Multitenant User Management for SaaS

Basic Idea

This scenario simulates situation where multiple organizations - customers ("tenants") wish to manage their own users and administrators in midPoint and in target systems (represented by OpenLDAP accounts and groups).

The first administrative accounts will be created by importing (one-time or periodic using LiveSync) from customer management database (represented by CSV file) to midPoint. Passwords will be sent to the first administrative accounts (simulated by midPoint notifications redirected to a file). The administrators can then manage their own users. There are two categories of users in midPoint for this scenario:

- customer-admin (able to use GUI administrative tasks for his/her organization)
- customer-user (able to use GUI for self-service such as change password in OpenLDAP)

The value "customer-admin" or "customer-user" is stored in `employeeType` attribute of the user.

For each customer, new organization will be created in midPoint, where all users of this customer will be stored. For example, for two customers named "Ultra One Cloud Inc." and "Your Things Ours - Cloud Solutions Inc.", the structure could look like:

- **Customers**
    - *Ultra One Cloud Inc.*
        - ultraone-admin
        - ultraone1
        - u1user1
    - *Your Things Ours - Cloud Solutions Inc.*
        - yourthingsours-admin
        - yto1

For each customer, at least one administrator will be created, who can create any number of administrators and users. Each administrator can manage content only for his/her organization using delegated administration.

Also, the directory structure will be replicated in OpenLDAP. for example:

- dc=example,dc=com
    - ou=customers
        - ou=ultra1
            - cn=ultra1-admins (groupOfNames)
            - cn=ultra1-powerusers (groupOfNames)
            - cn=ultra1-users (groupOfNames)
            - uid=ultraone-admin
            - uid=ultraone1
            - uid=u1user1
        - ou=yourthingsours
            - cn=yourthingsours-admins (groupOfNames)
            - cn=yourthingsours-powerusers (groupOfNames)
            - cn=yourthingsours-users (groupOfNames)
            - uid=yourthingsours-admin
            - uid=yto1

The accounts will be put to corresponding groups according to the role assignments in midPoint. This means that you can create and manage the OpenLDAP organizational structure from midPoint, as well as create and manage the OpenLDAP accounts and groups.

Creating suborganizations in customer organizations is not yet supported in this scenario.

Scenario Details

The source for creating initial organizational structure and administrators is a CSV file with the following structure:

- name: e.g. ultraone-admin
- givenName: e.g. Perry
- familyName: e.g. Houser
- customerName: e.g. ultra1
- customerDisplayName: e.g. Ultra One Cloud Inc.
- customerContact: e.g. ultraone-admin@example.com
- disabled: e.g. false

Result of importing such record (line) from CSV will cause the following to happen in midPoint:

- organization with name/displayName ultra1/Ultra One Cloud Inc. will be created in midPoint
- user ultraone-admin ("Perry Houser") will be created in midPoint
- employeeType attribute will be set to "customer-admin"
- password will be generated for ultraone-admin and sent to email address specified as customerContact attribute value in CSV file (ultraone-admin@example.com)
- organization ultra1 will be assigned to ultraone-admin user as both member and manager
- role "Customer Admin Role" will be assigned to ultraone-admin user
- role "Customer User Role" will be assigned to ultraone-admin user
- role "Delegated Administration Role" will be assigned to ultraone-admin user to allow manage objects in ultraone-admin's organization (ultra1)

Also, provisioning to OpenLDAP will do the following:

- organizationalUnit ou=ultra1,ou=customers,dc=example,dc=com will be created to replicate midPoint organizational structure (Generic Synchronization)
- account uid=ultraone-admin,ou=ultra1,ou=customers,dc=example,dc=com will be created
    - account will be added to cn=ultra1-admins,ou=ultra1,ou=customers,dc=example,dc=com
    - account will be added to cn=ultra1-users,ou=ultra1,ou=customers,dc=example,dc=com

The initial password to OpenLDAP account will be the same as generated in midPoint and will be sent to ultraone-admin's e-mail address provided in the CSV file record. User ultraone-admin can login to midPoint and create more administrators and/or users for his/her organization using delegated administration authorizations provided by role "Delegated Administration Role".

