在UCS Manager上配置LDAP &使用Linux OpenLDAP和389-DS服务器的CIMC

目录

简介

本文档介绍使用基于Linux的OpenLDAP和389目录服务器将LDAP配置为UCS Manager和CIMC的身份验证方法的各种选项。

背景信息

由于OpenLDAP服务器配置具有广泛的可变性，详尽的处理超出了本文档的范围。本文重点介绍跨多个Linux发行版、LDAP服务器包和属性架构的常用配置。为清晰和简单，本文档介绍标准LDAP配置。本文档不介绍安全LDAP(LDAPS)的配置。

先决条件:

强烈建议了解以下主题：

- UCS B系列
- UCS C系列
- Linux服务器管理

使用的组件

本文档中的信息基于以下软件和硬件版本：

- UCS Manager固件版本：4.3(2c)
- 交换矩阵互联型号：UCS-FI-6454
- UCS C系列独立服务器型号：UCSC-C240-M5
- UCS C系列独立固件版本：4.3(2.250045)
- Ubuntu 20.04
- CentOS流10

用于此演示的设置：

- LDAP服务器主机名：测试
- 服务器域：xxxxxxxxx.com
- 服务器FQDN:test.xxxxxxxxx.com
- Linux服务器（Ubuntu和CentOS）IP地址：X.X.X.19
- OpenLDAP用户：testuser1、testuser2

- OpenLDAP组：it
- OpenLDAP绑定用户帐户：bind_user

注意：本实验使用linux Nano文本编辑器。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

情形 1：乌班图 — 德比安

LDAP服务器配置可以使用图形界面（如LDAP帐户管理器）或命令行工具执行，具体取决于管理首选项和所需的控制级别。本场景使用基于Linux的OpenLDAP检查配置，从基于GUI的部署开始，然后过渡到命令行实用程序以探索高级功能，包括重叠插件（通常用于与Cisco UCS Manager的集成）。

选项 1：使用Ubuntu LDAP帐户管理器(LAM)配置OpenLDAP

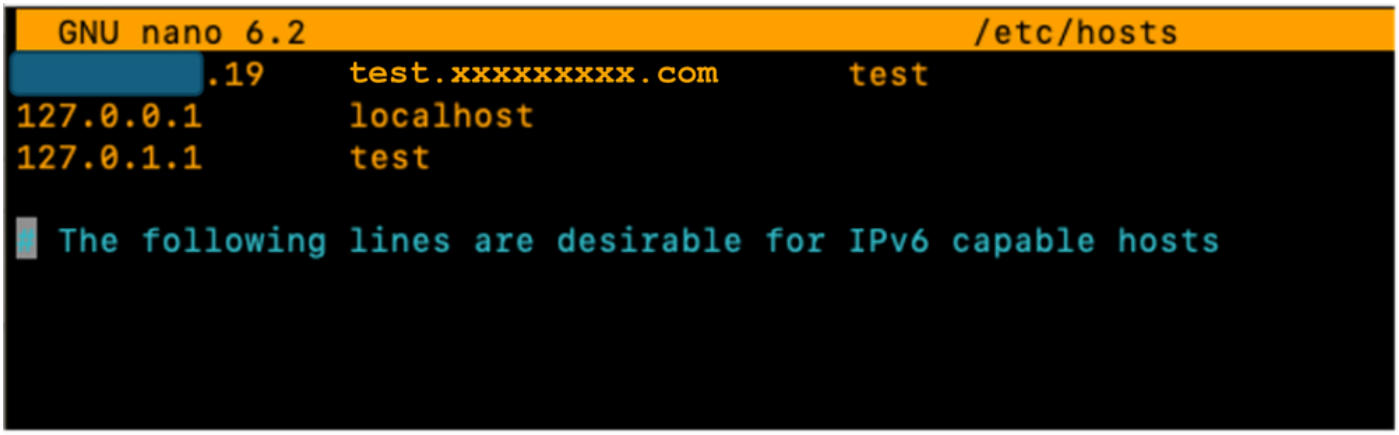
步骤 1：Linux服务器主机名和网络工具的初始配置。

更新ubuntu并安装net-tools软件包以访问ifconfig、netstat等工具：

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

使用“ifconfig”命令验证服务器IP地址，然后将其与服务器域名一起添加到“/etc/hosts”文件中(例如：“test.xxxxxxxxx.com”在本实验中使用)和主机名(例如：“test”)。

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



此外，请更新“/etc/hostname”文件，将其内容替换为主机名（测试）。

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

需要重新启动服务器才能使这些更改生效。

```
sudo reboot
```

第2步：安装SLAPD、Apache、PHP及其依赖关系

接下来，安装Apache、PHP及其依赖关系。这些用于启用通过网页的GUI交互：

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

安装Open LDAP服务器软件包“slapd”及其依赖项(ldap-utils)

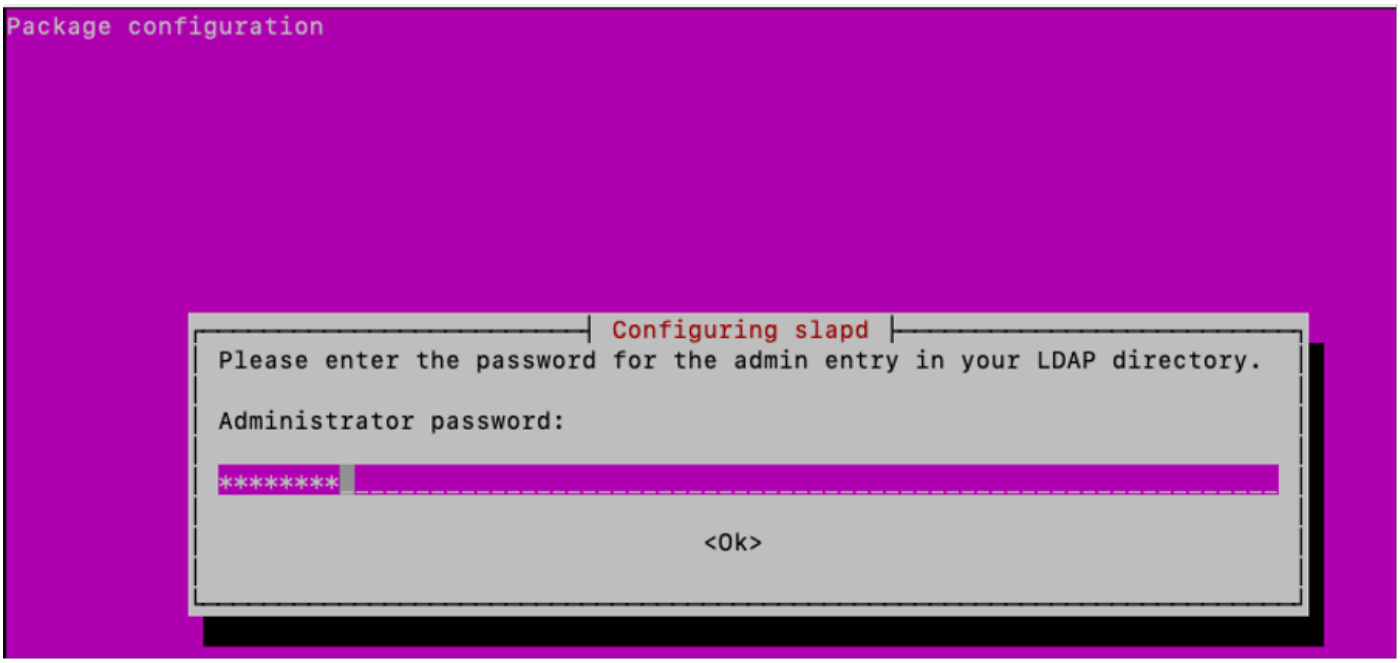
```
sudo apt install slapd ldap-utils -y
```

在slapd安装过程中，在显示的GUI弹出窗口中 — 输入额外所需的SLAPD软件包配置。

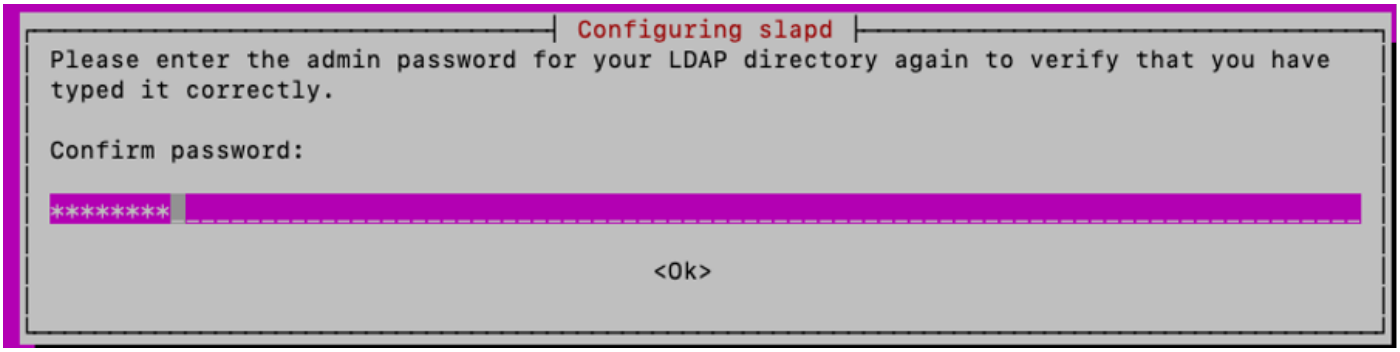
注意：丢失密码需要重新安装LDAP服务器。

此上下文中的“管理员”(admin)是用于管理OpenLDAP服务、模块和配置的帐户。

添加LDAP包“administrator”密码，然后按键盘上的Enter键选择“OK”。



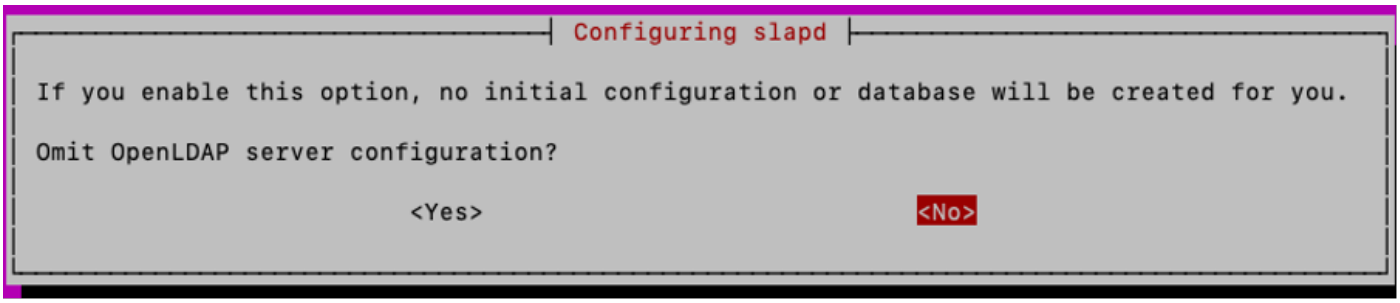
确认密码：



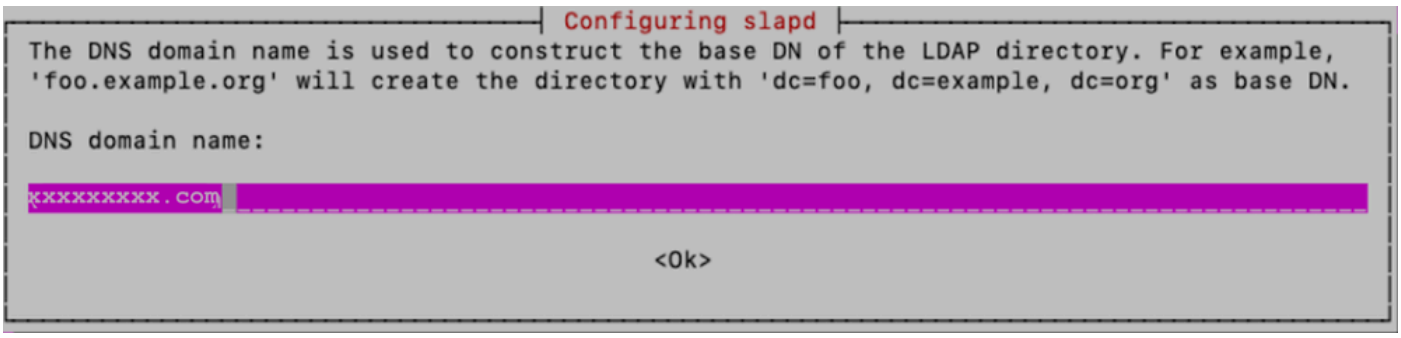
安装完成后，可以使用指定的命令重新配置SLAPD软件包，添加域信息：

```
sudo dpkg-reconfigure slapd
```

