | grep dn
dn: cn=user01,ou=users,dc=example,dc=com
dn: cn=user02,ou=users,dc=example,dc=com
```

Download the OpenLDAP connection configuration file provisioner.openicf-openldap.json from GitHub and copy it to the `conf` directory of OpenIDM.

The file is already configured according to the OpenLDAP connection settings in the Docker container and requires no changes.

OpenDJ Setup

If you do not have OpenDJ installed, install it as described in documentation.

Download the test data file Example.ldif from GitHub.

Perform the initial OpenDJ setup and import the data with the following command:

```
 cd /path/to/opendj

 ./setup --cli \
--hostname localhost \
--ldapPort 1389 \
--rootUserDN "cn=Directory Manager" \
--rootUserPassword password \
--adminConnectorPort 4444 \
--baseDN dc=com \
--ldifFile /path/to/Example.ldif \
--acceptLicense \
--no-prompt
...
Configuring Directory Server ..... Done.
Creating Base Entry dc=com ..... Done.
Starting Directory Server ....... Done.
...
```

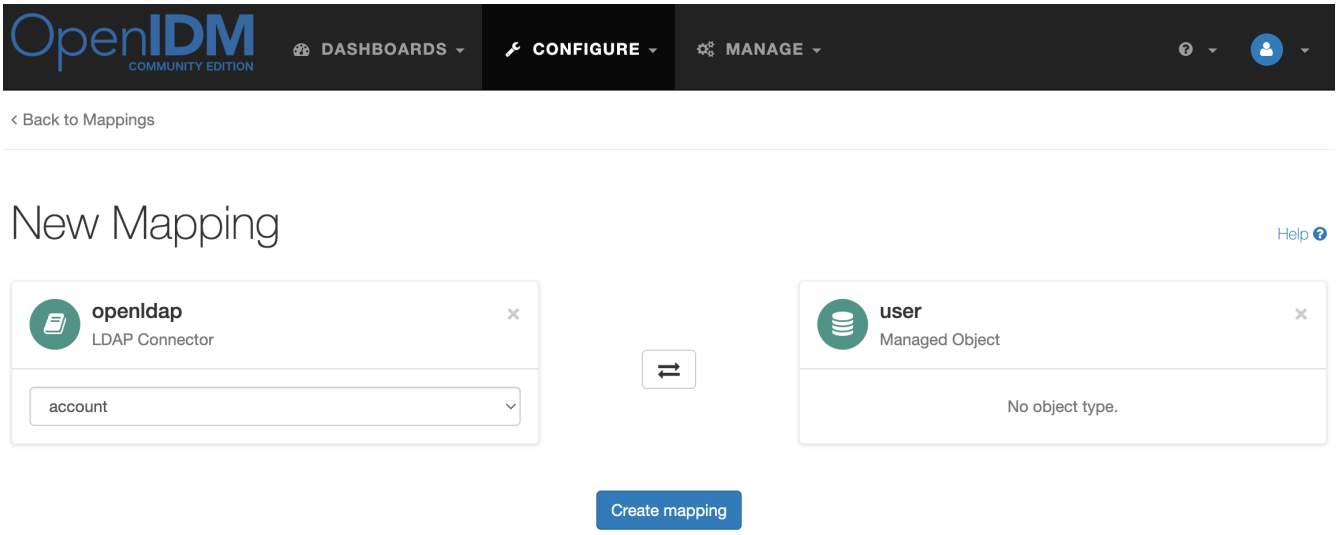
Download from GitHub the OpenDJ connection configuration file provisioner.openicf-ldap.json and copy it to the `conf` directory of OpenIDM.

The file can be left unchanged, it is already configured according to the default OpenDJ connection parameters.

Configuring OpenLDAP → OpenDJ Synchronization

Open the OpenIDM administrator console at http://localhost:8080/admin. Enter the value `openidm-admin` in the login and password fields.

In the top menu, open **Configure** → **Mappings** and create the mapping **openldap** → **user** as shown in the image below.



