

Linux OpenLDAPおよび389-DSサーバを使用したUCS Manager & CIMCでのLDAPの設定

内容

[はじめに](#)

[バックグラウンド情報](#)

[前提条件](#)

[使用するコンポーネント](#)

[シナリオ1:Ubuntu - Debian](#)

[オプション1:Ubuntu LDAPアカウントマネージャ\(LAM\)を使用してOpenLDAPを設定する](#)

[ステップ1:Linuxサーバのホスト名とnet-toolsの初期設定。](#)

[手順2:SLAPD、Apache、PHPおよびその依存関係をインストールします。](#)

[手順3:LDAPアカウントマネージャをインストールする](#)

[ステップ4:LDAPアカウントマネージャの設定](#)

[手順5:OU、グループ、ユーザを作成します。](#)

[手順6：ローカルLDAPログインをテストします。](#)

[CIMCの設定パラメータ](#)

[UCS Managerの設定パラメータ](#)

[オプション2:Ubuntu CLIツールとオーバーレイを使用してOpenLDAPを設定する](#)

[手順1：初期net-toolsとLinuxサーバホスト名の設定](#)

[手順2:SLAPDをインストールする](#)

[ステップ3:LDAPサーバに「memberOf」オーバーレイをインストールします。](#)

[ステップ4:LDAPサーバに「refint」オーバーレイをインストールします。](#)

[ステップ5:OU、ユーザ、およびグループの作成](#)

[手順6：ローカルLDAPログインをテストします。](#)

[CIMCの設定パラメータ](#)

[UCS Managerの設定パラメータ](#)

[シナリオ2:CentOSストリーム10 - Fedora](#)

[オプション1:CentOSストリーム10で389ディレクトリサーバを使用してLDAPを設定する](#)

[ステップ1：初期設定](#)

[ステップ2:EPELリポジトリと389 Serverパッケージをインストールします。](#)

[ステップ3:LDAPグループとユーザの作成](#)

[ステップ4:memberOf overlayのインストール](#)

[CIMCの設定パラメータ](#)

[UCS Managerの設定パラメータ](#)

[結論](#)

はじめに

このドキュメントでは、LinuxベースのOpenLDAPおよび389ディレクトリサーバを使用してUCS ManagerおよびCIMCの認証方式としてLDAPを設定するさまざまなオプションについて説明します。

バックグラウンド情報

OpenLDAPサーバの設定は多岐にわたるため、このドキュメントでは包括的な取り扱いについては説明しません。この記事では、複数のLinuxディストリビューション、LDAPサーバパッケージ、および属性スキーマに広がる、一般的に実装される設定に重点を置いています。わかりやすくするために、このドキュメントでは標準的なLDAP設定を取り上げています。Secure LDAP(LDAPS)の設定は、このドキュメントでは説明しません。

前提条件

次の項目に関する知識を持っていることが強く推奨されます。

- UCS B シリーズ
- UCS C シリーズ
- Linuxサーバの管理

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- UCS Managerファームウェアバージョン : 4.3(2c)
- ファブリックインターコネクトモデル : UCS-FI-6454
- UCS Cシリーズスタンドアロンサーバモデル : UCSC-C240-M5
- UCS Cシリーズスタンドアロンファームウェアバージョン : 4.3(2.250045)
- Ubuntu 20.04
- CentOSストリーム10

このデモンストレーションで使用する設定 :

- LDAPサーバホスト名 : test
- サーバドメイン : xxxxxxxxxxx.com
- サーバFQDN:test.xxxxxxxxx.com
- Linuxサーバ (UbuntuおよびCentOS) のIPアドレス : X.X.X.19

- OpenLDAPユーザ : testuser1、testuser2
- OpenLDAPグループ : it
- OpenLDAPバインドユーザアカウント : bind_user

注 : この実習では、linux Nanoテキストエディタを使用しました。

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

シナリオ1:Ubuntu - Debian

LDAPサーバの設定は、管理上の設定および必要な制御レベルに応じて、LDAPアカウントマネージャなどのグラフィカルインターフェイスまたはコマンドラインツールを使用して実行できます。このシナリオでは、LinuxベースのOpenLDAPを使用した設定について検討します。まず、GUIベースの導入から始め、その後、コマンドラインユーティリティに移行して、オーバーレイプラグイン（Cisco UCS Managerとの統合でよく使用される）などの高度な機能を使用してみます。

オプション1:Ubuntu LDAPアカウントマネージャ(LAM)を使用してOpenLDAPを設定する

ステップ1:Linuxサーバのホスト名とnet-toolsの初期設定。

ubuntuを更新し、ifconfigやnetstatなどのツールにアクセスするためのnet-toolsパッケージをインストールします。

```
sudo apt update
sudo apt install net-tools
```

「ifconfig」コマンドを使用してサーバのIPアドレスを確認し、サーバのドメイン名（この実習で使用した「test.xxxxxxxxx.com」など）およびホスト名（たとえば「test」）とともに指定した形式で「/etc/hosts」ファイルに追加します。

```
sudo nano /etc/hosts
```

```
GNU nano 6.2 /etc/hosts
.19 test.xxxxxxxxx.com test
127.0.0.1 localhost
127.0.1.1 test

The following lines are desirable for IPv6 capable hosts
```

また、「/etc/hostname」ファイルの内容をhostname (test)に置き換えて、ファイルを更新します。

```
sudo nano /etc/hostname
```

```
GNU nano 6.2 /etc/hostname
test
```

これらの変更を有効にするには、サーバをリブートする必要があります。

```
sudo reboot
```

ステップ2:SLAPD、Apache、PHPおよびその依存関係をインストールする

次に、Apache、PHP、およびそれらの依存関係をインストールします。これらは、Webページ上でGUIインタラクションを有効にするために使用されます (GUIの場合)。

```
sudo apt install apache2 php php-cgi libapache2-mod-php php-mbstring php-common php-pear -y
```

Open LDAPサーバパッケージ「slapd」とその依存関係(ldap-utils)のインストール

```
sudo apt install slapd ldap-utils -y
```

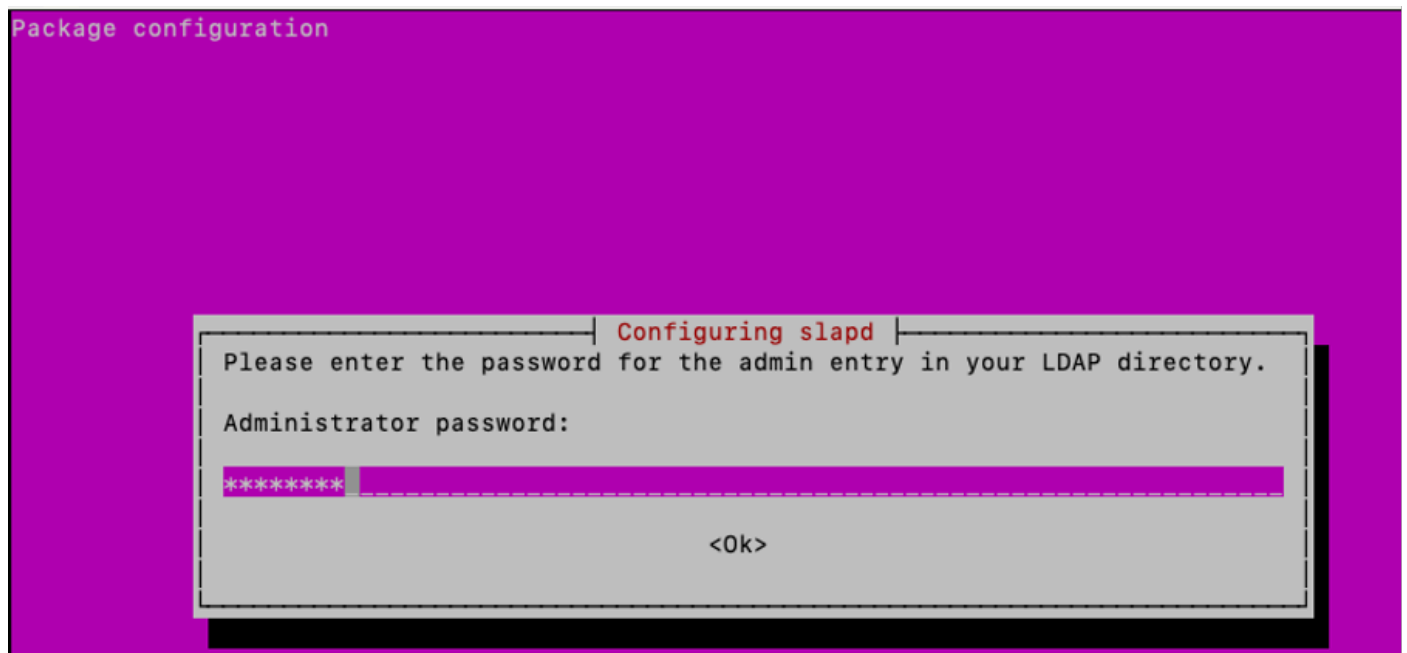
slapdのインストール中に、表示されたGUIポップアップで、必要な追加のSLAPDパッケージ設定を入力します。