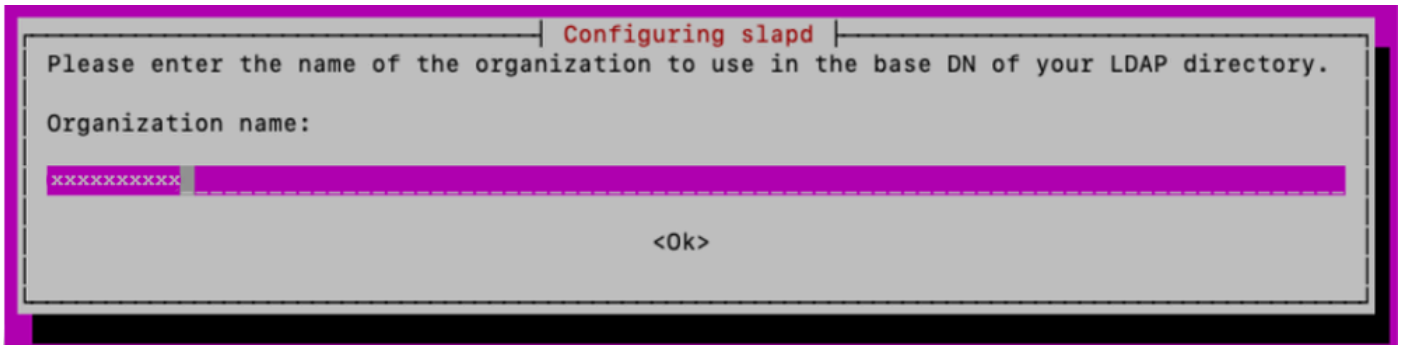
您可以接受“省略OpenLDAP服务器配置”的默认“否”选项，并按enter:



键入域名并按Enter键：



在本实验中，“xxxxxxxxx”用作“组织名称”：

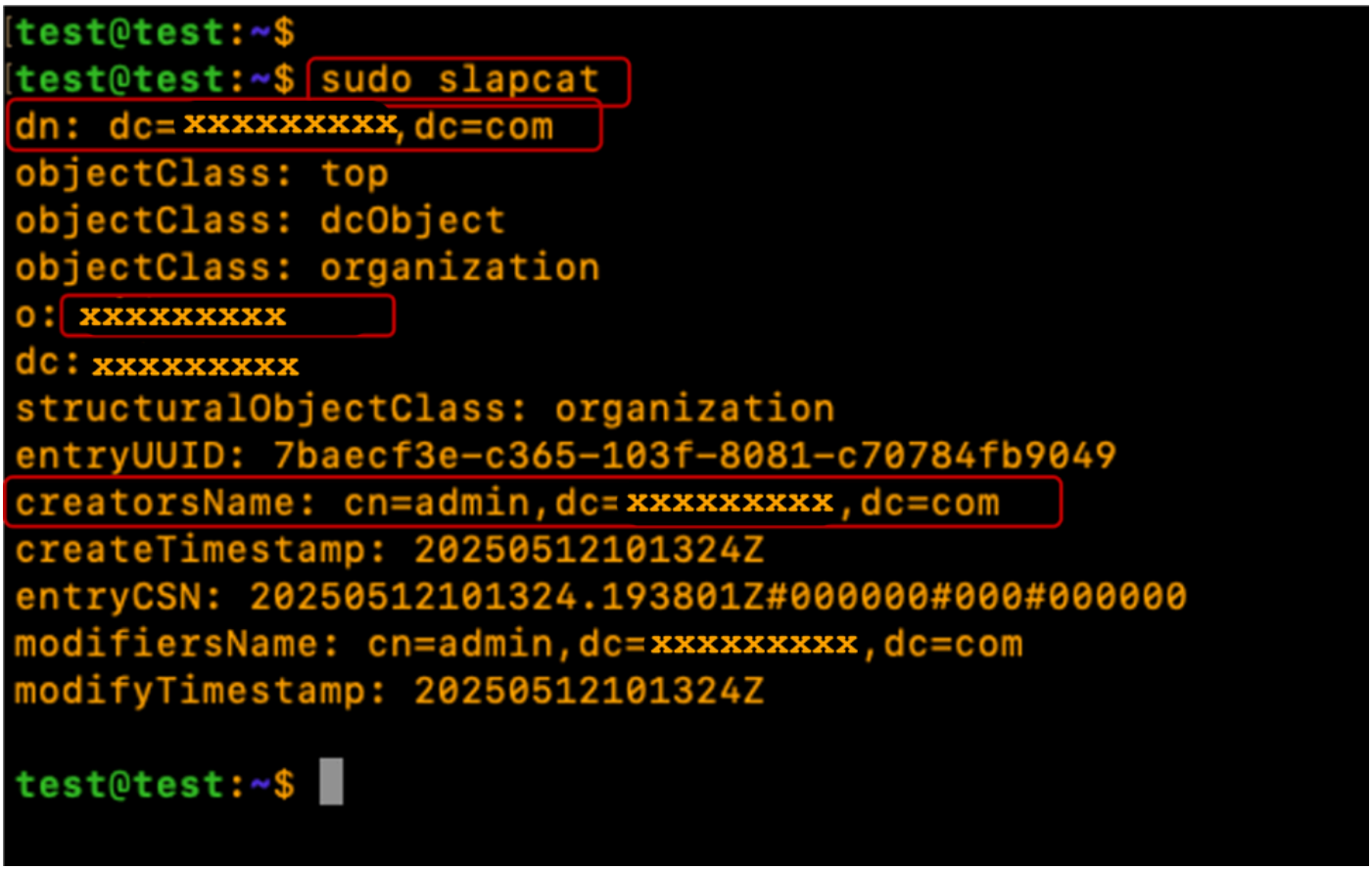


然后，键入“管理员密码”，确认

对于其他配置选项，保留默认值并按键盘上的Enter键完成配置。

使用命令验证SLAPD安装：

```
sudo slapcat
```



第3步：安装LDAP帐户管理器

安装LDAP帐户管理器(LAM)以创建和管理LDAP用户和组：

```
sudo apt -y install ldap-account-manager
```

启用LAM所需的PHP-CGI PHP扩展。

```
sudo a2enconf php*-cgi
```

重新加载Apache以激活新配置。

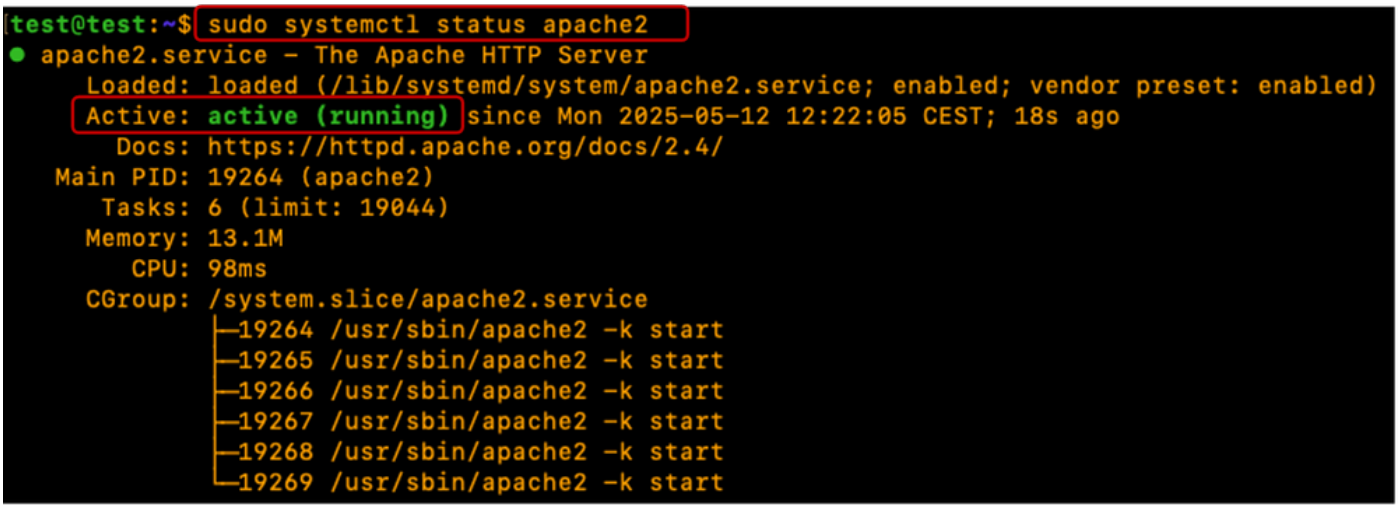
重新启动并启用Apache服务以在引导时自动启动：

```
sudo systemctl reload apache2
sudo systemctl restart apache2
```

```
sudo systemctl enable apache2
```

验证Apache服务器状态为“正在运行”和“活动”

```
sudo systemctl status apache2
```



配置Ubuntu防火墙以允许端口80(Web)、443（安全Web）、389(LDAP)和636(安全LDAP，如果需要可用)

```
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:~$
```

验证Ubuntu防火墙状态：

```
sudo ufw status
```

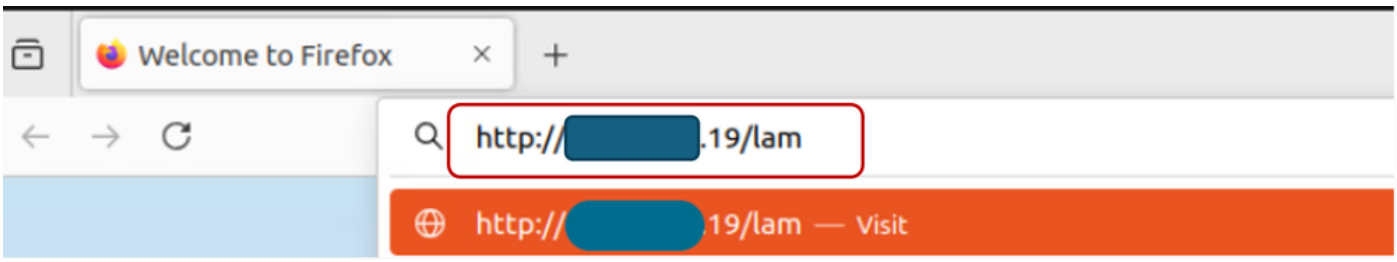
```
test@test:~$ sudo ufw status
Status: active

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)
```