The Files

OpenLDAP ACI	https://github.com/Evolveum/midpoint/blob/master/samples/stories/multitenant-idm-saas/aci.ldif	ACI for OpenLDAP user management for this sample. Update as you wish.
Sample CSV Source Data	https://github.com/Evolveum/midpoint/blob/master/samples/stories/multitenant-idm-saas/misc/midpoint-crm-flatfile.csv	CRM resource expects that this file is located in /var/tmp. Please update the filePath configuration property in the CRM resource.
System Configuration (fragment)	https://github.com/Evolveum/midpoint/blob/master/samples/stories/multitenant-idm-saas/misc/sysconfig-readme.txt	Notification configuration (redirected to file /usr/local/apache-tomcat-pokusy/logs/idm-mail-notifications.log) - change to match your system. Then edit System Configuration using Repository objects and replace notification configuration section with the content of this file.
Lookup Tables	https://github.com/Evolveum/midpoint/tree/master/samples/stories/multitenant-idm-saas/lookupTables	Configuration of employeeType lookup tables.
Password (Value) Policies	https://github.com/Evolveum/midpoint/tree/master/samples/stories/multitenant-idm-saas/valuePolicies	Sample password policies that can be assigned to organizations.
Organization Structure	https://github.com/Evolveum/midpoint/tree/master/samples/stories/multitenant-idm-saas/org	Organizational structure (root)
Resources	See below.	Configuration for source and target systems. Connection properties, schema handling and synchronization configuration.
Object Templates	https://github.com/Evolveum/midpoint/tree/master/samples/stories/multitenant-idm-saas/objectTemplates	Policies to apply for new/changed users and organizations
Roles	https://github.com/Evolveum/midpoint/tree/master/samples/stories/multitenant-idm-saas/roles	Basic roles for provisioning and delegated administration. Please do not import/use "role-meta-ldap-customer-group.xml", it's work in progress.

Resources

Resource	Type	Definition	Description
CRM Simulation	CSV	https://github.com/Evolveum/midpoint/blob/master/samples/stories/multitenant-idm-saas/resources/crm-simulation-sync.xml	Authoritative source.
OpenLDAP for Customers	LDAP	https://github.com/Evolveum/midpoint/blob/master/samples/stories/multitenant-idm-saas/resources/openldap-customers.xml	Target Resource

CRM Simulation

Authoritative source. It contains employee records, organizational structure and responsibilities. It contains:

Resource Objects	kind	intent	Description
customer records	account	default	Initial administrators along with organization information.

OpenLDAP for Customers

Target resource. Organizational structure allows separate containers for each customer accounts and groups.

Resource Objects	kind	intent	Description
user accounts	account	default	Accounts for customers (both admins and users)
LDAP groups	entitlement	group-org-admin	Groups created on demand for customer administrators
LDAP groups	entitlement	group-org-user	Groups created on demand for customer users
LDAP groups	entitlement	group-org-poweruser	Groups created on demand for customer power users
LDAP groups	entitlement	group-custom	(Not used yet, reserved for future scenario enhancements.)
LDAP groups	entitlement	ldapGroup	Any other groups.
OU	generic	ou-customer	Organizational units - created for each customer to contain accounts and groups

Setup

Before testing, import all the configuration from the files above:

1. organization structure objects
2. password policy objects
3. lookup tables
4. object templates
5. resources (change configuration properties if necessary, such as CSV file path, OpenLDAP hostname etc.)
6. roles (Please do not import/use "role-meta-ldap-customer-group.xml", it's work in progress.)

Go to Configuration - System and set the already imported object policies as global templates:

Object Type	Template	Notes
UserType	User Template	There is already global template "Default User Template". You will replace the reference with User Template using "Edit" button.
OrgType	Organization Object Template	There is no existing global template reference for organization, add new using "+" button.