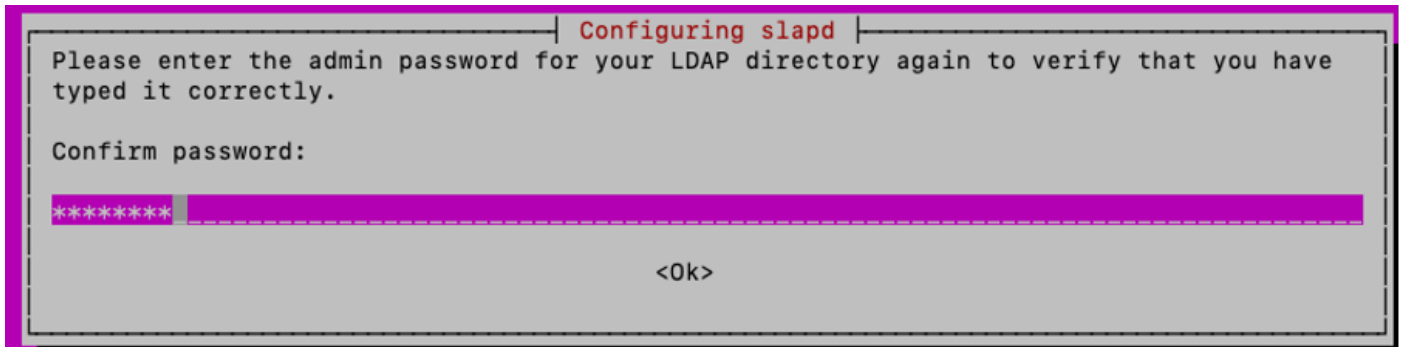
注：パスワードを失うと、LDAPサーバの再インストールが必要になります。

このコンテキストの「管理者」(admin)は、OpenLDAPサービス、モジュール、および設定の管理に使用されるアカウントです。

LDAPパッケージの「administrator」パスワードを追加し、キーボードでEnterキーを押して「OK」を選択します。



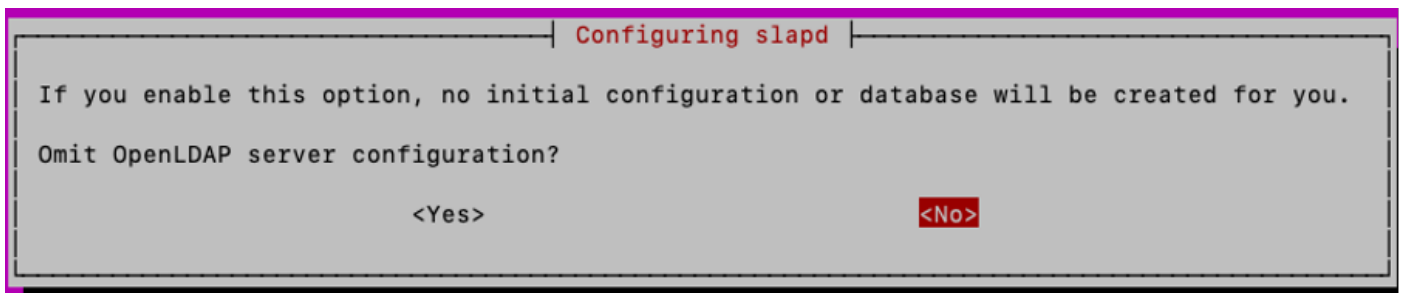
パスワードを確認します：



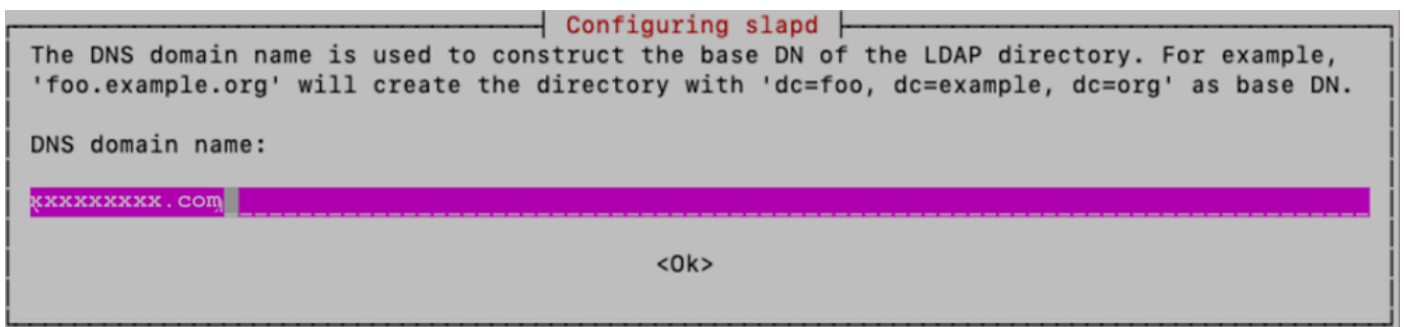
インストールが完了したら、指定したコマンドを使用してSLAPDパッケージを再構成し、ドメイン情報を追加できます。

```
sudo dpkg-reconfigure slapd
```

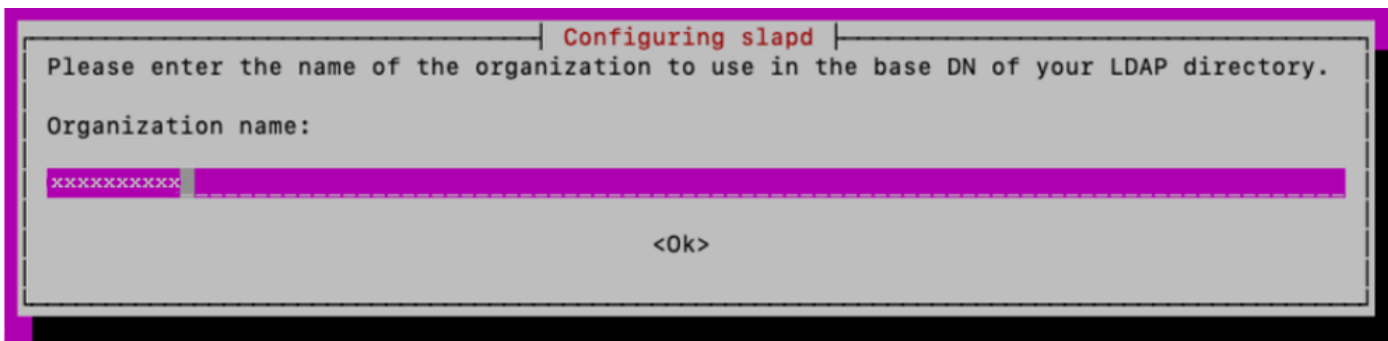
「Omit OpenLDAP server Configuration」のデフォルトの「No」オプションを受け入れて、Enterキーを押します。



ドメイン名を入力してEnterキーを押します。



このラボでは、「xxxxxxx」を「組織名」として使用します。



次に、「管理者パスワード」を入力して確認します

その他の設定オプションについては、デフォルトのままにして、キーボードのEnterキーを押して設定を完了します。

次のコマンドを使用して、SLAPDのインストールを確認します。

```
sudo slapcat
```

```
test@test:~$  
test@test:~$ sudo slapcat  
dn: dc=XXXXXXXXXX,dc=com  
objectClass: top  
objectClass: dcObject  
objectClass: organization  
o: XXXXXXXXXXXX  
dc: XXXXXXXXXXXX  
structuralObjectClass: organization  
entryUUID: 7baecf3e-c365-103f-8081-c70784fb9049  
creatorsName: cn=admin,dc=XXXXXXXXXX,dc=com  
createTimestamp: 20250512101324Z  
entryCSN: 20250512101324.193801Z#000000#000#000000  
modifiersName: cn=admin,dc=XXXXXXXXXX,dc=com  
modifyTimestamp: 20250512101324Z  
  
test@test:~$
```


ポート80(Web)、443 (セキュアWeb)、389(LDAP)、および636(必要に応じてセキュアLDAP)を許可するためのUbuntuファイアウォールの設定

```
sudo ufw enable  
sudo ufw allow 22
```

```
sudo ufw allow 80  
sudo ufw allow 443  
sudo ufw allow 389
```

```
sudo ufw allow 636
```

```
[test@test:~$ sudo ufw enable  
[Command may disrupt existing ssh connections. Proceed with operation (y|n)? y  
Firewall is active and enabled on system startup  
[test@test:~$ sudo ufw allow 22  
[[sudo] password for test:  
Rule added  
Rule added (v6)  
[test@test:~$ sudo ufw allow 80  
Rule added  
Rule added (v6)  
[test@test:~$ sudo ufw allow 443  
Rule added  
Rule added (v6)  
[test@test:~$ sudo ufw allow 389  
Rule added  
Rule added (v6)  
[test@test:~$ sudo ufw allow 636  
Rule added  
Rule added (v6)  
test@test:~$ █
```

Ubuntu Firewallのステータスを確認します。

```
sudo ufw status
```

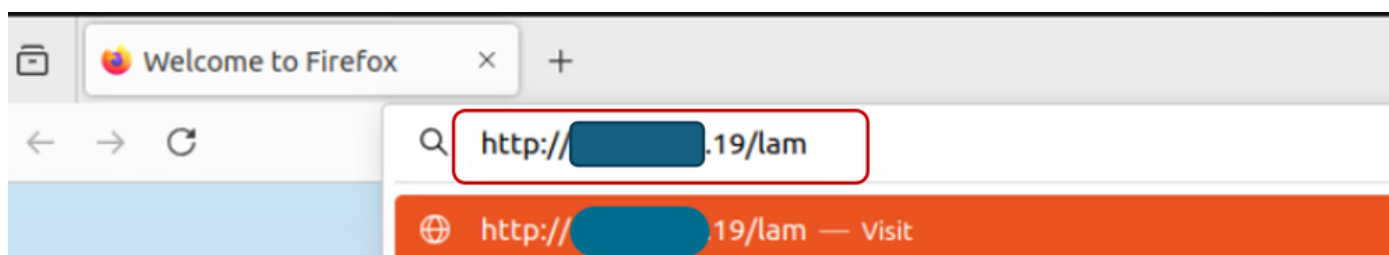
```
[test@test:~$ sudo ufw status  
Status: active
```

To	Action	From
22	ALLOW	Anywhere
80	ALLOW	Anywhere
443	ALLOW	Anywhere
389	ALLOW	Anywhere
636	ALLOW	Anywhere
22 (v6)	ALLOW	Anywhere (v6)
80 (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)
389 (v6)	ALLOW	Anywhere (v6)
636 (v6)	ALLOW	Anywhere (v6)

手順4:LDAPアカウントマネージャを設定します。

GUIからLDAPアカウントマネージャ(LAM)を設定するには、Webブラウザを開き、LinuxサーバのIPアドレスを入力し、次に示すように「lam」パスを追加します。

<http://X.X.X.19/lam>



「LAM設定」をクリックし、「サーバプロファイルの編集」を選択します。

LAM Login

User name

Password

Language

Login

LDAP server ldap://localhost:389
Server profile lam




Edit general settings



Edit server profiles



Import and export configuration

 [Back to login](#)

デフォルトのlamパスワード「lam」を入力してログインします。

Please enter your password to change the server preferences:

Profile name lam


Password

Ok

Manage server profiles

General Settingsタブで、Server settingsの「Language」と「Timezone」を確認します。

ツール設定セクションで、次に示すように、ツリーのサフィックスフィールドに必要なドメイン名を編集して追加します。

 Tool settings


Hidden tools

PDF editor	<input type="checkbox"/>	LDAP import/export	<input type="checkbox"/>	Tree view	<input type="checkbox"/>
Schema browser	<input type="checkbox"/>	WebAuthn devices	<input type="checkbox"/>	OU editor	<input type="checkbox"/>
Profile editor	<input type="checkbox"/>	Multi edit	<input type="checkbox"/>	Server information	<input type="checkbox"/>
File upload	<input type="checkbox"/>	Tests	<input type="checkbox"/>		