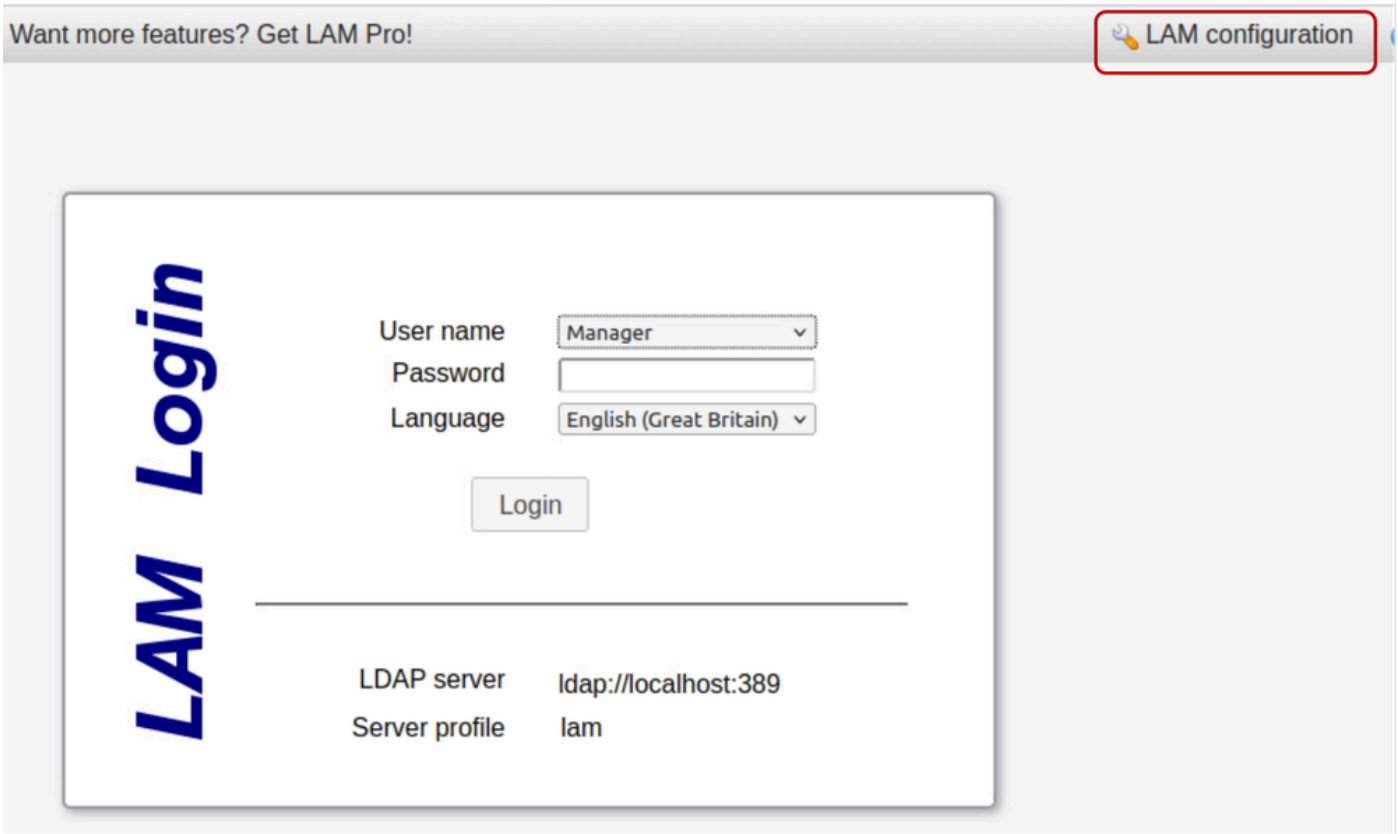
步骤 4： 配置LDAP帐户管理器

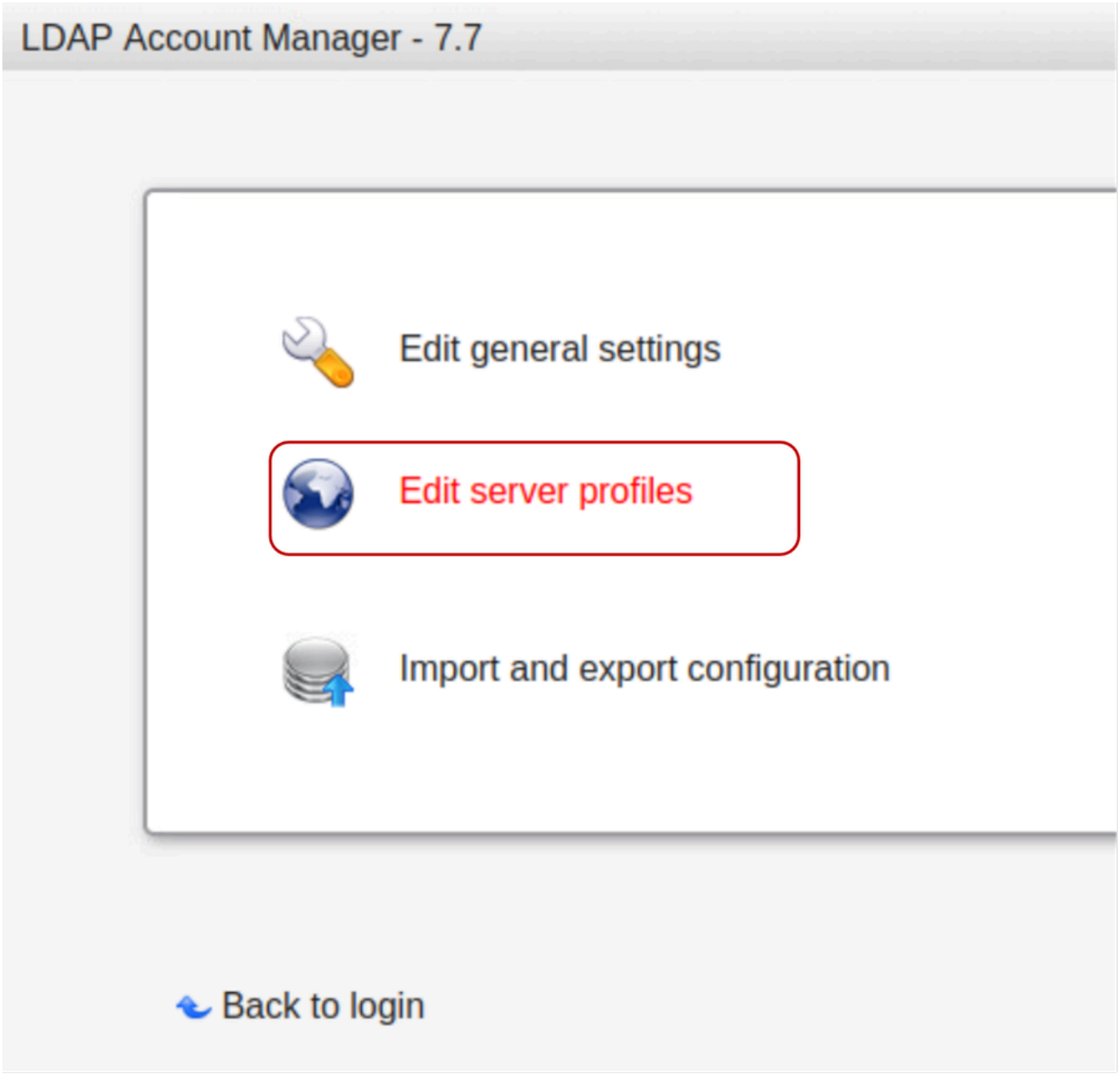
要从GUI配置LDAP帐户管理器(LAM)，请打开Web浏览器，输入Linux服务器IP地址并向其添加“lam”路径，如下所示：

```
http://X.X.X.19/lam
```

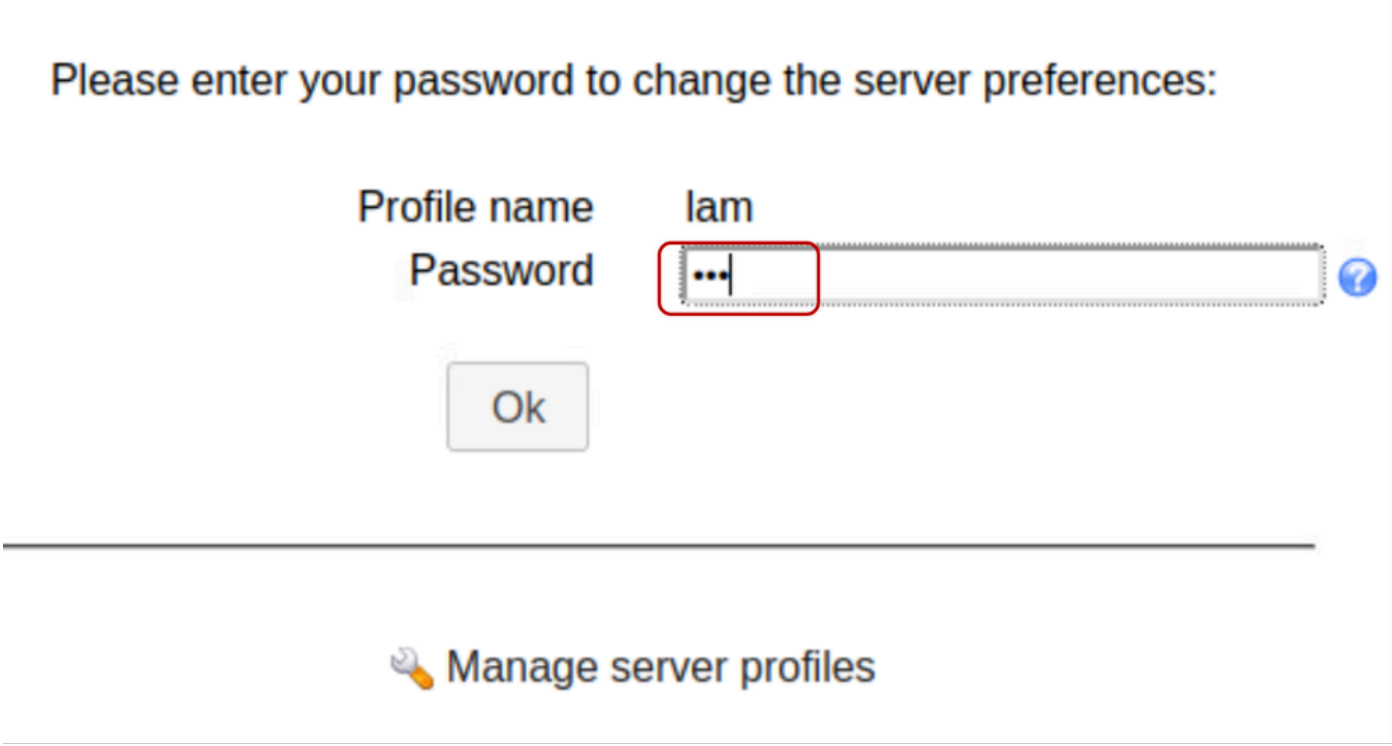


点击“LAM配置”，然后选择“编辑服务器配置文件”。





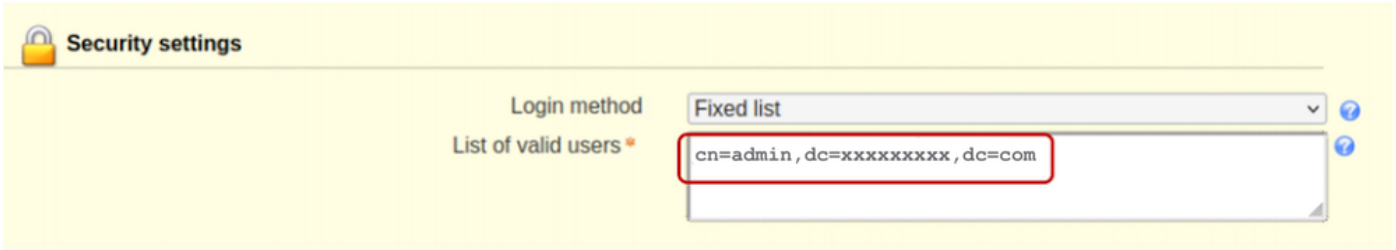
键入默认的lam密码“lam”进行登录。



在General Settings选项卡中，验证Server设置、“Language”和“Timezone”。

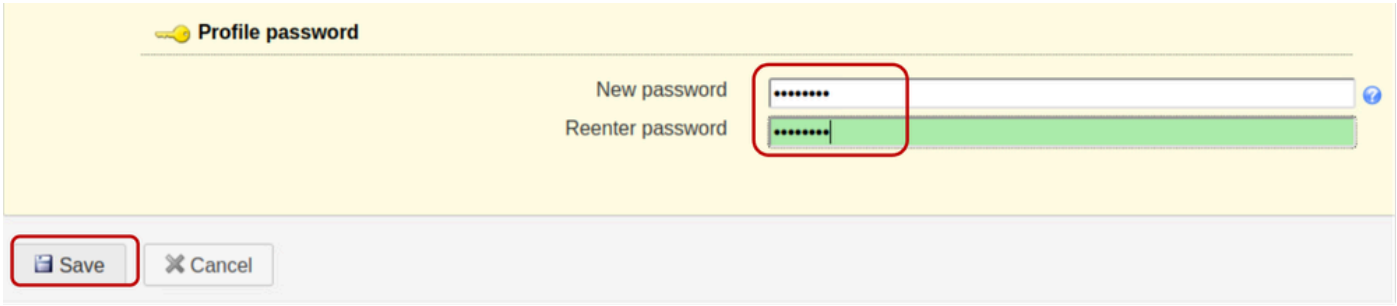
在“工具设置”(Tool settings)部分中，编辑并在“树后缀”(Tree suffix)字段中添加所需的域名，如下所示：