Do not forget to save the System Configuration.

To setup notifications, please go to Configuration - Repository objects and click the System Configuration object to open in XML editor:

1. find the <notificationConfiguration> element and replace it with <notificationConfiguration> element content from "sysconfig-readme.txt" file mentioned above. The file contains also leading sentence "Paste this to System Configuration replacing existing <notificationConfiguration>" which should not be copied there.
2. before saving you can modify the <redirectToFile> path
3. Save the object when done.

You can also modify the path to file with simulated e-mails later by these steps:

1. go to Configuration - Notifications
2. update the "Redirect to file" value
3. Save.

Scenarios

The following sections describe scenarios prepared for this sample.

New customer administrator in CSV

This scenario is used for creating the first administrators. They will be able to create any number of users and administrators using GUI.

1. New administrator record is created in CSV resource, e.g.
    a. `name`: ultraone-admin
    b. `familyName`: Houser
    c. `givenName`: Perry
    d. `customerName`: ultra1
    e. `customerDisplayName`: Ultra One Cloud Inc.
    f. `customerContact`: ultraone-admin@example.com
    g. `disabled`: false
2. The record is either picked up using LiveSync task (if configured), or can be manually imported using single account import in Resources / CRM Simulation / Content and clicking on the wheel icon for selected user and choosing Import. Synchronization policy will be consulted and because the situation is unmatched, reaction addFocus causes creation of new user ultraone-admin in midPoint.
3. The account data are copied to the new user by inbound mappings on CRM resource. Note that `customerName` attribute from CRM account is copied to user's `organizationalUnit` property and `customerDisplayName` attribute from CRM account is copied to user's `organization` property. `employeeType` property will be set to "customer-admin" value.
4. User template takes over (`object-template-user.xml`)
    a. The first mapping computes user's full name (this information is not stored in CRM)
    b. The "Org mapping - organization member" is trying to look up an Org into which the user should belong. It is using a query inside assignmentTargetSearch expression to do so. The expression inside the query is using the `organizationalUnit` value, e.g. in this case it would be "ultra1".
    c. The query finds no matching Org. The expression is set to `createOnDemand` therefore it will try to create the Org. A new empty Org object is created in memory. Then the `populateItem` expressions are used to fill in this object. Please note how the `organizationalUnit` value is copied from the user to the `name` and `organization` value is copied from the user to the `displayName` of the new Org object.. Then midPoint calls itself internally to create a new Org object.
        i. Org object template (`object-template-org.xml`) takes over the processing of the new Org object.
        ii. The "Org-org mapping" in object template assigns the common parent "CUSTOMERS" to this new Org. This organization already exists, so no more recursive organizational structure will be required.
        iii. The Org object template has another mapping. This mapping assigns a meta-role (`metarole-org.xml`) to each created orgstruct. This meta-role contains `inducements` which specifies that a new `ou` and three groups should be created as a projections for each Org.
            1. The projections are computed for an LDAP resource. The projections have a form of LDAP organizationalUnit and groupOfNames objects. This is defined in the schema handling part of LDAP resource definition. Each inducement specifies an (kind, intent) tuple which is used to locate a matching definition in the schemaHandling.
            2. The outbound mappings are used to compute a correct DN for the new "ou" object in LDAP and all three "groupOfNames".
            3. LDAP organizationalUnit object is created.
            4. LDAP groups are created after organizationalUnit is created using resource dependencies.
        iv. Org object is now created in MidPoint.
        v. Note: the user is not yet assigned to this org structure. The user does not even exists yet. All of this was just a "side-effect" of the mapping in a user template. But now we are getting back to the user ...
    d. We are back in the processing of user template. We have processed first two mappings and we are going to process the rest of them.
    e. The "Org mapping - organization manager" will assign the same Org as above, but this time with relationship flag "manager" and only if midPoint user has "customer-admin" value in the `employeeType` attribute.
    f. The "Basic Customer Admin role assignment" mapping in user template is processed. This just assigns the "Customer Admin Role" role (`role-customer-basic-admin.xml`) . This is a simple RBAC role that assigns an LDAP account to the user. The role is assigned to midPoint users with `employeeType` attribute value "customer-admin".
    g. The "Delegated Administration role assignment" mapping in user template is processed. This just assigns the "Delegated Administration Role" role (`role-customer-authz-admin.xml`) . This role assigns midPoint authorizations only (no provisioning). The role is assigned to midPoint users with `employeeType` attribute value "customer-admin".
    h. The "Basic Customer User role assignment" mapping in user template is processed. This just assigns the "Customer User Role" role (`role-customer-basic-user.xml`) . This is a simple RBAC role that assigns an LDAP account to the user. The role is assigned to midPoint users with `employeeType` attribute values "customer-user" or "customer-admin".
    i. User template processing finishes. The user now has a full name and several assignments:
        i. Assignement to the "ultra1" Org.
        ii. Assignement of the "Customer User Role" role.
        iii. Assignment of the "Customer Admin Role" role.
        iv. Assignment of the "Delegated Administration Role" role.
5. The assignments are now computed.
    a. User ultraone-admin is placed in the "ultra1" Org.
    b. The "Customer User Role" assigns an LDAP account to the user and using entitlement association adds user to "ultra1-users" group. The group name is derived from user's `organizationalUnit` attribute value. No other attributes are specified in the role. The outbound mappings from the OpenLDAP resource definition are used to fill in account attribute values. Similarly, "Customer Admin Role" assigns user to "ultra1-admins" group.