Tree view

Tree suffix

「セキュリティ設定」セクションを編集して、SLAPDサービスの管理に使用する「admin」ユーザを追加します。

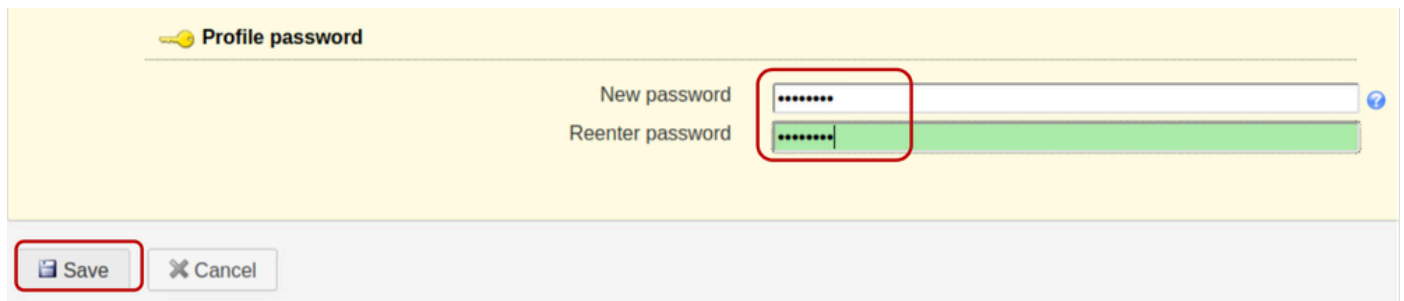
 Security settings

Login method Fixed list

List of valid users *

「プロファイルパスワード」を設定します。このパスワードは、LAM設定インターフェイスへの以降のログインに使用されます。この例では、デフォルトの「lam」パスワードの代わりに「cisco123」が設定されています。

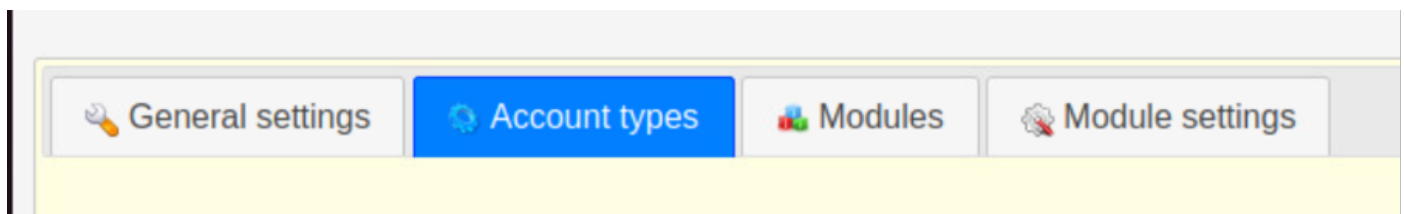
設定の保存:



その後、LAM設定GUIインターフェイスでセッションが再開されます。

作成した新しいパスワードを使用して再度ログインします (LAM設定>>サーバプロファイルの編集)。

「アカウントタイプ」をクリックします



下にスクロールし、LDAPサフィックスフィールドのドメイン名情報を使用して、デフォルトのアクティブアカウントタイプを編集します。たとえば、「LDAPサフィックス」フィールドのデフォルトの内容には、「ou=People,dc=my-domain,dc=com」という値が表示されます。

新しい組織ユニットを作成する必要がある場合は、「LDAPサフィックス」フィールドの内容を組織ユニットの名前に置き換えます。

形式は、「ou=<organizational_unit>,dc=xxxxxxxxx,dc=com」です。

このデモンストレーションでは、ユーザのOUは「People」、グループのOUは「Groups」です。

設定を保存します。

Active account types

Users User accounts (e.g. Unix, Samba and Kolab) ⬇️ ✖️

LDAP suffix ?

List attributes ?

Custom label ?

Additional LDAP filter ?

Hidden ?

Groups Group accounts (e.g. Unix and Samba) ⬆️ ✖️

LDAP suffix ?

List attributes ?

Custom label ?

Additional LDAP filter ?

Hidden ?

Optionsセクションまでスクロールし、Set primary group as memberUidにチェックマークが付いていることを確認します。

デフォルトでは、「Set primary group as memberUid」オプションはグループ・オブジェクトに設定されていません。これを有効にすると、「memberUid」を参照できる標準のLDAPグループのようにOpenLDAPの「プライマリグループ」を使用できます（例：UCS Cシリーズサーバ設定で）。このオプションをオフにすると、プライマリグループに属するユーザのログインが失敗します。

設定を保存します。

Options

Password hash type: SSHA

Login shells: /bin/dash, /bin/false, /bin/ksh, /bin/sh

Set primary group as memberUid

Unix

Groups

GID generator: Fixed range

Minimum GID number *: 10000

Maximum GID number *: 20000

Suffix for GID/group name check:

Disable membership management

Save Cancel

手順5:OU、グループ、およびユーザを作成します。

インストール時に作成したのと同じパスワードを使用して「admin」ユーザとしてLAMにログインし、以前作成したOU(PeopleおよびGroups)に属するUsersおよびGroupsをそれぞれ作成します。

LAM Login

User name admin

Password

Language English (Great Britain)

Login

LDAP server ldap://localhost:389

Server profile lam

LAM設定セクションで、先に指定したOUを作成します。
Createをクリックします。

Users Groups

The following suffixes are missing in LDAP. LAM can create them for you.
You can setup the LDAP suffixes for all account types in your LAM server profile on tab "Account types".

ou=People,dc=xxxxxxxx,dc=com
ou=Groups,dc=xxxxxxxx,dc=com

Create Cancel

次に、LDAPアカウントマネージャで「it」グループを作成します。

Groupsタブを選択し、New groupをクリックします

Users Groups

New group File upload

Group count: 0

Actions	Group name	GID number	Group
Sort sequence	▼▲	▼▲	▼▲
<input type="checkbox"/> Filter	<input type="text"/>	<input type="text"/>	<input type="text"/>

グループ名を「it」に設定します。



注: Cisco UCSシステムは一般的に大文字と小文字の区別に対して回復力がありますが、小文字の命名規則を維持することは、さまざまなLDAPサーバインフラストラクチャ環境で長期の相互運用性を確保するためのベストプラクティスです。

[GID番号]フィールドは空白のままにします。LDAPアカウントマネージャ(LAM)は、このフィールドに次に使用可能な値を自動的に入力するように設計されています。

必要に応じて説明を入力し、「保存」をクリックします。

Users Groups

Save Set password default Load profile

New group

Suffix Groups > xxxxxxxx > com RDN identifier cn

Unix

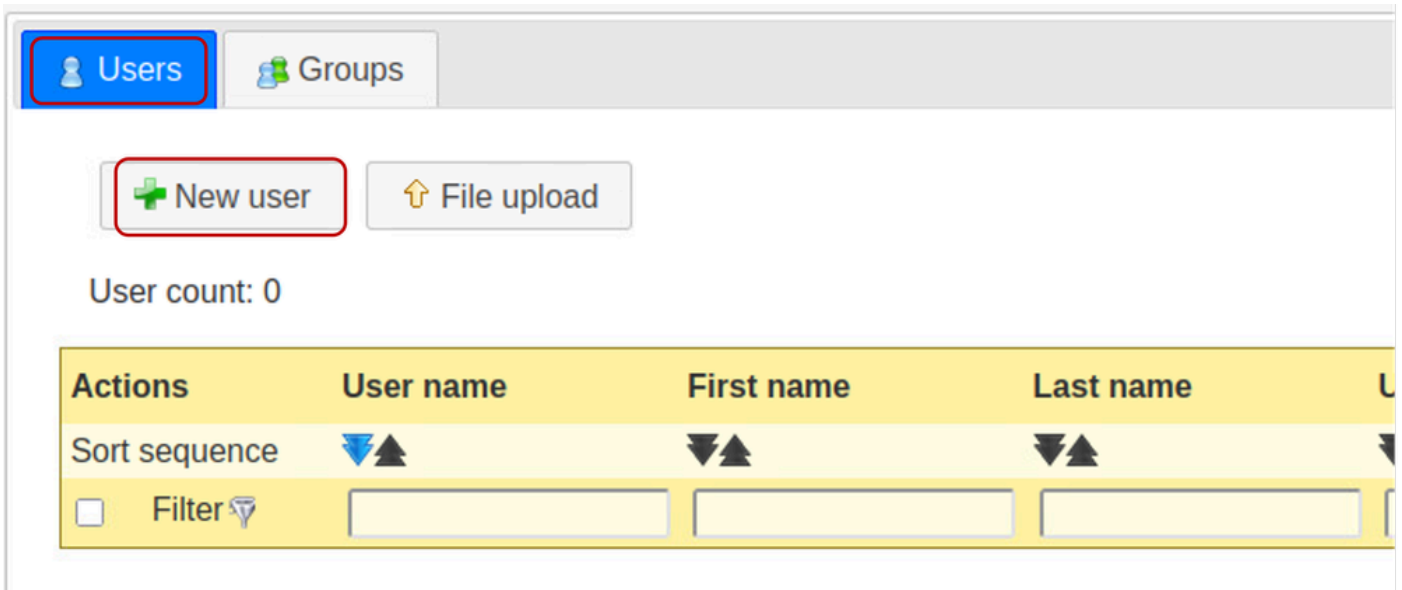
Group name * it

GID number

Description

Group members Edit members

「Users」タブをクリックしてユーザアカウントを作成し、「New user」を選択します。



「Personal」タブの「testuser1」ユーザの必須フィールドに入力します。

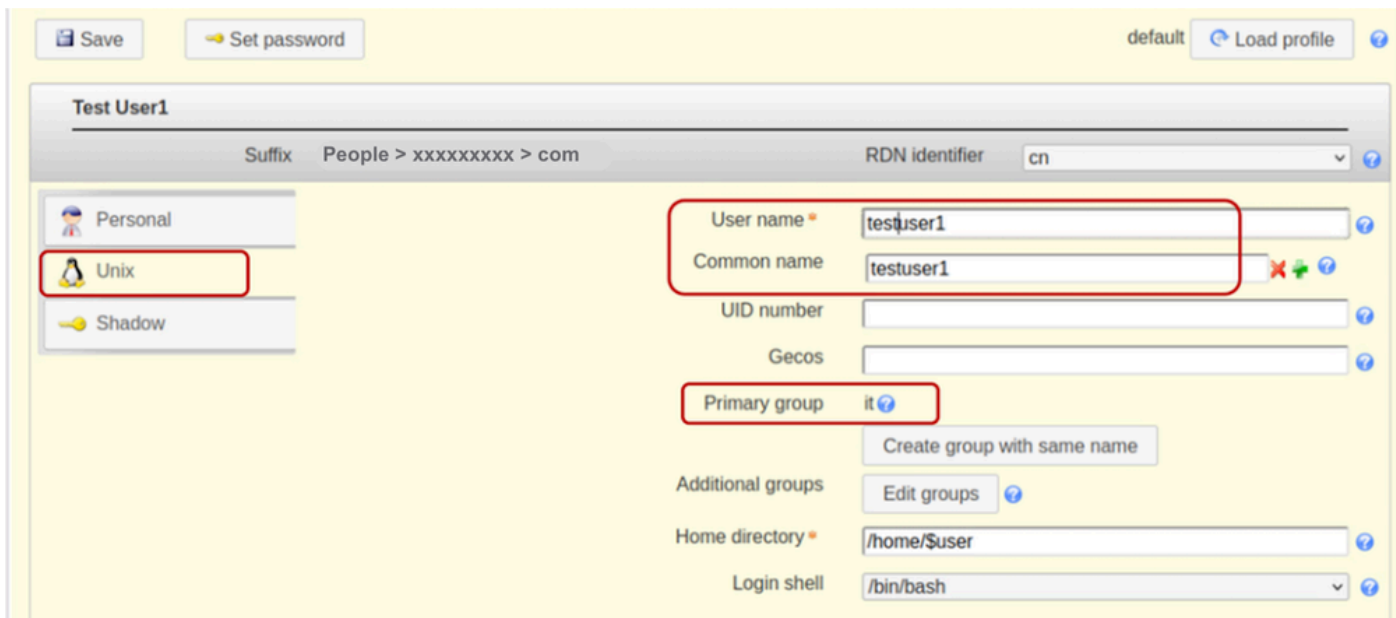


Unixタブを選択し、User nameフィールドにtestuser1を追加します。「it」グループにユーザを含めます。

このデモンストレーションでは、「it」グループのみが存在するため、すでに入力されています。

RDN IDを「Common Name」(cn)として維持します。これにより、「ユーザー名」フィールドに指定した値を使用して、「共通名」フィールドに自動的に入力できます。