编辑Security settings部分以包含用于管理SLAPD服务的“admin”用户。



设置“配置文件密码”。此密码用于后续登录LAM配置接口，例如，配置“cisco123”而不是默认“lam”密码。

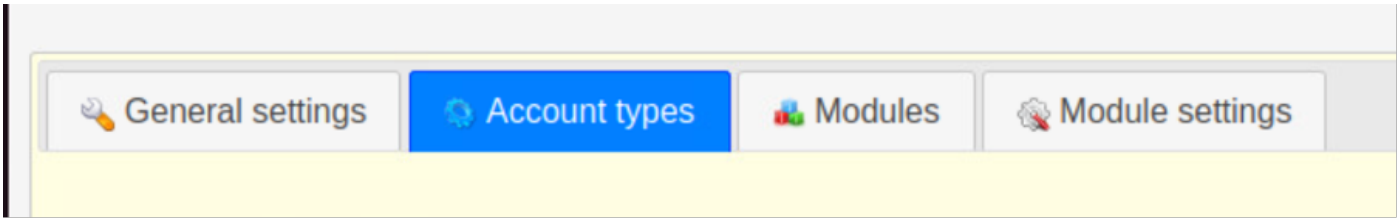
保存配置:



然后，在LAM配置GUI界面上重新启动会话。

使用创建的新密码重新登录（LAM配置>>编辑服务器配置文件）。

点击“Account types”，



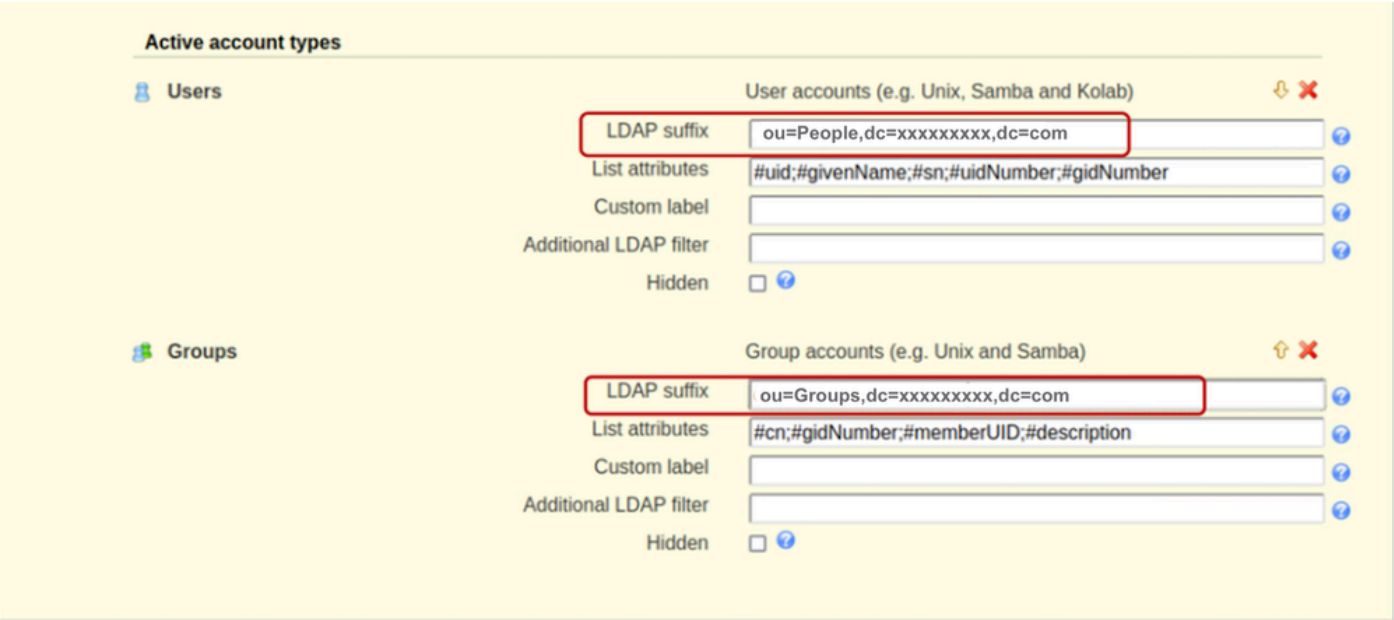
向下滚动并编辑LDAP后缀字段中包含域名信息的默认活动帐户类型。例如，“LDAP后缀”字段的默认内容显示值为“ou=People，dc=my-domain，dc=com”。

如果需要创建新的组织单位，请替换“LDAP后缀”字段的内容以包含组织单位的名称。

格式显示为“ou=<organizational_unit>,dc=xxxxxxxxx，dc=com”。

在本演示中，用户的OU是“People”，组的OU是“Groups”。

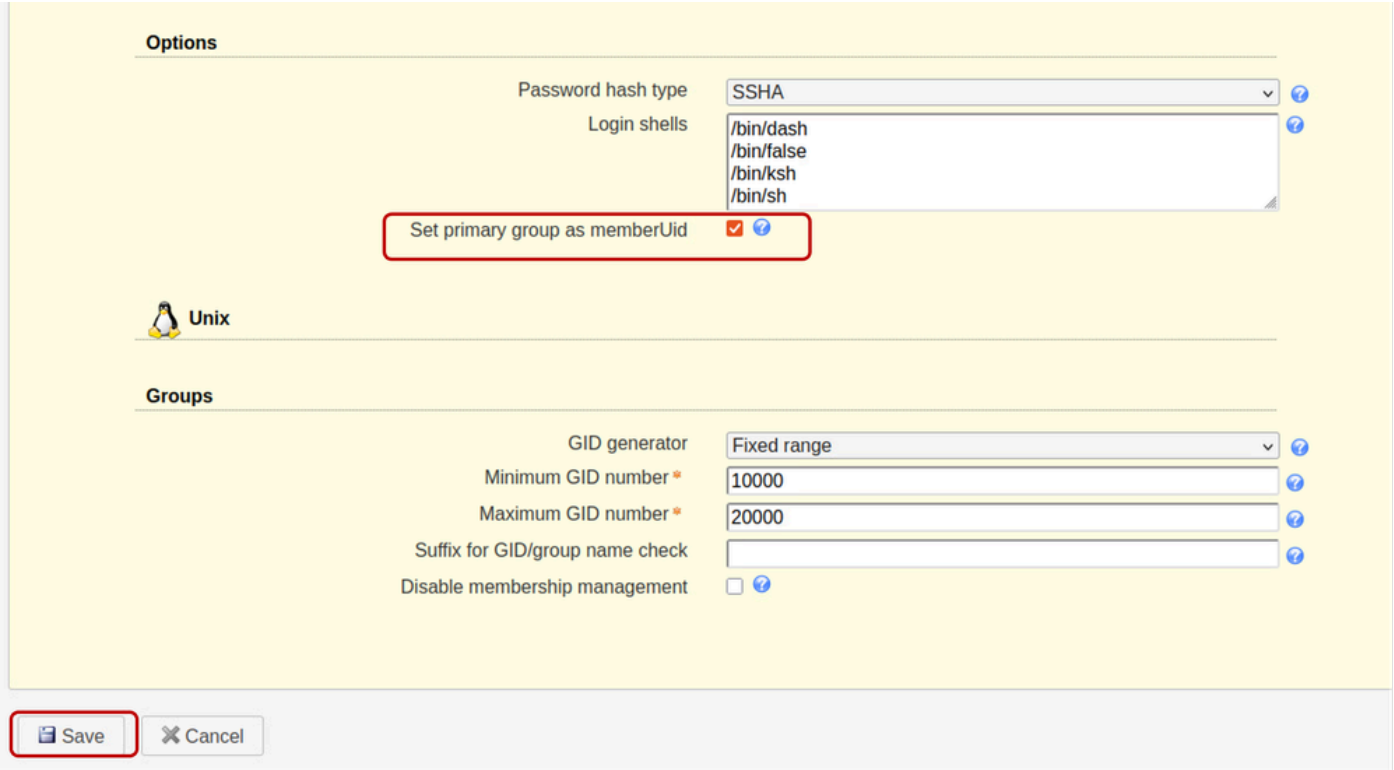
保存配置。



向下滚动到“选项”部分，确保选中“将主组设置为memberUid”。

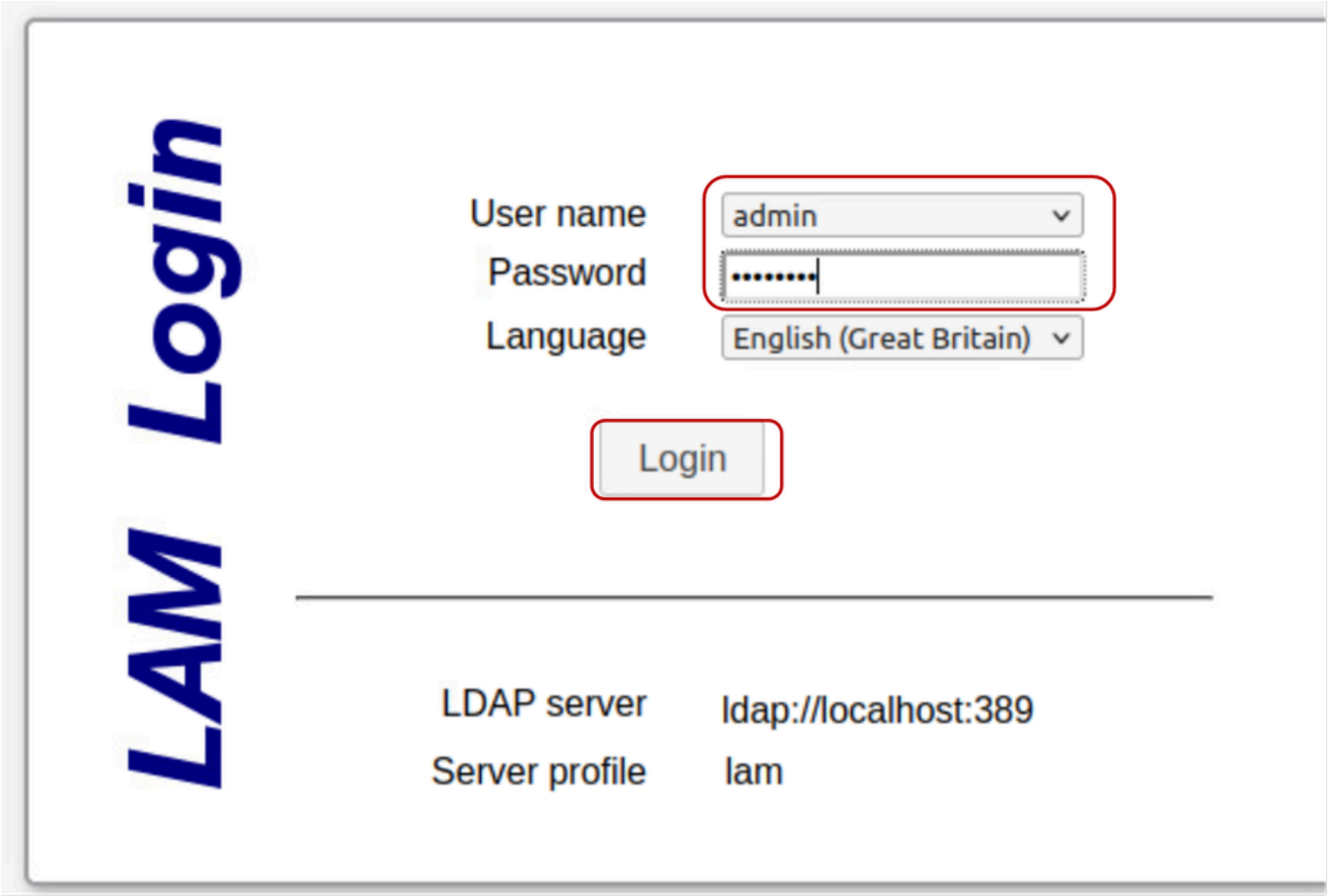
默认情况下在组对象上不设置“将主组设置为memberUid”选项。通过激活此项，可以将OpenLDAP“Primary group”用作标准LDAP组，在该组中可以引用“memberUid”(例如：在UCS C系列服务器配置中)。 如果未选中此选项，则属于任何主组的用户的登录都将失败。