Open the created mapping **systemOpenldapAccount_managedUser** and on the **Properties** tab customize the field matches as shown in the table below

Source	**Target**
uid	userName
cn	cn
sn	sn
givenName	givenName
mail	mail
telephoneNumber	telephoneNumber

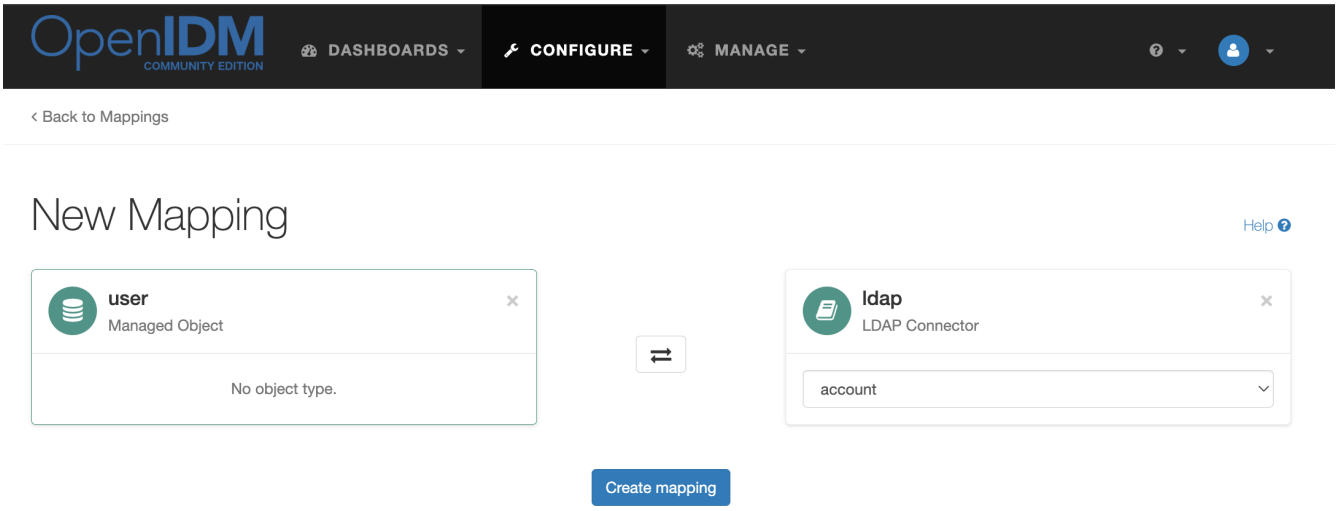
On the **Behaviors** tab, configure the behavior for different synchronization situations.

Situation	**Action**
Ambiguous	Ignore
Source Missing	Delete
Missing	Ignore
Found Already Linked	Exception
Unqualified	Delete
Unassigned	Ignore
Link Only	Exception
Target Ignored	Ignore
Source Ignored	Ignore
All Gone	Ignore
Confirmed	Update
Found	Ignore

Situation	Action
Absent	Create

Save the changes.

On the **Mappings** tab, create another mapping as shown in the figure below



Open the settings of the created mapping **managedUser_systemLdapAccounts** and on the **Properties** tab configure the field mappings as shown in the table below:

Source	Target	Transformation script
userName	uid	
sn	sn	
	cn	`source.cn || (source.givenName + ' ' + source.sn)`
givenName	givenName	
mail	mail	
description	description	
telephoneNumber	telephoneNumber	

For the `description` field, specify the default value `Created in OpenIDM`

On the **Behaviors** tab configure the behavior for different synchronization situations:

Situation	Action
Ambiguous	Ignore
Source Missing	Delete
Missing	Ignore
Found Already Linked	Exception
Unqualified	Delete

Situation	Action
Unassigned	Ignore
Link Only	Exception
Target Ignored	Ignore
Source Ignored	Ignore
All Gone	Ignore
Confirmed	Update
Found	Update
Absent	Create

On the same tab, under **Situational Event Scripts**, add a script for the onCreate event.