LAMが使用可能な値をフィールドに自動的に入力するため、[UID番号]フィールドは空白のままにします。



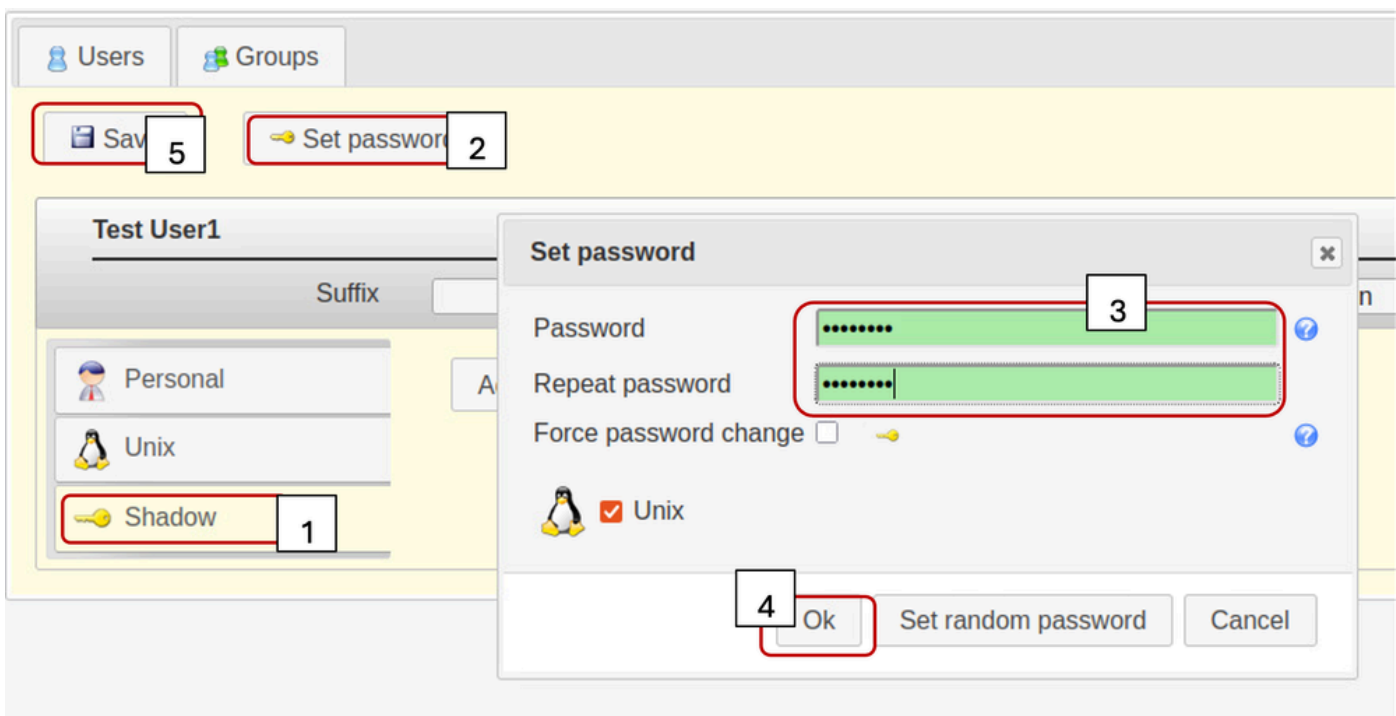
「シャドウ」タブを選択します。

シャドウアカウント拡張は使用されません。

「パスワードの設定」をクリックしてください。

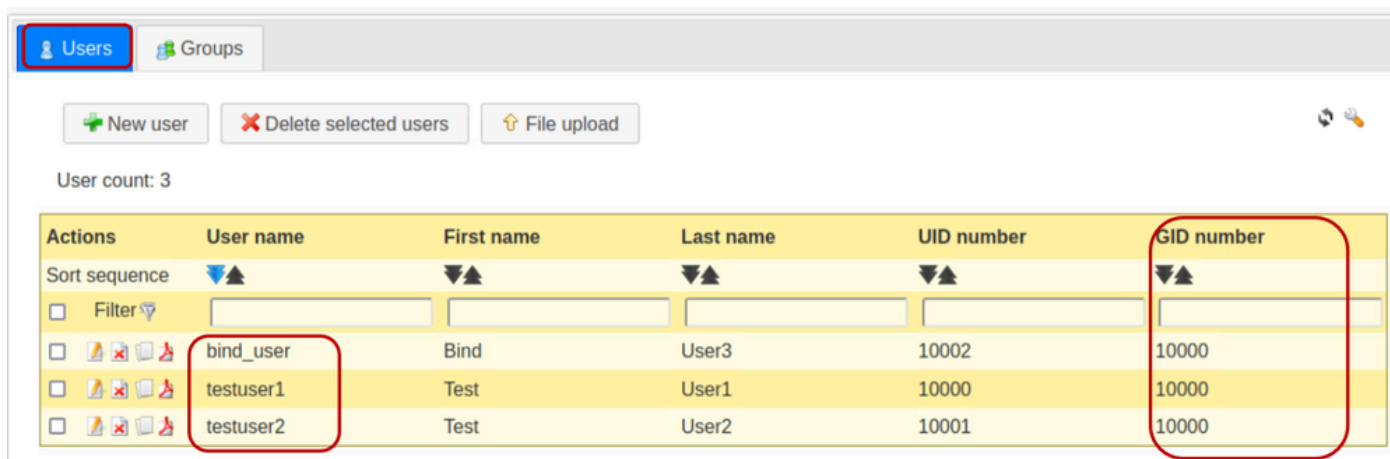
ユーザパスワードの設定

「OK」をクリックして保存します



前述の手順を繰り返して、「testuser2」ユーザアカウントと「bind_user」アカウントを作成します。

「ユーザー」タブをクリックして、目的のすべてのユーザーの作成を確認します。
(gidNumberカラムに同じ値を入力すると、作成されたユーザが同じグループに属していることを確認できます)。



Actions	User name	First name	Last name	UID number	GID number
Sort sequence					
Filter					
<input type="checkbox"/>	bind_user	Bind	User3	10002	10000
<input type="checkbox"/>	testuser1	Test	User1	10000	10000
<input type="checkbox"/>	testuser2	Test	User2	10001	10000

ステップ6：ローカルLDAPログインをテストします。

OpenLDAPサーバに到達可能な別のLinuxベースのシステムにログインします。
指定されたldapsearchコマンドを実行して、LDAPが機能していることを確認します。

```
ldapsearch -x -h X.X.X.19 -p 389 -b "dc=xxxxxxxx,dc=com" "uid=testuser1" sn cn givenName
```

```
root@kali:~# ldapsearch -x -h 192.168.1.19 -p 389 -b "dc=xxxxxxxx,dc=com" "uid=testuser1" sn cn givenName
n givenName
# extended LDIF
#
# LDAPv3
# base <dc=xxxxxxxx,dc=com> with scope subtree
# filter: uid=testuser1
# requesting: sn cn givenName
#
# testuser1, People, xxxxxxxx,dc=com
dn: cn=testuser1,ou=People,dc= xxxxxxxx,dc=com
cn: testuser1
sn: User1
givenName: Test
# search result
search: 2
result: 0 Success
# numResponses: 2
# numEntries: 1
root@kali:~#
```

CIMCの設定パラメータ

CIMCにログインします。

ナビゲーションペインでAdmin、User Management、およびLDAPを選択します。

次に示すように、LDAP設定パラメータを入力します。

- Enable LDAP : オン
- ベースDN:dc=xxxxxxxxx,dc=com

- ドメイン : xxxxxxxxxxx.com

- LDAPサーバ : <ldap_server_IPまたはFQDN> X.X.X.19

- バインドパラメータ : 「ログインクレデンシャル」または「設定されたクレデンシャル」
 - 設定済みのクレデンシャルを使用する場合は、LDAPサーバで設定されているとおりにbind_user DNを追加します。
 - 例 : cn=bind_user,ou=People,dc=xxxxxxxxx,dc=com

- 検索パラメータ :
 - フィルタ属性 : 「cn」または「uid」
 - グループ属性 : memberUID

- LDAPグループの許可 : オン
 - グループ名 : it
 - グループドメイン : xxxxxxxxxxx.com
 - ロール : 読み取り専用 (任意のロール)

Test LDAP Binding | Export LDAP CA Certificate

LDAP Settings

Enable LDAP:
 Base DN: dc=xxxxxxxx,dc=com
 Domain: xxxxxxxx.com

Enable Secure LDAP:
 Timeout (for each server): 60 (0-180) seconds

Binding Parameters

Method: Configured Credentials
 Binding DN: cn=bind_user,ou=People,dc=xx
 Password:

Search Parameters

Filter Attribute: uid
 Group Attribute: memberUID
 Attribute:
 Nested Group Search Depth: 128 (1 - 128)

LDAP CA

Configure LDAP Servers

Pre-Configure LDAP Servers

LDAP Servers

1.	9	389
2.		389
3.		389
4.		3268
5.		3268
6.		3268

Use DNS to Configure LDAP Servers
 DNS Parameters

Group Authorization

LDAP Group Authorization:

Index	Group Name	Group Domain	Role
<input type="checkbox"/> 1	it	xxxxxxxx.com	read-only
<input type="checkbox"/> 2			
<input type="checkbox"/> 3			
<input type="checkbox"/> 4			

設定を保存し、LDAPユーザログインをテストします。

UCS Managerの設定パラメータ

UCS Managerにログインします。

ナビゲーションペインでAdmin、User Management、およびLDAPを選択します。

次に示すように、LDAP設定パラメータを入力します。

- LDAPプロバイダー :
 - Hostname: <LDAPサーバのFQDNまたはIPアドレス>
 - バインドDN: cn=bind_user,ou=People,dc=xxxxxxxx,dc=com
 - ベースDN:dc=xxxxxxxx,dc=com
 - ポート : 389
 - SSLを有効にする : 無効
 - フィルタ : uid=\$userid
 - グループ許可 : 有効
 - グループの再帰 : 再帰なし
 - ターゲット属性 : gidNumber
- LDAPグループマップ :
 - LDAPグループDN:10000 <gidNumber for "it" group>