保存配置。

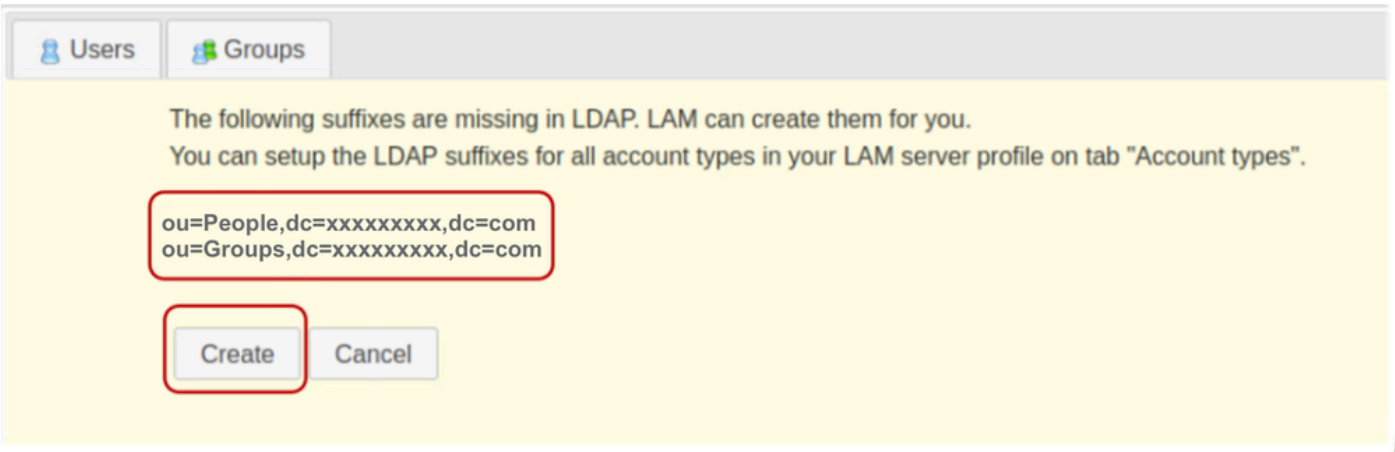


第5步：创建OU、组和用户

以“admin”用户身份登录LAM，使用与安装期间创建的密码相同的密码，以分别创建属于先前创建的OU(People和Groups)的Users和Groups:

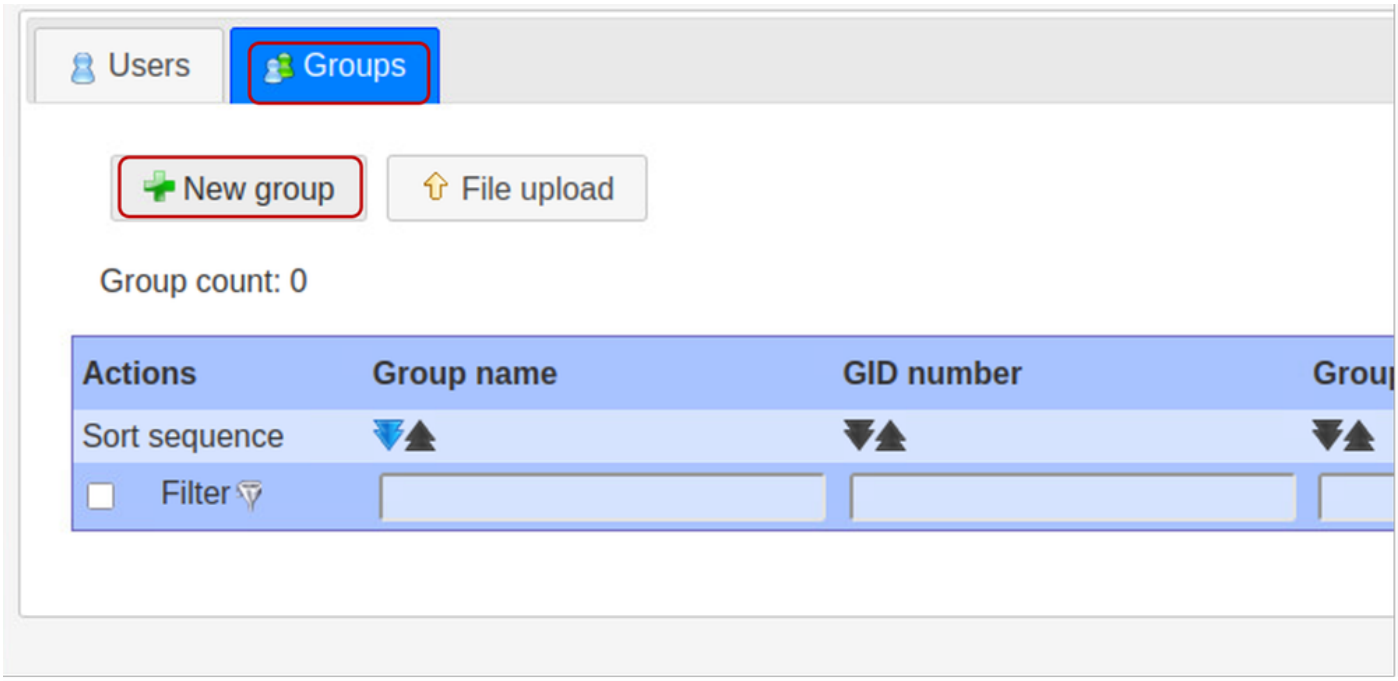


在LAM Configuration部分创建较早指定的OU。
点击Create。



然后，在LDAP帐户管理器中创建“it”组：

选择“组”选项卡，然后单击“新建组”

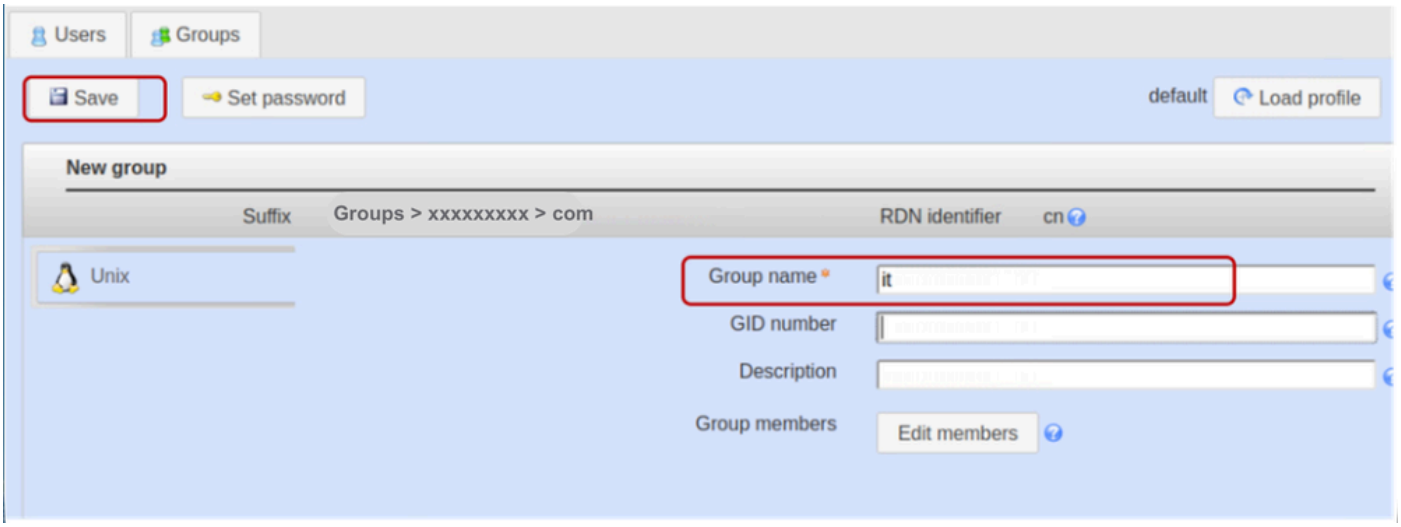


将组名称设置为“it”。

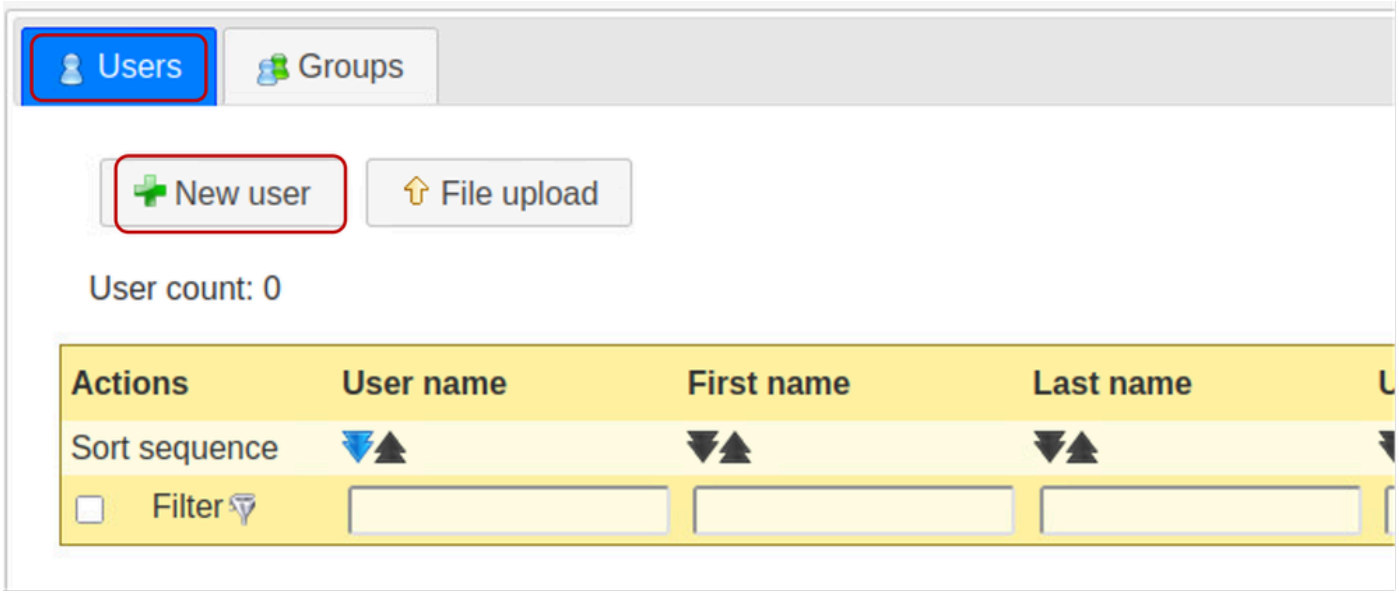
注意：虽然Cisco UCS系统通常可适应各种大小写变化，但保持小写命名约定是确保不同LDAP服务器基础设施环境之间长期互操作性的最佳实践。