```
target.dn = 'uid=' + source.userName + ',ou=People,dc=example,dc=com';
```

Configuring OpenDJ → OpenLDAP Synchronization

Create synchronization **OpenDJ → Managed User**

In the **Linked Mapping** field, select **managedUser_systemLdapAccounts.**

Open the created mapping **systemLdapAccount_managedUser** and on the **Properties** tab customize the field mappings as shown in the tables below.

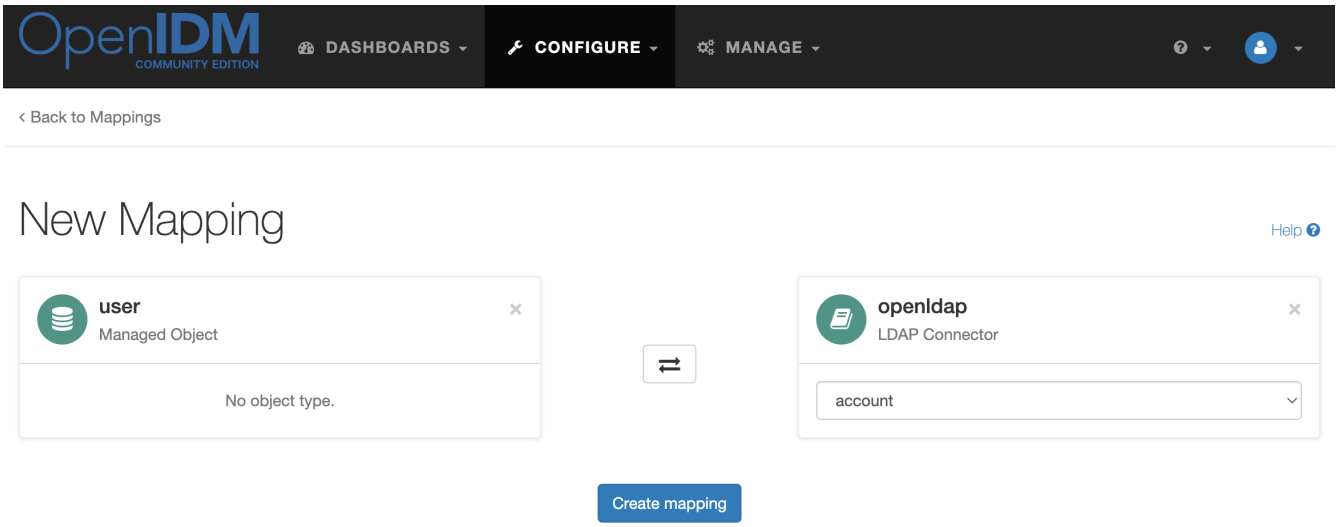
Source	Target
uid	userName
sn	sn
givenName	givenName
mail	mail
telephoneNumber	telephoneNumber

On the **Behaviors** tab, configure the behavior.

Situation	Action
Ambiguous	Ignore
Source Missing	Delete
Missing	Ignore
Found Already Linked	Exception
Unqualified	Delete
Unassigned	Ignore

Situation	Action
Link Only	Exception
Target Ignored	Ignore
Source Ignored	Ignore
All Gone	Ignore
Confirmed	Update
Found	Ignore
Absent	Create

Create a mapping **Managed User → OpenLDAP**



When creating the **Linked Mapping** setting, set it to **systemOpenldapAccount_managedUser**.

In the created mapping **managedUser_systemOpenldapAccount**, set it to match:

Source	Target	Transformation Script
userName	dn	`'cn=' + source + ',ou=users,dc=example,dc=com'`
givenName	givenName	
sn	sn	
userName	cn	
description	description	
telephoneNumber	telephoneNumber	
userName	uid	