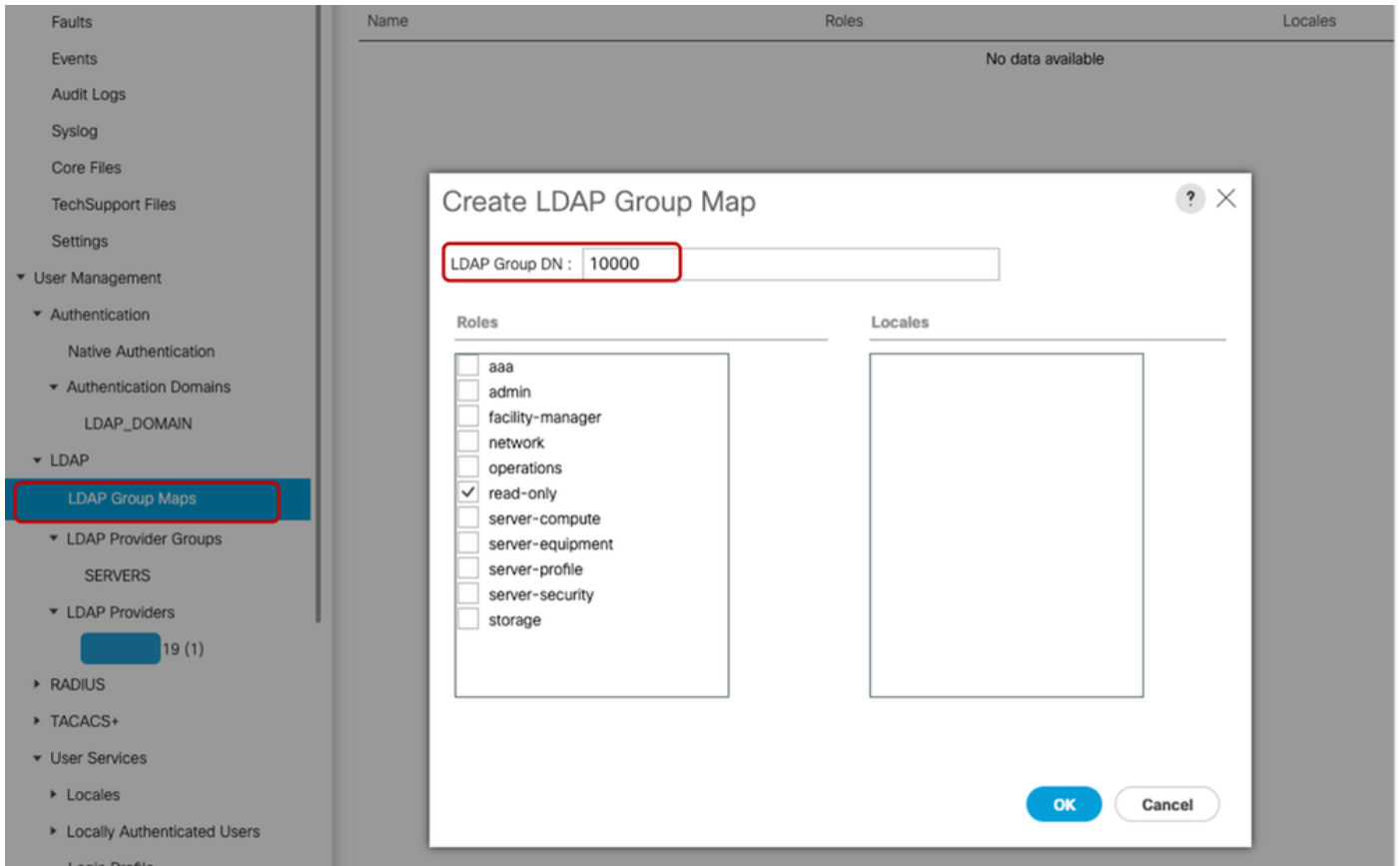
The screenshot displays the configuration page for an LDAP provider in the UCS Manager. The left-hand navigation pane shows the hierarchy: All > User Management > LDAP > LDAP Providers. The main content area is titled 'Properties' and contains several configuration fields, many of which are circled in red in the image. These include: Hostname/FQDN (or IP Address) set to '19'; Bind DN set to 'cn=bind_user,ou=People,dc=xxxxxxxx,dc=com'; Base DN set to 'dc=xxxxxxxx,dc=com'; Port set to '389'; Filter set to 'uid=\$userid'; Vendor set to 'Open Ldap' (with 'MS AD' also visible); Group Authorization set to 'Enable'; Group Recursion set to 'Non Recursive'; and Target Attribute set to 'gidNumber'. A 'Set: Yes' button is located to the right of the configuration fields.

All >> User Management >> LDAP >> LDAP Providers>> LDAP Group Rulesの下で、UCS Managerのデフォルトのターゲット属性は「memberOf」です。デフォルトでは、OpenLDAPサーバでその属性が有効になっていません。そのため、Target Attributeの値を「memberOf」に設定すると（または空白のままにすると）、要求されたAttributeの値がOpenLDAPサーバで認識されないため、ユーザログインが失敗します。

この例では、「Target Attribute」値が「gidNumber」に設定されています。

設定したLDAPプロバイダーをLDAPプロバイダーグループに追加します。このデモンストレーションでは、「SERVERS」LDAPプロバイダーグループが作成されました。

「All >> User Management >> LDAP >> LDAP Group Maps>>」で「LDAP Group Maps」を設定する場合、gidNumber値(この場合は「10000」)が次のように「Group DN Map」として使用されます。



LDAPプロバイダグループを参照する「All >> User Management >> Authentication >> Authentication Domains」でLDAP認証ドメイン(LDAP_DOMAIN)を設定し、LDAPユーザログインをテストします。



注：特定の環境要件を満たすために、または「グループ再帰」機能を実装するためにmemberOf属性が必要な場合は、次の2番目の設定オプションを使用することをお勧めします。この設定オプションではオーバーレイ拡張を有効にしたLDAPが必要です。

LDAPアカウントマネージャ(LAM)はオーバーレイ設定をサポートしていますが、この機能には適切なライセンスが必要です。

LAMを使用したLDAPの設定の詳細については、[LDAPアカウントマネージャの公式ドキュメント](#)を参照してください。

オプション2:Ubuntu CLIツールとオーバーレイを使用してOpenLDAPを設定する

UCS Managerの認証にOpenLDAPを使用するには、2つのオーバーレイを用意して、UCSシステム (UCS ManagerとCIMC) が認識できる方法でグループをユーザに関連付ける必要があります。

OpenLDAP側の設定には次が必要です。

- 「memberof」オーバーレイ：このオーバーレイは、ユーザDNがクエリーされた場合に、そのクエリーの一部としてmemberOf属性を要求できるように、ユーザとグループ間のマッピングを作成します。デフォルトでは、openLDAPにmemberof overlayが追加されない限り、グループメンバーシップのユーザの属性はありません
- "refint"オーバーレイ：このオーバーレイは、グループオブジェクトのmember属性のエントリがユーザーオブジェクトのmemberOf属性と同期されていることを検証するように構成されます。このサービスを使用しない場合、グループも変更せずにユーザを削除すると、孤立したDNがグループオブジェクトに残る可能性があります。絞り込みサービスは、両方向の一貫性を保証します。

ステップ1:初期net-toolsとLinuxサーバホスト名の設定

オプション1内でステップ1を繰り返します。

ステップ2: SLAPDのインストール

オプション1内でステップ2を繰り返します (オプション2では動作する必要がないため、PHPおよびApacheのインストールを除き、LAMはありません)。

必要なポートがUbuntuファイアウォールを通過できることを確認します。

ステップ3:LDAPサーバに「memberOf」オーバーレイをインストールします。

「memberOf」オーバーレイがインストールされているかどうかを確認します

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config '(olcModuleLoad=*)'
```

```
test@test:~$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config '(olcModuleLoad=*)'  
dn: cn=module{0},cn=config  
objectClass: olcModuleList  
cn: module{0}  
olcModulePath: /usr/lib/ldap  
olcModuleLoad: {0}back_mdb
```

「memberOf」オーバーレイをインストールするには、ldap.memberof.load.ldifという名前のldifファイルを作成し (任意の命名規則を使用)、指定した構成を追加します。

```
cat <
```

```
./ldap.memberof.load.ldif
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module olcModuleLoad: memberof
EOF
```

指定したコマンドを使用して、ldap.memberof.load.ldifファイル内の設定をLDAPプロファイルに追加します。

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f ./ldap.memberof.load.ldif
```

Linuxディストリビューションに応じて、配備要件に一致するようにmemberOfモジュールとolcDatabaseエントリを設定します。

2つの必須属性値は、次に示すように「olcDatabase={1}mdb」と「groupOfNames」です。

ldap.memberof.config.ldifファイルを作成し、その属性を入力して、その内容をLDAPプロファイルにインポートします。

```
cat <
```

```
./ldap.memberof.config.ldif
dn: olcOverlay=memberof,olcDatabase={1}mdb,cn=config
objectClass: olcMemberOf
objectClass: olcOverlayConfig
olcOverlay: memberof
olcMemberOfGroupOC: groupOfNames
olcMemberOfMemberAD: member
olcMemberOfMemberOfAD: memberOf
olcMemberOfRefInt: TRUE
olcMemberOfDangling: ignore
EOF
```

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f ./ldap.memberof.config.ldif
```

ステップ4:LDAPサーバに「refint」オーバーレイをインストールします。

次に、refintをopenldapにインストールします。

ldap.refint.load.ldifという名前の.ldifファイルを作成し（任意の命名規則を使用）、指定した設定を追加します。

```
cat <
```

```
./ldap.refint.load.ldif
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModuleLoad: refint
EOF
```

次のコマンドを使用して、ldap.refint.load.ldifファイルの設定をLDAPプロファイルにインポートします。

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f ./ldap.refint.load.ldif
```

絞り込みを構成します。これにより、グループとユーザー間の参照整合性が維持されます。

展開の要件に合わせて絞り込みモジュールとそのolcDatabaseエントリを構成します。

ldap.refint.config.ldifファイルを作成し、その内容をLDAPプロファイルにインポートします。

```
cat <
```

```
./ldap.refint.config.ldif
dn: olcOverlay=refint,olcDatabase={1}mdb,cn=config
objectClass: olcConfig
objectClass: olcOverlayConfig
objectClass: olcRefintConfig
olcOverlay: refint
olcRefintAttribute: memberOf member
EOF
```