将GID编号字段留空。LDAP客户经理(LAM)旨在使用下一个可用值自动填充此字段。

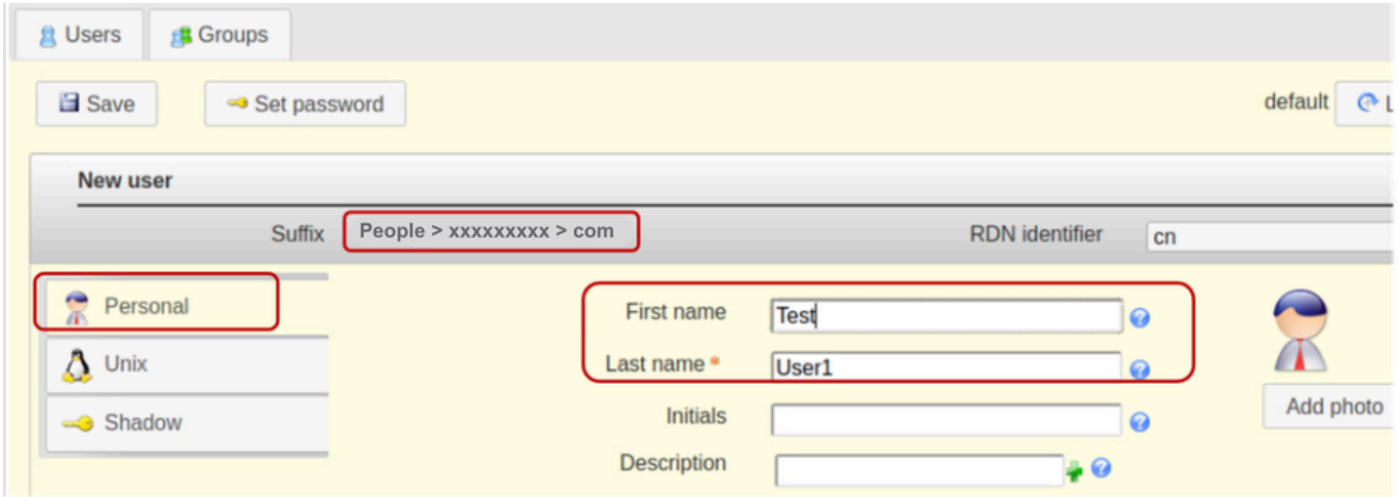
如有需要，提供说明并点击Save



点击“Users”(用户)选项卡以创建用户帐户，然后选择“New user”（新用户）。



在Personal选项卡中填充“testuser1”用户的必填字段。

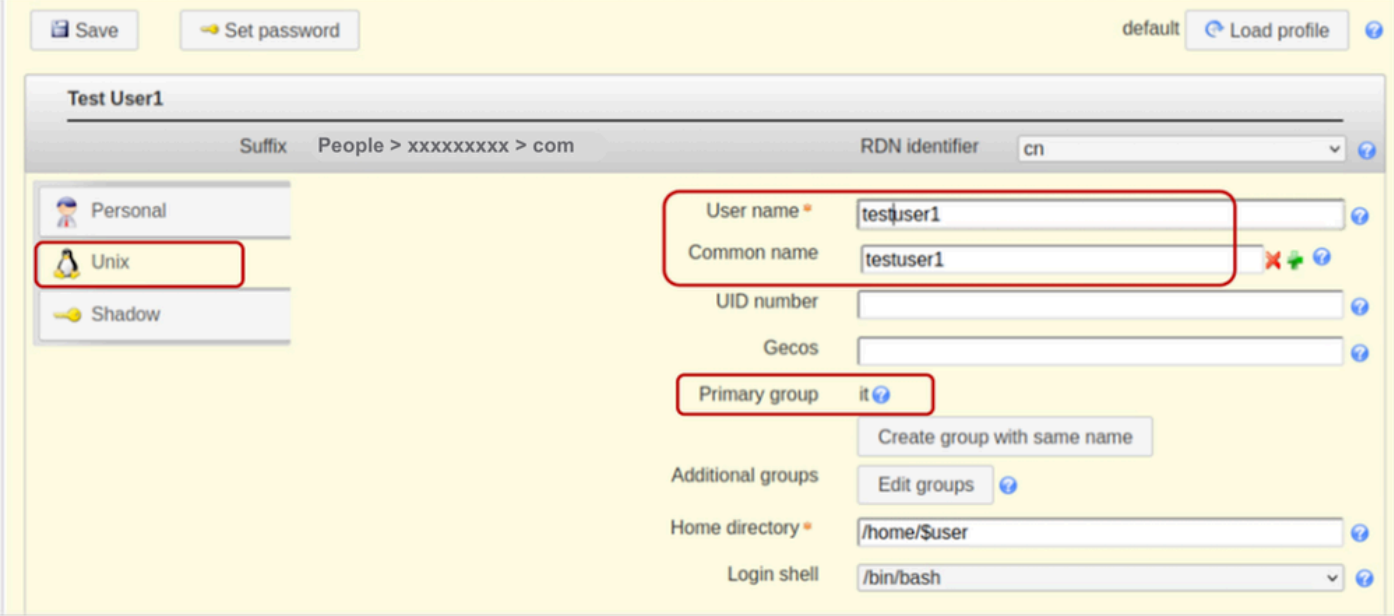


选择Unix选项卡，在User name字段中添加testuser1。在“it”组中包括用户。

对于此演示，只有“it”组存在，因此它已预填充。

保留RDN标识符作为“公用名”(cn)。 这使系统能够使用“用户名”字段中指定的值自动填充“公用名”字段。

将UID编号字段留空，因为LAM会自动使用可用值填充该字段。



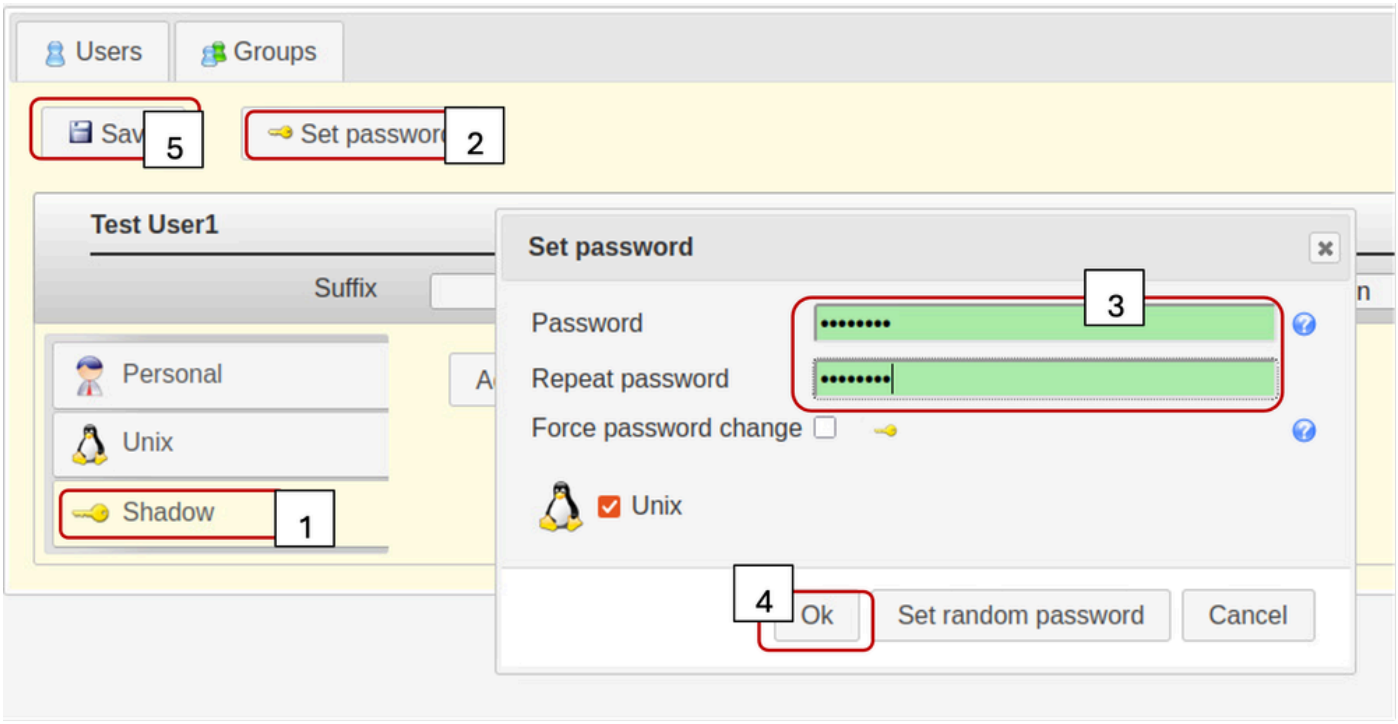
选择Shadow选项卡，

不使用影子帐户扩展。

点击“设置密码”。

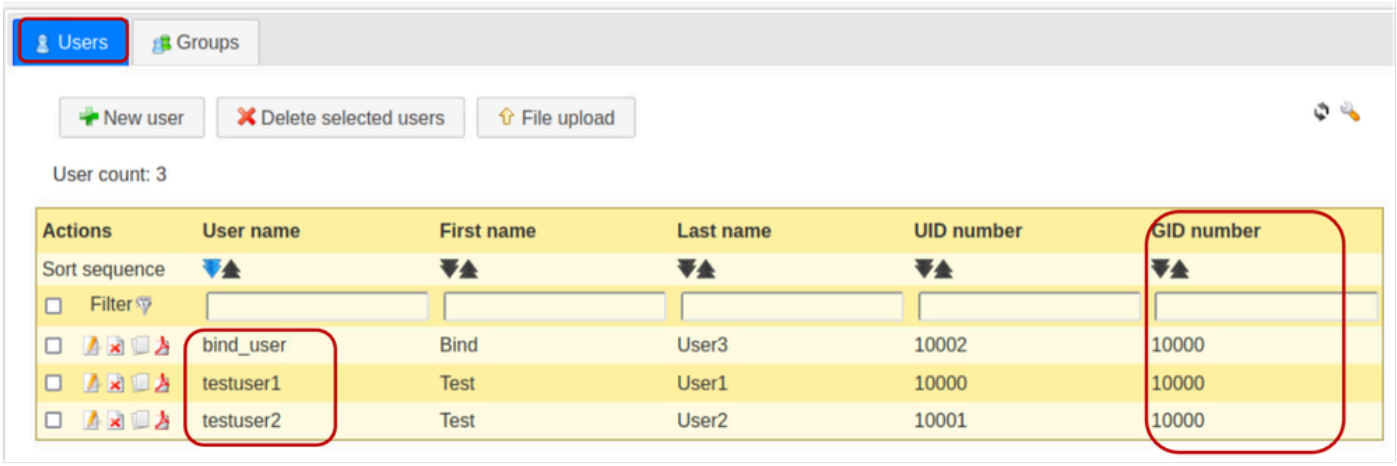
设置用户密码

点击OK并保存



重复前面所述的指定步骤，以创建“testuser2”用户帐户和“bind_user”帐户。

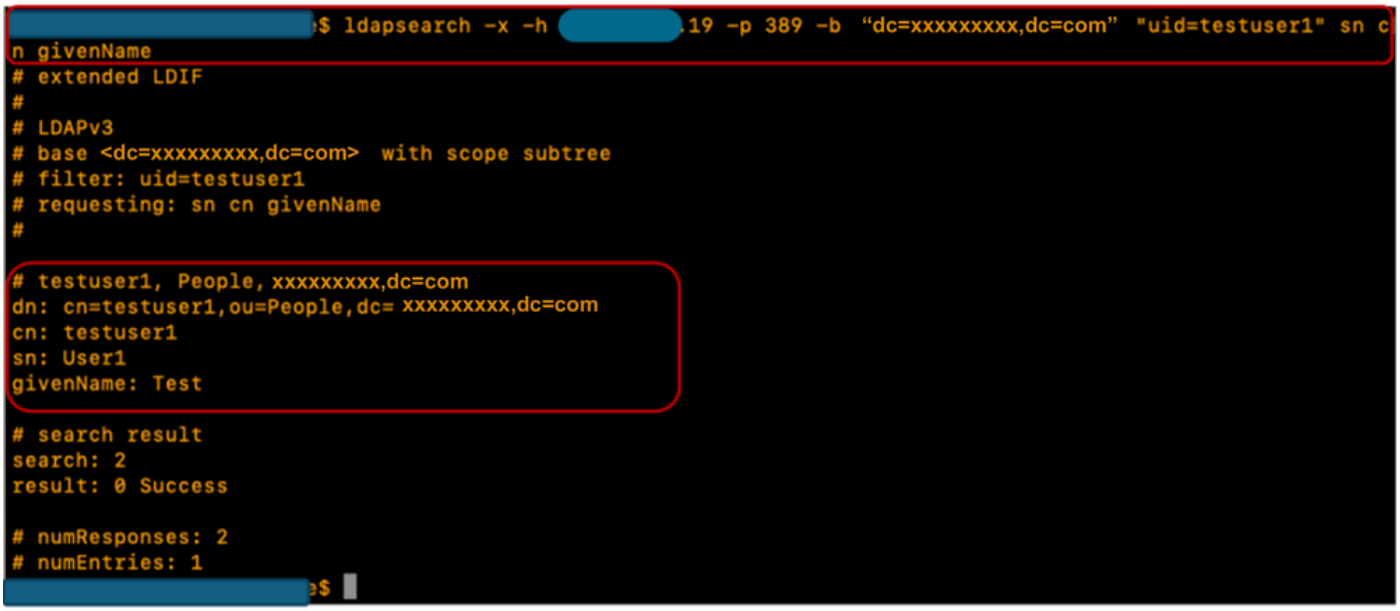
点击“用户”选项卡以验证所有所需用户的创建。（在gidNumber列中具有相同的值可确认创建的用户属于同一组 — 它）