        i. The outbound mappings are used especially to construct the DN of the account. The `organizationalUnit` value is used once again to do that. The DN is constructed in such a way that the account is placed into a correct organizationalUnit object.
6. Now we have everything to create the user in the repository and to create his LDAP account.
7. Synchronization reaction is finished. Everything returns to normal. The liveSync task (if configured) periodically checks for any new changes.

After first administrator is created for organization, all other tasks use delegated administration and midPoint GUI.

New customer administrator (GUI)

1. administrator logs in using midPoint GUI
2. administrator expands Org. structure menu entry and then clicks on Organization tree. Only organization managed by administrator will be displayed along with the users.
3. administrator clicks the wheel icon in the Members part and selected "Create member" action.
4. administrators fills in the following attributes:
    a. `name` (this will be the login of the new administrator)
    b. `givenName`
    c. `familyName`
    d. `employeeType`: click to the field and select "customer-admin" from the lookup table results. This will cause automatic role assignments in object template.
    e. `emailAddress`: type the new administrator e-mail address, it will be used for initial password notification. If no address is entered, the administrator who is creating the new user is expected to deliver the password.
    f. (organization does not need to be assigned, as "Create member" action will automatically assign the organization)
    g. Save the form
5. object template takes over (`object-template-user.xml`). The behaviour is almost the same as when creating new administrator from CSV file, but it will be simpler as the organization already exists in midPoint:
    a. The first mapping computes user's full name
    b. The "Org mapping - organization member" is trying to look up an Org into which the user should belong. It is using a query inside assignmentTargetSearch expression to do so. The expression inside the query is using the `organizationalUnit` value, e.g. in this case it would be "ultra1" and it was created before. The `organizationalUnit` attribute was set automatically by assigning the organization ("Add member").
    c. The "Org mapping - organization manager" will assign the same Org as above, but this time with relationship flag "manager" and only if midPoint user has "customer-admin" value in the `employeeType` attribute.
    d. The "Basic Customer Admin role assignment" mapping in user template is processed. This just assigns the "Customer Admin Role" role (`role-customer-basic-admin.xml`) . This is a simple RBAC role that assigns an LDAP account to the user. The role is assigned to midPoint users with `employeeType` attribute value "customer-admin".
    e. The "Delegated Administration role assignment" mapping in user template is processed. This just assigns the "Delegated Administration Role" role (`role-customer-authz-admin.xml`) . This role assigns midPoint authorizations only (no provisioning). The role is assigned to midPoint users with `employeeType` attribute value "customer-admin".
    f. The "Basic Customer User role assignment" mapping in user template is processed. This just assigns the "Customer User Role" role (`role-customer-basic-user.xml`) . This is a simple RBAC role that assigns an LDAP account to the user. The role is assigned to midPoint users with `employeeType` attribute values "customer-user" or "customer-admin".
    g. User template processing finishes. The user now has a full name and several assignments:
        i. Assignement to the "ultra1" Org. (because of "Add member" action in GUI)
        ii. Assignement of the "Customer User Role" role.
        iii. Assignment of the "Customer Admin Role" role.
        iv. Assignment of the "Delegated Administration Role" role.
6. The assignments are now computed.
    a. New administrator is placed in the "ultra1" Org.
    b. The "Customer User Role" assigns an LDAP account to the user and using entitlement association adds user to "ultra1-users" group. The group name is derived from user's `organizationalUnit` attribute value. No other attributes are specified in the role. The outbound mappings from the OpenLDAP resource definition are used to fill in account attribute values. Similarly, "Customer Admin Role" assigns user to "ultra1-admins" group.
        i. The outbound mappings are used especially to construct the DN of the account. The `organizationalUnit` value is used once again to do that. The DN is constructed in such a way that the account is placed into a correct organizationalUnit object.
7. Now we have everything to create the user in the repository and to create his LDAP account.
8. The new administrator is able to log in using midPoint GUI.