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f ./ldap.refint.config.ldif
```

両方のプラグイン/拡張機能をインストールすると、指定したldapsearchコマンドの出力は次のようになります。

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config '(olcModuleLoad=*)'
```

```
[test@test:~$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config '(olcModuleLoad=*)'  
dn: cn=module{0},cn=config  
objectClass: olcModuleList  
cn: module{0}  
olcModulePath: /usr/lib/ldap  
olcModuleLoad: {0}back_mdb  
  
dn: cn=module{1},cn=config  
objectClass: olcModuleList  
cn: module{1}  
olcModuleLoad: {0}memberof  
  
dn: cn=module{2},cn=config  
objectClass: olcModuleList  
cn: module{2}  
olcModuleLoad: {0}refint
```

両方のプラグイン/拡張機能が構成されている場合、指定したldapsearchコマンドの出力は次のようになります。

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config '(olcOverlay=memberof)'
```

```
[test@test:~$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config '(olcOverlay=memberof)'  
dn: olcOverlay={0}memberof,olcDatabase={1}mdb,cn=config  
objectClass: olcMemberOfConfig  
objectClass: olcOverlayConfig  
olcOverlay: {0}memberof  
olcMemberOfDangling: ignore  
olcMemberOfRefInt: TRUE  
olcMemberOfGroupOC: groupOfNames  
olcMemberOfMemberAD: member  
olcMemberOfMemberOfAD: memberOf  
  
test@test:~$
```

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config '(olcOverlay=refint)'
```

```
test@test:~$ sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config '(olcOverlay=refint)'
dn: olcOverlay={1}refint,olcDatabase={1}mdb,cn=config
objectClass: olcConfig
objectClass: olcOverlayConfig
objectClass: olcRefintConfig
olcOverlay: {1}refint
olcRefintAttribute: memberOf member
```

新しくインストールしたプラグイン/モジュールを使用可能にするために、slapdサービスを再起動します。

```
sudo systemctl restart slapd
```

ステップ5:OU、ユーザ、およびグループの作成

組織単位 (ユーザーとグループ)、ユーザーとグループを作成します。

ユーザ(People)OUとグループ(Groups)OUを作成し、LDAPプロファイルにインポートします。これには、「admin」アカウントのパスワードが必要です。

```
cat <
```

```
./ldap.ou.add.ldif
dn: ou=People,dc=xxxxxxxx,dc=com
objectClass: organizationalUnit
ou: People

dn: ou=Groups,dc=xxxxxxxx,dc=com
objectClass: organizationalUnit
ou: Groups
EOF
```

```
sudo ldapadd -xwD cn=admin,dc=xxxxxxxx,dc=com -f ./ldap.ou.add.ldif
```

```

test@test:~$ cat <<EOF > ./ldap.ou.add.ldif
dn: ou=People,dc=xxxxxxxx,dc=com
objectClass: organizationalUnit
ou: People

dn: ou=Groups,dc=xxxxxxxx,dc=com
objectClass: organizationalUnit
ou: Groups
EOF
test@test:~$
test@test:~$ sudo ldapadd -xwD cn=admin,dc=xxxxxxxx,dc=com -f ./ldap.ou.add.ldif
Enter LDAP Password:
adding new entry "ou=People,dc=xxxxxxxx,dc=com"

adding new entry "ou=Groups,dc=xxxxxxxx,dc=com"

test@test:~$ █

```

ユーザ(testuser1、testuser2、bind_user)を作成し、それぞれのOU(People)にマッピングし、gidNumbersを使用してグループに追加し(適切な方法)、ユーザをLDAPプロファイルにインポートします。

cat <

```

./ldap.users.ldif
dn: uid=testuser1,ou=People,dc=xxxxxxxx,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: testuser1
sn: User1
givenName: Test
cn: testuser1
displayName: Test User1
gidNumber: 10000
uidNumber: 10000
userPassword: cisco123
gecos: Test User1
loginShell: /bin/bash
homeDirectory: /home/testuser1

```

```

dn: uid=testuser2,ou=People,dc=xxxxxxxx,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: testuser2
sn: User2
givenName: Test
cn: testuser2
displayName: Test User2
gidNumber: 10000
uidNumber: 10001
userPassword: cisco123
gecos: Test User2
loginShell: /bin/bash
homeDirectory: /home/testuser2

```

```
dn: uid=bind_user,ou=People,dc=xxxxxxxx,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: bind_user
sn: User3
givenName: Bind
cn: bind_user
displayName: Bind User3
gidNumber: 10001
uidNumber: 10002
userPassword: cisco123
gecos: Bind User3
loginShell: /bin/bash
homeDirectory: /home/bind_user
EOF
```

```
sudo ldapadd -x cWD cn=admin,dc=xxxxxxxx,dc=com -f ./ldap.users.ldif
```

```
test@test:~$ cat <<EOF > ./ldap.users.ldif
dn: uid=testuser1,ou=People,dc=xxxxxxxx,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: testuser1
sn: User1
givenName: Test
cn: testuser1
displayName: Test User1
gidNumber: 10000
uidNumber: 10000
userPassword: cisco123
gecos: Test User1
loginShell: /bin/bash
homeDirectory: /home/testuser1

dn: uid=testuser2,ou=People,dc=xxxxxxxx,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: testuser2
sn: User2
givenName: Test
cn: testuser2
displayName: Test User2
gidNumber: 10000
uidNumber: 10001
userPassword: cisco123
gecos: Test User2
loginShell: /bin/bash
homeDirectory: /home/testuser2

dn: uid=bind_user,ou=People,dc=xxxxxxxx,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: bind_user
sn: User3
givenName: Bind
cn: bind_user
displayName: Bind User3
gidNumber: 10001
uidNumber: 10002
userPassword: cisco123
gecos: Bind User3
loginShell: /bin/bash
homeDirectory: /home/bind_user
EOF
[test@test:~$ sudo ldapadd -xwD cn=admin,dc=xxxxxxxx,dc=com -f ./ldap.users.ldif
[Enter LDAP Password:
adding new entry "uid=testuser1,ou=People,dc=xxxxxxxx,dc=com

adding new entry "uid=testuser2,ou=People,dc=xxxxxxxx,dc=com

adding new entry "uid=bind_user,ou=People,dc=xxxxxxxx,dc=com

test@test:~$ █
```

グループ(it)を作成し、対応するOU(Groups)にマッピングし、グループメンバー(testuser1、testuser2)を関連付けて、LDAPプロファイルにインポートします。

```
cat <
```

```
./ldap.group.create.ldif
dn: cn=it,ou=Groups,dc=xxxxxxxx,dc=com
objectClass: groupofnames
cn: it
member: uid=testuser1,ou=People,dc=xxxxxxxx,dc=com
member: uid=testuser2,ou=People,dc=xxxxxxxx,dc=com
EOF
```

```
sudo ldapadd -x cWD cn=admin,dc=xxxxxxxx,dc=com -f ./ldap.group.create.ldif
```

```
test@test:~$ cat <<EOF > ./ldap.group.create.ldif
dn: cn=it,ou=Groups,dc=xxxxxxxx,dc=com
objectClass: groupofnames
cn: it
member: uid=testuser1,ou=People,dc=xxxxxxxx,dc=com
member: uid=testuser2,ou=People,dc=xxxxxxxx,dc=com
EOF
test@test:~$ sudo ldapadd -x cWD cn=admin,dc=xxxxxxxx,dc=com -f ./ldap.group.create.ldif
Enter LDAP Password:
adding new entry "cn=it,ou=Groups,dc=xxxxxxxx,dc=com"
test@test:~$
```



注：ユーザまたはグループの作成時にmemberOf属性が明示的に定義されていない場合でも、この参照は自動的に生成されて維持されます。ユーザをグループに関連付けると、memberOf属性にこれらのメンバーシップが自動的に反映され、ディレクトリと現在のアクセス構造との同期が維持されます。

ステップ6: ローカルLDAPログインをテストします。

指定したコマンドを使用して、LDAPサーバへのユーザログインを確認します（環境に応じてログインパラメータを置き換えます）。

```
sudo ldapsearch -x -LLL -b uid=testuser1,ou=People,dc=xxxxxxxx,dc=com memberOf
```

```
test@test:~$ sudo ldapsearch -x -LLL -b uid=testuser1,ou=People,dc=xxxxxxxx,dc=com memberOf
dn: uid=testuser1,ou=People,dc=xxxxxxxx,dc=com
memberOf: cn=it,ou=Groups,dc=xxxxxxxx,dc=com

test@test:~$ █
```

CIMCの設定パラメータ

CIMCにログインします。

ナビゲーションペインでAdmin、User Management、およびLDAPを選択します。

次に示すように、LDAP設定パラメータを入力します。

- Enable LDAP : オン
- ベースDN:dc=xxxxxxxx,dc=com

- ドメイン : xxxxxxxxxxx.com

- LDAPサーバ : <ldap_server_IPまたはFQDN> X.X.X.19

- バインドパラメータ : 「ログインクレデンシャル」または「設定されたクレデンシャル」
 - 設定済みのクレデンシャルを使用する場合は、LDAPサーバで設定されているとおりにbind_user DNを追加します。
 - 例 : 「cn=bind_user,ou=People,dc=xxxxxxxx,dc=com」または「uid=bind_user,ou=People,dc=xxxxxxx,dc=com」

- 検索パラメータ :
 - フィルタ属性 : 「cn」または「uid」
 - グループ属性 : メンバー

- LDAPグループの許可 : オン
 - グループ名 : it
 - グループドメイン : xxxxxxxxxxx.com
 - ロール : 読み取り専用 (任意の優先ロール)

Home / ... / User Management / LDAP ★ Refresh | Help

Local User Management | LDAP | TACACS+ | Session Management

Test LDAP Binding | Export LDAP CA Certificate

▼ LDAP Settings

Enable LDAP:
 Base DN:
 Domain:
 Enable Secure LDAP:
 Timeout (for each server): (0-180) seconds

▼ Binding Parameters

Method:
 Binding DN:
 Password:

▼ Search Parameters

Filter Attribute:
 Group Attribute:
 Attribute:
 Nested Group Search Depth: (1 - 128)

▶ LDAP CA

▼ Configure LDAP Servers

Pre-Configure LDAP Servers
 LDAP Servers

1.	<input type="text" value="9"/>	<input type="text" value="389"/>
2.	<input type="text"/>	<input type="text" value="389"/>
3.	<input type="text"/>	<input type="text" value="389"/>
4.	<input type="text"/>	<input type="text" value="3268"/>
5.	<input type="text"/>	<input type="text" value="3268"/>
6.	<input type="text"/>	<input type="text" value="3268"/>

Use DNS to Configure LDAP Servers
 DNS Parameters

▼ Group Authorization

LDAP Group Authorization:

Index	Group Name	Group Domain	Role
<input type="checkbox"/> 1	it	xxxxxxxx.com	read-only
<input type="checkbox"/> 2			
<input type="checkbox"/> 3			
<input type="checkbox"/> 4			