第6步：测试本地LDAP登录

登录到另一个基于Linux的系统，可以访问OpenLDAP服务器。
运行指定的ldapsearch命令以验证LDAP是否正常工作：

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



CIMC上的配置参数

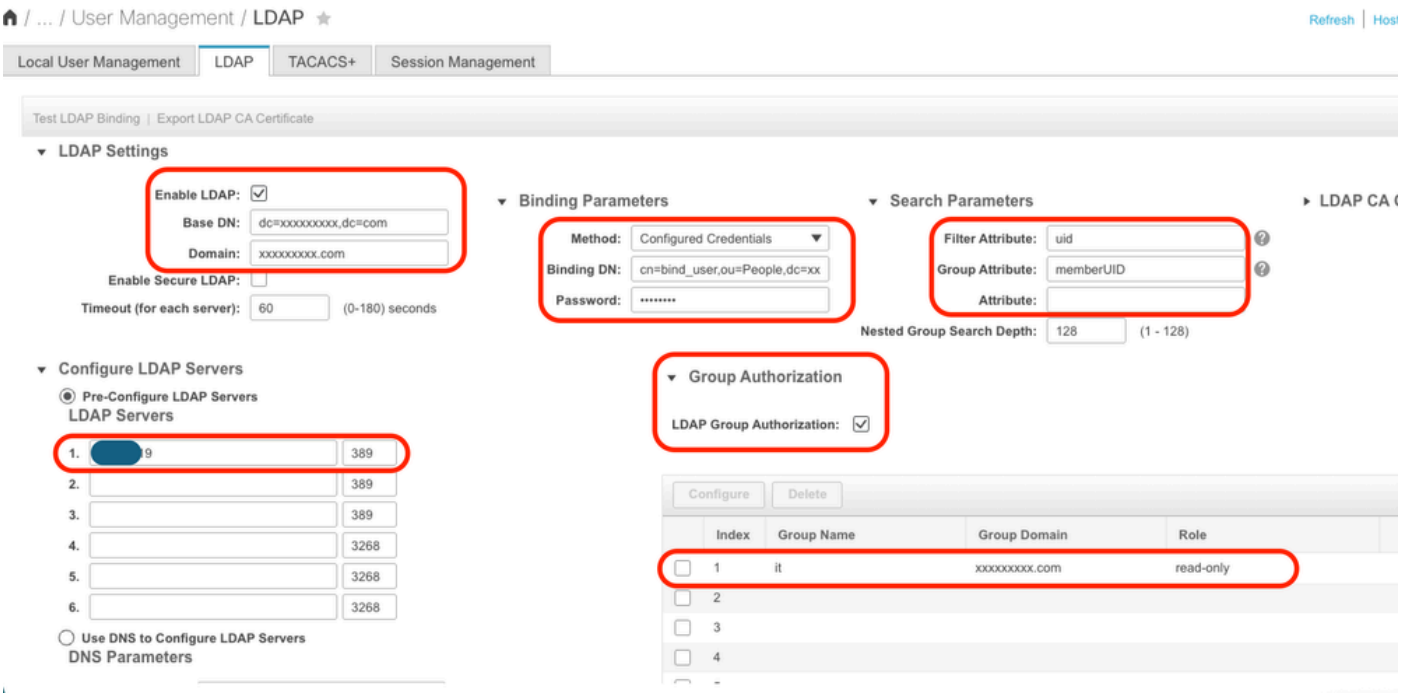
登录CIMC。

在“导航”(Navigation)窗格中，依次选择管理(Admin)、用户管理(User Management)和LDAP。

如下所示填充LDAP配置参数：

- 启用LDAP：选中
- 基准 DN:dc=xxxxxxxxx，dc=com
- 域名：xxxxxxxxx.com
- LDAP 服务器:<ldap_server_IP或FQDN> X.X.X.19
- 绑定参数：“登录凭证”或“配置的凭证”
  - 使用配置的凭证时，请完全按照在LDAP服务器上配置的步骤添加bind_user DN:
    - 例如：cn=bind_user，ou=People，dc=xxxxxxxxx，dc=com
- 搜索参数：
  - 过滤器属性："cn"或"uid"
  - 组属性：memberUID
- LDAP组授权 — 已选中
  - 组名称:it

- 组域：xxxxxxxxx.com
- 角色：只读（任何所需的角色）



保存配置并测试LDAP用户登录。

UCS Manager上的配置参数

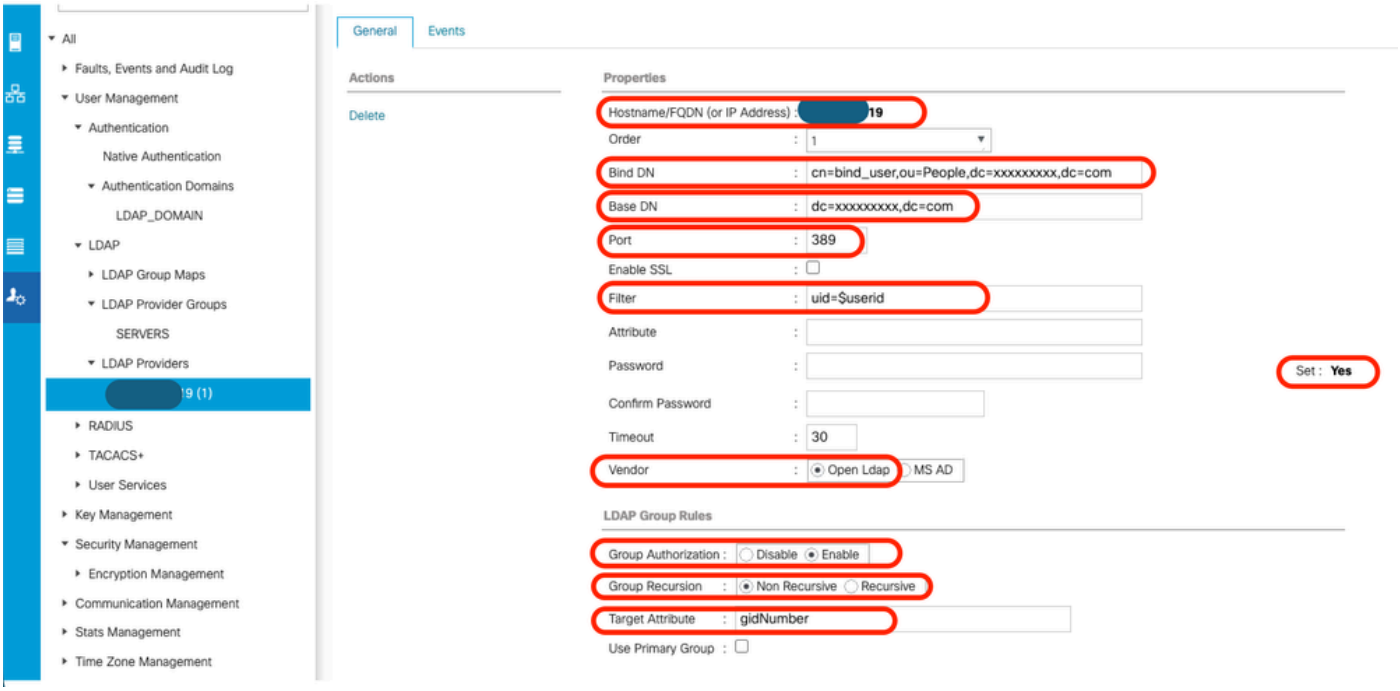
登录UCS Manager。

在“导航”(Navigation)窗格中，依次选择管理(Admin)、用户管理(User Management)和LDAP。

如下所示填充LDAP配置参数：

- LDAP提供程序：
  - 主机名：<LDAP服务器的FQDN或IP地址>
  - 绑定DN:cn=bind_user，ou=People，dc=xxxxxxxxx，dc=com
  - 基准 DN:dc=xxxxxxxxx，dc=com
  - 端口：389
  - 启用 SSL:禁用
  - 过滤器：uid=$userid
  - 组授权：启用
  - 组递归：非递归
  - 目标属性：gidNumber
- LDAP组映射：

- LDAP组DN:10000 <it"组的gidNumber>

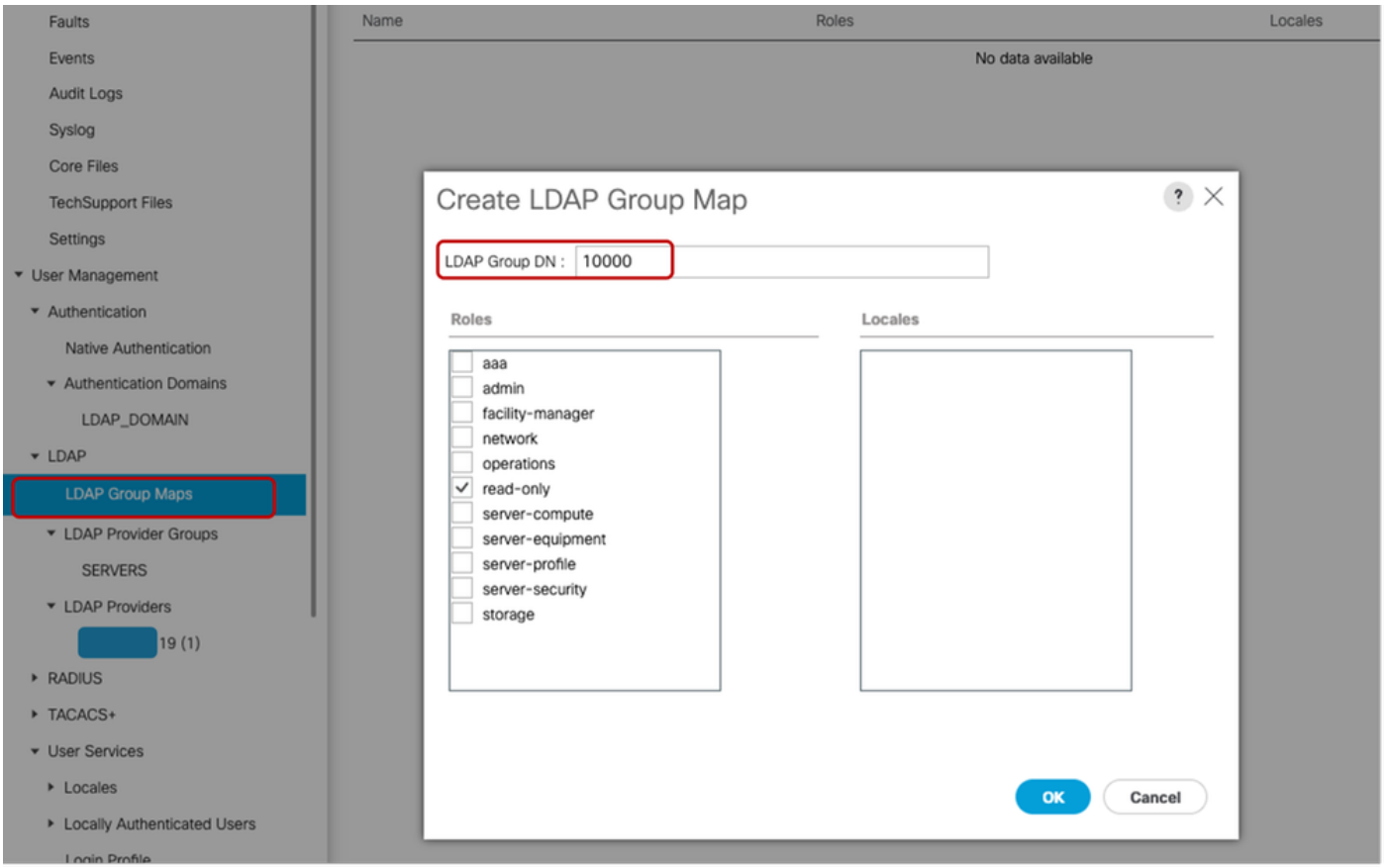


在All >> User Management >> LDAP >> LDAP Providers>> LDAP Group Rules下，UCS Manager的默认目标属性为“memberOf”。默认情况下，OpenLDAP服务器未启用该属性，因此将“目标属性”值设置为“memberOf”（或将其留空）会导致用户登录失败，因为OpenLDAP服务器无法识别所请求的属性值。

在本示例中，“Target Attribute”值已设置为“gidNumber”。

将已配置的LDAP提供程序添加到LDAP提供程序组。在本演示中，已创建“SERVERS” LDAP提供程序组。

在“All >> User Management >> LDAP >> LDAP Group Maps>>”中配置“LDAP组映射”时，gidNumber值(在本例中为“10000”)将用作“组DN映射”，如下所示：



在引用LDAP提供程序组的“All >> User Management >> Authentication >> Authentication Domains”中配置LDAP身份验证域(LDAP_DOMAIN),并测试LDAP用户登录。