New customer user (GUI)

1. administrator logs in using midPoint GUI
2. administrator expands Org. structure menu entry and then clicks on Organization tree. Only organization managed by administrator will be displayed along with the users.
3. administrator clicks the wheel icon in the Members part and selects "Create member" action.
4. administrators fills in the following attributes:
    a. `name` (this will be the login of the new user)
    b. `givenName`
    c. `familyName`
    d. `employeeType`: click to the field and select "customer-user" from the lookup table results. This will cause automatic role assignments in object template.
    e. `emailAddress`: type the new user e-mail address, it will be used for initial password notification. If no address is entered, the administrator who is creating the new user is expected to deliver the password.
    f. (organization does not need to be assigned, as "Create member" action will automatically assign the organization)
    g. switch to "Assignments" tab
    h. click the wheel near "Assignments" title and select "Assign role"
    i. select "Customer End User Role" role and click "Assign"
    j. Save the form

5. object template takes over (`object-template-user.xml`). The behaviour is almost the same as when creating new administrator from CSV file, but it will be simpler as the organization already exists in midPoint:
    a. The first mapping computes user's full name
    b. The "Org mapping - organization member" is trying to look up an Org into which the user should belong. It is using a query inside assignmentTargetSearch expression to do so. The expression inside the query is using the `organizationalUnit` value, e.g. in this case it would be "ultra1" and it was created before. The `organizationalUnit` attribute was set automatically by assigning the organization ("Add member").
    c. The "Org mapping - organization manager" will not be applied this time as the user has "customer-user" (and not "customer-admin") value in the `employeeType` attribute.
    d. The "Basic Customer Admin role assignment" mapping will not be applied, as the user has "customer-user" (and not "customer-admin") e`mployeeType` attribute.
    e. The "Delegated Administration role assignment" mapping will not be applied, as the user has "customer-user" (and not "customer-admin") `employeeType` attribute.
    f. The "Basic Customer User role assignment" mapping in user template is processed. This just assigns the "Customer User Role" role (`role-customer-basic-user.xml`) . This is a simple RBAC role that assigns an LDAP account to the user. The role is assigned to midPoint users with `employeeType` attribute values "customer-user" or "customer-admin".
    g. User template processing finishes. The user now has a full name and two assignments:
        i. Assignement to the "ultra1" Org. (because of "Add member" action in GUI)
        ii. Assignement of the "Customer User Role" role.
6. The assignments are now computed.
    a. New user is placed in the "ultra1" Org.
    b. The "Customer User Role" assigns an LDAP account to the user and using entitlement association adds user to "ultra1-users" group. The group name is derived from user's `organizationalUnit` attribute value. No other attributes are specified in the role. The outbound mappings from the OpenLDAP resource definition are used to fill in account attribute values.
        i. The outbound mappings are used especially to construct the DN of the account. The `organizationalUnit` value is used once again to do that. The DN is constructed in such a way that the account is placed into a correct organizationalUnit object.
7. Now we have everything to create the user in the repository and to create his LDAP account
8. The new user is able to log in using midPoint GUI. Only self-service part will be accessible (and only if you assigned "Customer End User" role).