設定を保存し、LDAPユーザログインをテストします。

UCS Managerの設定パラメータ

UCS Managerにログインします。

ナビゲーションペインでAdmin、User Management、およびLDAPを選択します。

次に示すように、LDAP設定パラメータを入力します。

- LDAPプロバイダー：
 - Hostname: <LDAPサーバのFQDNまたはIPアドレス>
 - バインドDN: uid=bind_user,ou=People,dc=xxxxxxxx,dc=com
 - ベースDN: dc=xxxxxxxx,dc=com
 - ポート：389
 - SSLを有効にする：無効
 - フィルタ：uid=\$userid
 - グループ許可：有効
 - グループ再帰：再帰
 - ターゲット属性：memberOf
- LDAPグループマップ：
 - LDAPグループDN: cn=it,ou=Groups,dc=xxxxxxxx,dc=com

General Events

Actions

Delete

Properties

Hostname/FQDN (or IP Address) : 19

Order : 1

Bind DN : uid=bind_user,ou=People,dc=xxxxxxxx,dc=com

Base DN : dc=xxxxxxxx,dc=com

Port : 389

Enable SSL :

Filter : uid=\$userid

Attribute :

Password :

Confirm Password :

Timeout : 30

Vendor : Open Ldap MS AD

LDAP Group Rules

Group Authorization : Disable Enable

Group Recursion : Non Recursive Recursive

Target Attribute : memberOf

Use Primary Group :

Set: Yes

設定したLDAPプロバイダーをLDAPプロバイダーグループに追加します。このデモンストレーションでは、「SERVERS」LDAPプロバイダーグループを使用します。

LDAPサーバから取得した「LDAPグループDN」を追加して、LDAPグループマップを設定します。

LDAP Group Maps

Advanced Filter Export Print

Name	Roles

Create LDAP Group Map

LDAP Group DN : cn=it,ou=Groups,dc=xxxxxxxx,dc=com

Roles

Locales

aaa

admin

facility-manager

network

operations

read-only

server-compute

server-equipment

server-profile

server-security

storage

testrole

OK Cancel

LDAPプロバイダグループ(SERVERS)を参照する「All >> User Management >> Authentication >> Authentication Domains」でLDAP認証ドメイン(LDAP_DOMAIN)を設定し、LDAPユーザログインをテストします。

次に、別のLinuxディストリビューション(CentOS 10)で同じ設定 (オーバーレイあり) を行う方法を見てみましょう

シナリオ2:CentOSストリーム10 - Fedora

Lightweight Directory Access Protocol(LDAP)の設定手順は、基盤となるオペレーティングシステムのバージョンによって異なります。このセクションでは、CentOS Stream 10でのLDAPの実装を中心に説明します。

Linuxディストリビューションの多くはOpenLDAPを利用していますが、CentOS Stream 10や現在のFedoraベースのシステムでは、デフォルトのLDAPプロバイダーとして389 Directory Server(389 DS)を利用しています。



注:389 DSはCentOSおよびRed Hatエコシステム内のOpenLDAPの後継と見なされていますが、この2つのソリューションは直接的には互換性はありません。それぞれのディレクトリ構造、コンフィギュレーションファイル、および運用環境は大きく異なります。

このガイドでは、CentOS Stream 10環境内で389 DSを使用してLDAPを正しく設定するために必要な手順について説明します。

オプション1:CentOSストリーム10で389ディレクトリサーバを使用してLDAPを設定する

ステップ1：初期設定

シナリオ1、オプション1のステップ1を繰り返します。

CentOSシステムでは、APTパッケージ管理スイートは使用されません。CentOS Stream 10で必要なソフトウェアインストールを実行するには、dnf(Dandified YUM)またはyumパッケージマネージャを使用します

```
sudo yum update
sudo yum install net-tools
```

「ifconfig」コマンドを使用して、サーバのIPアドレスを確認します。

サーバのIPアドレスを「/etc/hosts」ファイルに追加します。このファイルには、サーバの完全修飾ドメイン名(例：この実習で使ったtest.xxxxxxx.com)とホスト名(例：test)を指定します。

```
sudo nano /etc/hosts
```

```
GNU nano 8.1 /etc/hosts
Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
.19 test.xxxxxxxx.com test
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.example.org foo
# 192.168.1.13 bar.example.org bar
```

「/etc/hostname」ファイルの内容をホスト名に置き換えて(test)、ファイルを更新します。

```
sudo nano /etc/hostname
```

```
GNU nano 8.1 /etc/hostname
test
```

これらの変更を有効にするには、サーバをリブートする必要があります。

```
sudo reboot
```

ステップ2: EPELリポジトリと389 Serverパッケージをインストールします。

EPELリポジトリのインストールと更新

389 Directory Serverパッケージをインストールします。

```
sudo dnf install -y epel-release
sudo dnf update -y epel-release
sudo dnf install 389-ds-base
```

目的のLDAPサーバ設定パラメータを含むディレクトリテンプレートファイルを作成します。

```
sudo dscreate create-template ldapconfig.conf
```

作成したテンプレートファイル(ldapconfig.conf)の内容を確認します。

```
sudo cat ldapconfig.conf
```

ldapconfig.confテンプレートファイルを編集します。

```
sudo nano ldapconfig.conf
```

指定した構成エントリをファイルに挿入し、変更を保存します。



注：各環境の特定のニーズまたは要件に応じて、異なる変更が必要になる場合があります。

この例では、このデモンストレーションのベースライン設定について説明します。

```
[general]
config_version = 2
selinux      = True
```

```
[slapd]
instance_name = localhost
root_dn       = cn=admin
root_password = cisco123
```

```
[backend-userroot]
sample_entries = yes
suffix = dc=xxxxxxxx,dc=com
```

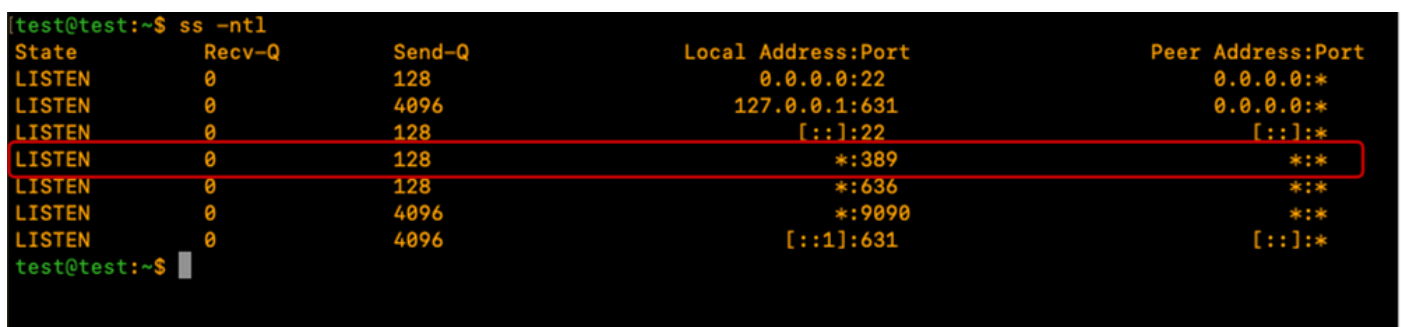
テンプレートファイルは、「localhost」ディレクトリインスタンスの設定パラメータを定義します。これには、管理ユーザ(「admin」)、関連するパスワード、およびドメインコンテキスト(「xxxxxxxx.com」)の設定が含まれます。

先ほど編集したテンプレートを使用して、「localhost」ディレクトリインスタンスを作成します。指定したコマンドにより、LDAPディレクトリサーバが作成され、起動されます。

```
sudo dscreate -v from-file ldapconfig.conf
```

LDAPサービスがサーバで実行されていることを確認します

```
ss -ntl
```



```
test@test:~$ ss -ntl
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
LISTEN     0            128         0.0.0.0:22               0.0.0.0:*
LISTEN     0            4096        127.0.0.1:631            0.0.0.0:*
LISTEN     0            128         [::]:22                  [::]:*
LISTEN     0            128         *:389                    *:
LISTEN     0            128         *:636                    *:
LISTEN     0            4096        *:9090                   *:
LISTEN     0            4096        [::1]:631                [::]:*
```

CentOSファイアウォールを調整して、LDAP (389または636) に必要なポートを許可します。

このデモでは、ファイアウォールはオフになっています。

```
sudo systemctl stop firewallld
```

指定されたコマンドを実行して、LDAPがLDAPサーバでローカルに動作することを確認し、次に示すようにLDAP出力が返されることを確認します。

```
sudo ldapsearch -x ldap://localhost -b "dc=xxxxxxxx,dc=com"
```

```
[test@test:~$ sudo ldapsearch -x ldap://localhost -b "dc=xxxxxxxx,dc=com"
# extended LDIF
#
# LDAPv3
# base <dc=xxxxxxxx,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ldap://localhost
#
# xxxxxxxxxx,com
dn: dc=xxxxxxxx,dc=com

# groups, xxxxxxxxxx,com
dn: ou=groups, dc=xxxxxxxx,dc=com

# people, xxxxxxxxxx,com
dn: ou=people, dc=xxxxxxxx,dc=com

# permissions, xxxxxxxxxx,com
dn: ou=permissions, dc=xxxxxxxx,dc=com

# services, xxxxxxxxxx,com
dn: ou=services, dc=xxxxxxxx,dc=com

# demo_user, people, xxxxxxxxxx,com
dn: uid=demo_user,ou=people, dc=xxxxxxxx,dc=com

# demo_group, Groups, xxxxxxxxxx,com
dn: cn=demo_group,ou=Groups, dc=xxxxxxxx,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 8
# numEntries: 7
```

出力には、389DSサーバによって作成されたデモアカウントが含まれています。LDAPサーバが自動的にデフォルトOUを作成しました。

ユーザのユーザOUとグループのグループOU必要に応じて、追加のOUを作成できます。

このデモンストレーションでは、デフォルトまたは自動作成のOUを使用します。

389DSパッケージの広範な使用方法の詳細については、[公式の389DSのドキュメント](#)を参照してください。

ステップ3:LDAPグループとユーザの作成

指定されたコマンド `sudo dsidm <instance_name> group create` を使用してグループ(it)を作成します。

このデモンストレーションでは、インスタンス名は「localhost」です。

```
sudo dsidm localhost group create
```

ターミナルプロンプトを入力して、次に示すようにグループの詳細を入力します。

```
[test@test:~$ sudo dsidm localhost group create
[sudo] password for test:
[Enter basedn : dc=xxxxxxxxx,dc=com
[Enter value for cn : it
Successfully created it
test@test:~$ █
```

次のコマンドを使用して、testuser1ユーザアカウントを作成します。

```
sudo dsidm localhost user create
```

ターミナルプロンプトを入力し、次に示すようにユーザの詳細を入力します

```
[test@test:~$ sudo dsidm localhost user create
[Enter basedn : dc=xxxxxxxx,dc=com
[Enter value for uid : testuser1
[Enter value for cn : testuser1
[Enter value for displayName : Test User1
[Enter value for uidNumber : 10000
[Enter value for gidNumber : 10000
[Enter value for homeDirectory : /home/testuser1
Successfully created testuser1
```