Map `userName` directly to `cn` so the attribute keeps the RDN value used by `dn`. Replacing it with a display name or a given name and surname would remove that RDN value and cause updates of existing OpenLDAP entries to fail.

On the Behaviors tab, configure behavior similar to mapping

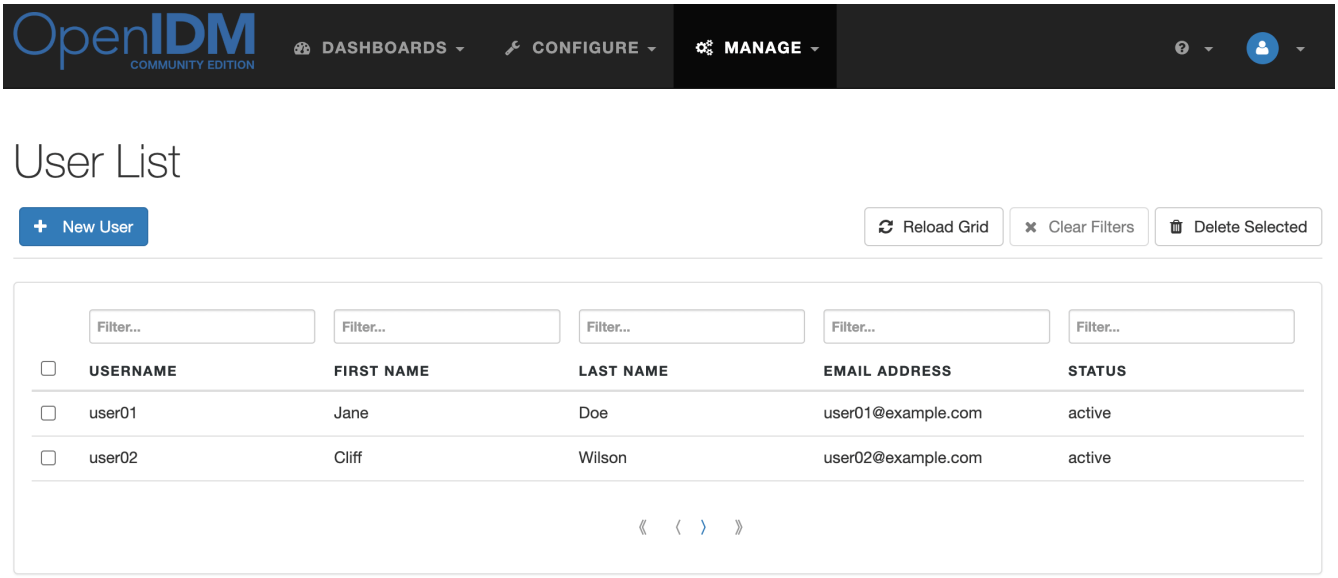
managedUser_systemLdapAccounts.

Test the Solution

OpenLDAP → OpenDJ Synchronization

In the admin console, select Mapping **systemOpenldapAccount_managedUser** and click Reconcile.

In the admin console, navigate to the Manage → User list. Accounts from OpenLDAP will appear in the user list



In the admin console, select Mapping **managedUser_systemLdapAccounts** and click Reconcile. After successful synchronization, the OpenLDAP entries created in Managed Users will appear in OpenDJ.

Check if the account exists with the command

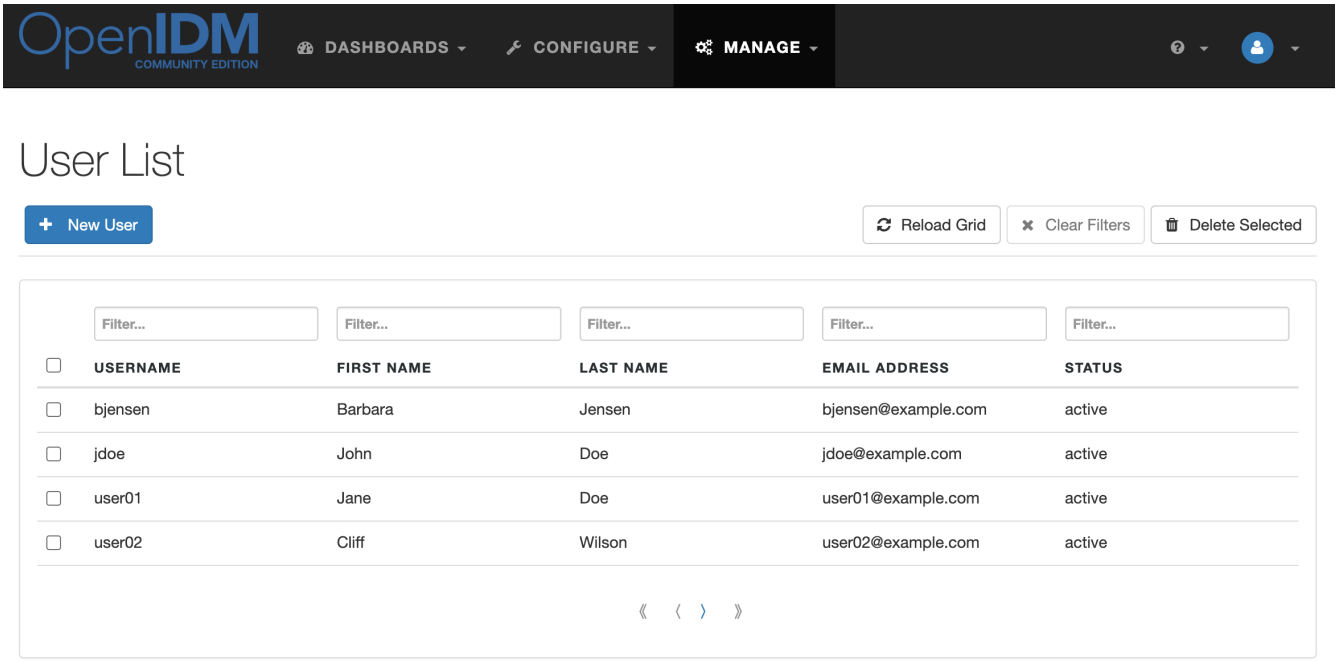
```
./opendj/bin/ldapsearch -p 1389 -b dc=example,dc=com  "(uid=user01)" uid
dn: uid=user01,ou=People,dc=example,dc=com
uid: user01
```

OpenDJ → OpenLDAP Synchronization

In the admin console, under **Configure** → **Mappings**, select the mapping **systemLdapAccount_managedUser.** And click Reconcile.

Go to **Manage** → **User**.

Accounts from OpenDJ will appear in the user list



Next, select mapping **managedUser_systemOpenldapAccount** and click Reconcile.

After successful synchronization, accounts from OpenDJ will appear in OpenLDAP.

Check for them with the command