Editing user attributes (GUI)

1. administrator logs in using midPoint GUI
2. administrator expands Org. structure menu entry and then clicks on Organization tree. Only organization managed by administrator will be displayed along with the users.
3. administrator clicks the user which should be modified
4. administrators can change the following user attribute values (all other attributes are deliberately disabled from editing or will be computed, e.g. fullName or account DN):
    a. `name`
    b. `description`
    c. `givenName`
    d. `familyName`
    e. `employeeType` (to transition between "customer-admin" and "customer-user")
    f. `emailAddress`
    g. `activation`
    h. `password` (enter new password twice)
    i. Save the form
5. The above scenarios still apply, so object template will handle the transition between employeeType values.
6. Provisioning will ensure that account is updated in OpenLDAP.

Assigning more roles (GUI)

Users and administrators are created using pre-configured policies in object templates which assign the roles automatically based on `employeeType` attribute value. There is also a role "Customer Power User Role" which has no such behaviour and can be assigned/unassigned manually. This role is intended to simulate more advanced users which are not administrators but should have a special group assigned.

1. administrator logs in using midPoint GUI
2. administrator expands Org. structure menu entry and then clicks on Organization tree. Only organization managed by administrator will be displayed along with the users.
3. administrator clicks the user which should be modified and clicks on the tab "Assignments"
4. administrators clicks the wheel near "Assignments" and selects "Assign role" action
5. the list of assignable roles is deliberately restricted. Only "Customer Power User Role", "Customer Admin Role", "Customer User Role" and "Customer End User" can be assigned (along with standard "End user" role, which is superseded by "Customer End User Role" and should not be used). Select the checkboxes for the roles you wish to assign and then click "Assign" button:
    a. to make "power user" from normal user, assign "Customer Power User Role". This will cause the account to be added to the "cn=ultra1-powerusers" group in OpenLDAP
    b. to allow normal user to log in to midPoint and use self-service GUI, assign "End user" role.
    c. roles "Customer User Role" and "Customer Admin Role" are automatically (un)assigned based on `employeeType` attribute, but they can also be assigned manually (even if it does not make any sense).
    d. Save the form
6. Provisioning will ensure that account is updated in OpenLDAP.

Editing organization (GUI)

Administrator can modify organization attributes such as description of password policy for all users in the organization.

1. administrator logs in using midPoint GUI
2. administrator expands Org. structure menu entry and then clicks "Edit" on the organization in the Organization tree part of the page. Only organization managed by administrator will be displayed along with the users.
3. administrator clicks the organization which should be modified on the right side (not on the left side - this would display content (users) of that organization)
4. administrators clicks "Show empty fields" icon
5. attributes such as Description or Password Policy can be changed.
6. Save the form
7. Organization will be updated in midPoint and/or in OpenLDAP.

Changing password policy will influence all new passwords/password changes of the users in the organization. If no password policy is selected for organization, "Default Password Policy" is used.

Current Limitations

The scenario has currently some limitations:

- usernames must be globally unique (**MID-1629** - Getting issue details... STATUS , **MID-1977** - Getting issue details... STATUS). You cannot have multiple users "example01" even if they are in multiple tenants
- creating organizations in organizations (tenant) is not supported

See Also

- Configuration Samples