指定したコマンドを使用してtestuser1のパスワードを作成し、CLIプロンプトを入力します。

```
sudo dsidm localhost account reset_password uid=testuser1,ou=people,dc=xxxxxxxx,dc=com
```

```
test@test:~$ sudo dsidm localhost account reset_password uid=testuser1,ou=people,dc=xxxxxxxx,dc=com
Enter basedn : dc=xxxxxxxx,dc=com
Enter new password for uid=testuser1,ou=people,dc=xxxxxxxx,dc=com :
CONFIRM - Enter new password for uid=testuser1,ou=people,dc=xxxxxxxx,dc=com :
reset password for uid=testuser1,ou=people,dc=xxxxxxxx,dc=com
test@test:~$
```

次の指定したコマンドを使用して、ユーザをグループに追加します。sudo dsidm <directory_instance> group add_member <group_cn> <user_dn>」

```
sudo dsidm localhost group add_member it uid=testuser1,ou=people,dc=xxxxxxxx,dc=com
```

「ユーザーの作成」の手順を繰り返して、「testuser2」と「bind_user」を作成します。



注：各ユーザが目的のグループに明示的に追加されていることを確認してください。

この手順を省略すると、アクセスが制限されたり、認証が失敗したりする可能性があります。

bind_userアカウントはスタンドアロン・アカウントとして構成できるため、特定のグループのメンバーである必要はありません。このアカウントは、ディレクトリ環境内で管理およびサービス・レベルのアクセスを柔軟に管理できます。

Directoryインスタンスを再起動します。

```
sudo dsctl localhost restart
```

ステップ4:memberOf overlayのインストール

「memberOf」プラグインをインストールし、Directoryインスタンスを再起動します。

```
sudo dsconf localhost plugin memberof status
sudo dsconf localhost plugin memberof enable
sudo dsctl localhost restart
```

指定されたコマンドを使用して、「memberOf」プラグインを設定します：「sudo dsconf <directory_instance> plugin memberof set --scope <base_dn>」

```
sudo dsconf localhost plugin memberof set --scope dc=xxxxxxxx,dc=com
```

指定されたコマンド「sudo dsidm <directory_instance> user modify <uid> add:objectclass:nsmemberof」を使用して、ユーザーを有効な「memberOf」ターゲットとしてマークします。

```
sudo dsidm localhost user modify testuser1 add:objectclass:nsmemberof
sudo dsidm localhost user modify testuser2 add:objectclass:nsmemberof
```

```
[test@test:~$ sudo dsidm localhost user modify testuser1 add:objectclass:nsmemberof
Enter basedn : dc=xxxxxxxx,dc=com
Successfully modified uid=testuser1,ou=people, dc=xxxxxxxx,dc=com
[test@test:~$ sudo dsidm localhost user modify testuser2 add:objectclass:nsmemberof
Enter basedn : dc=xxxxxxxx,dc=com
Successfully modified uid=testuser2,ou=people, dc=xxxxxxxx,dc=com
[test@test:~$
```

ベースDNに「memberOf」フィックスアップを生成：「sudo dsconf <directory_instance> plugin memberof fixup <base_dn>」

```
sudo dsconf localhost plugin memberof fixup dc=xxxxxxxx,dc=com
```

```
test@test:~$ sudo dsconf localhost plugin memberof fixup dc=xxxxxxxx,dc=com
Adding fixup task entry...
Successfully added task entry "cn=memberOf_fixup_2025-05-13T14:54:11.926390,cn=memberOf task,cn=tasks,cn=config". This task is running in the background. To track its progress you can use the "fixup-status" command.
test@test:~$
```

ユーザ設定を確認します。

```
sudo dsidm localhost user get testuser1
sudo dsidm localhost user get testuser2
```

```
test@test:~$ sudo dsidm localhost user get testuser1
Enter basedn : dc=xxxxxxxx,dc=com
dn: uid=testuser1,ou=people, dc=xxxxxxxx,dc=com
cn: testuser1
displayName: Test User1
gidNumber: 10000
homeDirectory: /home/testuser1
memberOf: cn=it,ou=Groups,dc=xxxxxxxx,dc=com
objectClass: top
objectClass: nsPerson
objectClass: nsAccount
objectClass: nsOrgPerson
objectClass: posixAccount
objectClass: nsmemberof
uid: testuser1
uidNumber: 10000
userPassword: {PBKDF2-SHA512}100000$uJ+bQ90AQ4L2uynoUBt+QeV1W0tj0KZJ$B/1yULxaE3F3wrE+Qo/+KPnynHgN5vWUz fM9Mxp01qeHq9gXs863u
rkAZakF$mlrZVduqN/TRNZE4W/ZbRmECw==

test@test:~$ sudo dsidm localhost user get testuser2
Enter basedn : dc=xxxxxxxx,dc=com
dn: uid=testuser2,ou=people, dc=xxxxxxxx,dc=com
cn: testuser2
displayName: Test User2
gidNumber: 10000
homeDirectory: /home/testuser2
memberOf: cn=it,ou=Groups,dc=xxxxxxxx,dc=com
objectClass: top
objectClass: nsPerson
objectClass: nsAccount
objectClass: nsOrgPerson
objectClass: posixAccount
objectClass: nsmemberof
uid: testuser2
uidNumber: 10001
userPassword: {PBKDF2-SHA512}100000$efAcaYcRRHIU60AIMeHxvHPAAhW7yWc$TzeynBPP6qXBWpGe9nyq1sHetEsCq7ngwt+41hSwY2syZ9tvcSd
ZCXZbo8RK80hBSCoqTYpi1N5o0BqU6A1w==

test@test:~$
```

389DS LDAPサーバは、memberOf属性をサポートするようにmemberOfプラグインで設定されま
す。

CIMCの設定パラメータ

CIMCにログインします。

ナビゲーションペインでAdmin、User Management、およびLDAPを選択します。

次に示すように、LDAP設定パラメータを入力します。

- Enable LDAP : オン
- ベースDN:dc=xxxxxxxxx,dc=com

- ドメイン : xxxxxxxxxxx.com

- LDAPサーバ : <ldap_server_IPまたはFQDN> X.X.X.19

- バインドパラメータ : 「ログインクレデンシャル」または「設定されたクレデンシャル」
 - 設定済みのクレデンシャルを使用する場合は、LDAPサーバで設定されているとおりにbind_user DNを追加します。
 - 例 : 「cn=bind_user,ou=People,dc=xxxxxxxxx,dc=com」または「uid=bind_user,ou=People,dc=xxxxxxx,dc=com」

- 検索パラメータ :
 - フィルタ属性 : 「cn」または「uid」
 - グループ属性 : memberOf

- LDAPグループの許可 : オン
 - グループ名 : it
 - グループドメイン : xxxxxxxxxxx.com
 - ロール : 読み取り専用 (任意の優先ロール)

Home / ... / User Management / LDAP ★ Refresh | Help

Local User Management | **LDAP** | TACACS+ | Session Management

Test LDAP Binding | Export LDAP CA Certificate

▼ LDAP Settings

Enable LDAP: Base DN: dc=xxxxxxxx,dc=com
 Domain: xxxxxxxx.com

Enable Secure LDAP:
 Timeout (for each server): 60 (0-180) seconds

▼ Binding Parameters

Method: Configured Credentials Binding DN: uid=bind_user,ou=People,dc=xx
 Password:

▼ Search Parameters

Filter Attribute: uid Group Attribute: memberOf
 Attribute:
 Nested Group Search Depth: 128 (1 - 128)

▼ LDAP CA

▼ Configure LDAP Servers

Pre-Configure LDAP Servers
 LDAP Servers

1.	9	389
2.		389
3.		389
4.		3268
5.		3268
6.		3268

Use DNS to Configure LDAP Servers
 DNS Parameters

▼ Group Authorization

LDAP Group Authorization:

Configure Delete

Index	Group Name	Group Domain	Role
<input checked="" type="checkbox"/>	it	xxxxxxxx.com	read-only
<input type="checkbox"/>	2		
<input type="checkbox"/>	3		
<input type="checkbox"/>	4		
<input type="checkbox"/>	-		

設定を保存し、LDAPユーザログインをテストします。

UCS Managerの設定パラメータ

UCS Managerにログインします。

ナビゲーションペインでAdmin、User Management、およびLDAPを選択します。

次に示すように、LDAP設定パラメータを入力します。

- LDAPプロバイダー：
 - Hostname: <LDAPサーバのFQDNまたはIPアドレス>
 - バインドDN: uid=bind_user,ou=people,dc=xxxxxxxx,dc=com
 - ベースDN: dc=xxxxxxxx,dc=com
 - ポート：389
 - SSLを有効にする：無効
 - フィルタ：uid=\$userid
 - グループ許可：有効
 - グループ再帰：再帰
 - ターゲット属性：memberOf
- LDAPグループマップ：
 - LDAPグループDN: cn=it,ou=Groups,dc=xxxxxxxx,dc=com

LDAP Provider configuration page showing the following properties:

- Hostname/FQDN (or IP Address): 19
- Order: 1
- Bind DN: uid=bind_user,ou=People,dc=xxxxxxxx,dc=com
- Base DN: dc=xxxxxxxx,dc=com
- Port: 389
- Enable SSL:
- Filter: uid=\$userid
- Attribute:
- Password:
- Confirm Password:
- Timeout: 30
- Vendor: Open Ldap MS AD

LDAP Group Rules:

- Group Authorization: Disable Enable
- Group Recursion: Non Recursive Recursive
- Target Attribute: memberOf
- Use Primary Group:

Set: Yes

設定したLDAPプロバイダーをLDAPプロバイダーグループに追加します。このデモンストレーションでは、「SERVERS」LDAPプロバイダーグループを使用します。

LDAPサーバから取得した「LDAPグループDN」を追加して、LDAPグループマップを設定します。

Create LDAP Group Map dialog box showing the following configuration:

- LDAP Group DN: cn=it,ou=Groups,dc=xxxxxxxx,dc=com
- Roles:
 - aaa
 - admin
 - facility-manager
 - network
 - operations
 - read-only
 - server-compute
 - server-equipment
 - server-profile
 - server-security
 - storage
 - testrole
- Locales:

OK Cancel

LDAPプロバイダグループを参照する「All >> User Management >> Authentication >> Authentication Domains」でLDAP認証ドメイン(LDAP_DOMAIN)を設定し、LDAPユーザログインをテストします。

結論

このガイドでは基本的な導入シナリオを扱いますが、LDAP機能を詳しく調べることで、ディレクトリのパフォーマンスとセキュリティを大幅に向上させることができます。

追加情報、ベストプラクティス、および高度な設定の詳細については、指定のリソースを参照してください。

- [OpenLDAPの公式ドキュメント](#)
- [LDAPアカウントマネージャー手動](#)
- [389 Directory Serverに関するドキュメント](#)
- [UCS ManagerでのLDAPの設定](#)
- [UCS CシリーズサーバでのセキュアLDAPの設定](#)

翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。