```
ldapsearch -x -H ldap://localhost:2389 -D "cn=admin,dc=example,dc=com" \
 -w adminpassword -b "dc=example,dc=com" "(objectClass=inetOrgPerson)" | grep
cn=bjensen

dn: cn=bjensen,ou=users,dc=example,dc=com
```

Also verify the existing `user01` and `user02` entries. Reconcile **systemLdapAccount_managedUser** and then **managedUser_systemOpenldapAccount** again. Both runs should complete without naming violations. Run the following search after the repeated synchronization:

```
ldapsearch -x -H ldap://localhost:2389 -D "cn=admin,dc=example,dc=com" \
 -w adminpassword -b "dc=example,dc=com" \
 "(|(uid=user01)(uid=user02)(uid=bjensen))" cn | grep -E '^(dn|cn):'
```

The three entries must retain the `cn` value used in their RDN. The order of the results may differ:

```
dn: cn=user01,ou=users,dc=example,dc=com
cn: user01
dn: cn=user02,ou=users,dc=example,dc=com
cn: user02
dn: cn=bjensen,ou=users,dc=example,dc=com
cn: bjensen
```

How to Persist OpenIDM Configuration Data Between Restarts

If you need to persist OpenIDM configuration data between container restarts you need to mount the following directories

Directory	Description
`/opt/openidm/conf`	OpenIDM configuration files storage
`/opt/openidm/db`	OpenIDM database location in case you are using OrientDB (default DB)
`/opt/openidm/logs`	OpenIDM logs
`/opt/openidm/security`	OpenIDM keystore, trust store and realm configuration

But these directories except `/opt/openidm/logs` contains files, so you need to create a copy of pre-existing data:

Run the following commands:

```
mkdir -p openidm-data/logs
docker create --name tmp-openidm openidentityplatform/openidm
docker cp tmp-openidm:/opt/openidm/conf ./openidm-data/conf
docker cp tmp-openidm:/opt/openidm/db ./openidm-data/db
docker cp tmp-openidm:/opt/openidm/security ./openidm-data/security
docker rm tmp-openidm
sudo chown -R 1001:1001 ./openidm-data
```

Then you can mount the data.

With docker-compose.yaml file

```
services:
  openidm:
    image: openidentityplatform/openidm:latest
    volumes:
      - ./openidm-data/conf:/opt/openidm/conf
      - ./openidm-data/db:/opt/openidm/db
      - ./openidm-data/logs:/opt/openidm/logs
      - ./openidm-data/security:/opt/openidm/security
    ports:
      - "8080:8080"
```

With `docker run` command:

```
docker run -h idm-01.domain.com \
    -p 8080:8080 -p 8443:8443 \
    --name idm-01 \
    -v ./openidm-data/conf:/opt/openidm/conf \
    -v ./openidm-data/db:/opt/openidm/db \
    -v ./openidm-data/logs:/opt/openidm/logs \
    -v ./openidm-data/security:/opt/openidm/security \
    openidentityplatform/openidm
```