注意：如果要求memberOf属性满足特定的环境要求或实施“组递归”功能，建议使用以下第二个配置选项，该选项要求启用重叠扩展的LDAP。

虽然LDAP帐户管理器(LAM)支持重叠配置，但请注意，此功能需要适当的许可。

有关使用LAM配置LDAP的详细信息，请参阅官方LDAP帐户管理器文档。

选项 2：使用Ubuntu CLI工具和重叠配置OpenLDAP

要将OpenLDAP用于UCS Manager身份验证，需要两个重叠，以确保组以UCS系统（UCS Manager和CIMC）可以理解的方式与用户关联。

OpenLDAP端的配置要求：

- “memberof”重叠：此重叠会在用户和组之间创建映射，以便查询用户DN时，可在该查询中请求memberOf属性。默认情况下，除非将重叠成员添加到openLDAP，否则组成员资格的用户

没有属性

- “refint”重叠：此覆盖配置为验证组对象中成员属性中的条目是否仍与用户对象的memberOf属性保持同步。如果没有此服务，如果删除用户时未同时修改组，则孤立DN可以保留在组对象中。精简服务可确保两个方向的一致性。

第1步：初始网络工具和配置Linux服务器主机名

在选项1中重复步骤1。

第2步：安装SLAPD

在选项1中重复步骤2。（除了选项2中的PHP和Apache安装之外，它们不需要工作 — 无LAM）

确保允许所需端口通过Ubuntu防火墙。

步骤 3：在LDAP服务器上安装“memberOf”重叠

检查是否已安装“memberOf”重叠

```
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
```

要安装“memberOf”重叠，请创建名为ldap.memberof.load.ldif的.ldif文件（使用任何所需的命名约定）并添加指定的配置：

```
cat <

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

使用指定的命令将ldap.memberof.load.ldif文件中的配置添加到LDAP配置文件：

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

根据Linux发行版配置memberOf模块和olcDatabase条目以匹配部署要求。

两个强制属性值是“olcDatabase={1}mdb”和“groupOfNames”，如下所示。

创建ldap.memberof.config.ldif文件，填充其属性并将其内容导入到LDAP配置文件中。

```
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
```

步骤 4：在LDAP服务器上安装“精简版”重叠

接下来，安装精简到openldap:

创建名为ldap.refint.load.ldif的.ldif文件（使用任何所需的命名约定）并添加指定的配置：

```
cat <


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

使用指定的命令将ldap.refint.load.ldif文件中的配置导入LDAP配置文件：

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

配置refint，以维护组和用户之间的引用完整性。

配置精简模块及其olcDatabase条目以匹配部署要求。

创建ldap.refint.config.ldif文件并将其内容导入到LDAP配置文件中。

```
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
```

安装两个插件/扩展插件时，指定的ldapsearch命令的输出与下面显示的输出类似：

```
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
```

当两个插件/扩展都已配置时，指定的ldapsearch命令的输出与显示的输出类似：

```
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
```

重新启动slapd服务，使新安装的插件/模块可用：

```
sudo systemctl restart slapd
```

步骤 5：创建OU、用户和组

创建组织单位（用于用户和组）、用户和组。

创建用户（人员）和组（组）OU并将其导入到LDAP配置文件中。这需要“admin”帐户密码：

```
cat <


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

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

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

创建用户（testuser1、testuser2和bind_user），将它们映射到各自的OU(People)，使用gidNumbers将它们添加到其组（良好做法），并将用户导入到LDAP配置文件中。

```
cat <



      ./ldap.users.ldif
dn: uid=testuser1,ou=People,dc=xxxxxxxxx,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=xxxxxxxxx,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=xxxxxxxxx,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 -xcWD cn=admin,dc=xxxxxxxxx,dc=com -f ./ldap.users.ldif
```

```
test@test:~$ cat <<EOF > ./ldap.users.ldif
dn: uid=testuser1,ou=People,dc=xxxxxxxxx,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=xxxxxxxxx,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=xxxxxxxxx,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 -xcWD  cn=admin,dc=xxxxxxxxx,dc=com -f ./ldap.users.ldif
Enter LDAP Password:
adding new entry "uid=testuser1,ou=People,dc=xxxxxxxxx,dc=com

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

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

test@test:~$
```

创建组(it)，将它们映射到各自的OU（组），关联组成员(testuser1、testuser2)，然后将它们导入到LDAP配置文件中：

```
cat <


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

```
sudo ldapadd -xcWD cn=admin,dc=xxxxxxxxx,dc=com -f ./ldap.group.create.ldif
```

注意：即使memberOf属性在创建“用户”或“组”期间未显式定义，系统也会自动生成并维护此引用。一旦用户与组相关联，memberOf属性将自动反映这些成员身份，以确保目录保持与当前访问结构同步。

第6步：测试本地LDAP登录

使用指定的命令验证用户是否登录到LDAP服务器（根据您的环境替换登录参数）：

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

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

test@test:~$
```

CIMC上的配置参数

登录CIMC。

在“导航”(Navigation)窗格中，依次选择管理(Admin)、用户管理(User Management)和LDAP。

如下所示填充LDAP配置参数：

- 启用LDAP：选中
- 基准 DN:dc=xxxxxxxxx，dc=com

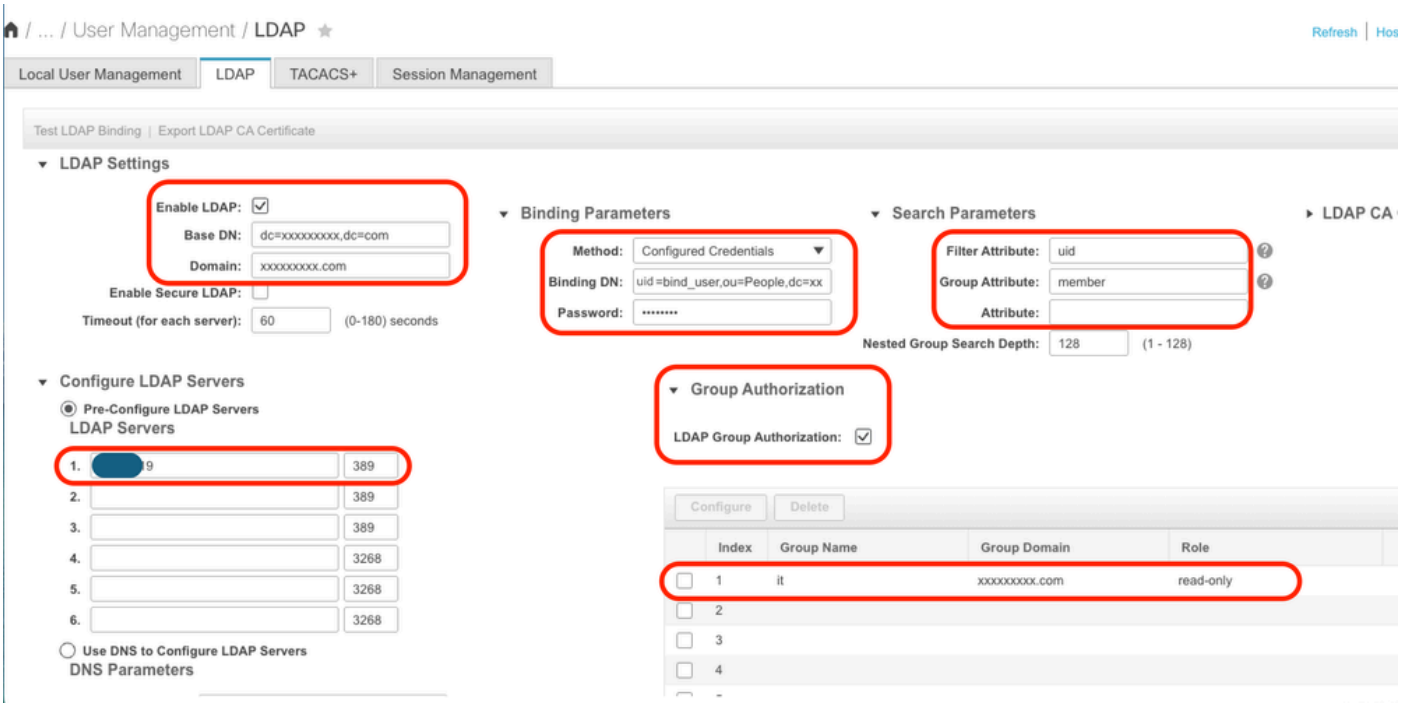
- 域名：xxxxxxxxx.com

- LDAP服务器：<ldap_server_IP或FQDN> X.X.X.19

- 绑定参数：可以是“登录凭证”或“配置的凭证”
  - 使用配置的凭证时，请完全按照在LDAP服务器上配置的步骤添加bind_user DN:
  - 例如："cn=bind_user，ou=People，dc=xxxxxxxxx，dc=com"或 "uid=bind_user，ou=People，dc=xxxxxxxxx，dc=com"

- 搜索参数：
  - 过滤器属性："cn"或"uid"
  - 组属性：成员

- LDAP组授权 — 已选中
  - 组名称:it
  - 组域：xxxxxxxxx.com
  - 角色：只读（任何首选角色）



保存配置并测试LDAP用户登录。

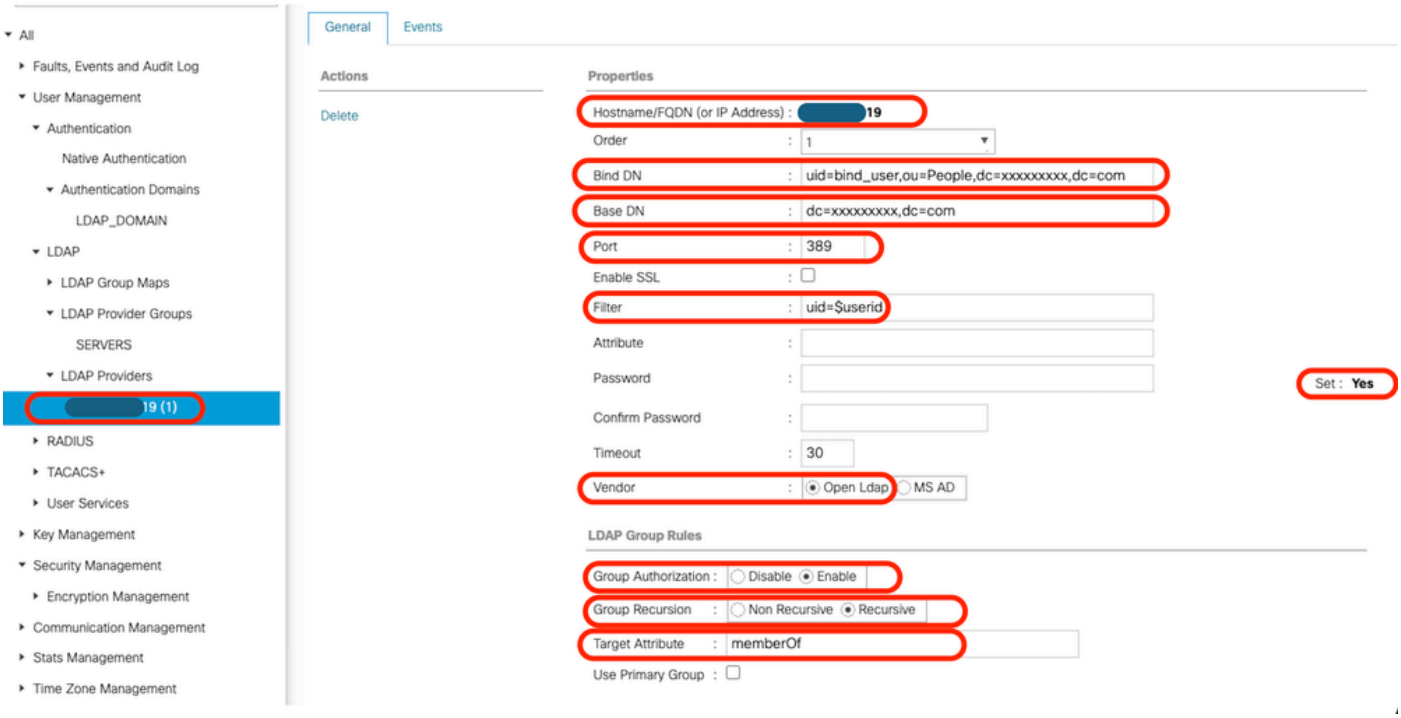
UCS Manager上的配置参数

登录UCS Manager。

在“导航”(Navigation)窗格中，依次选择管理(Admin)、用户管理(User Management)和LDAP。

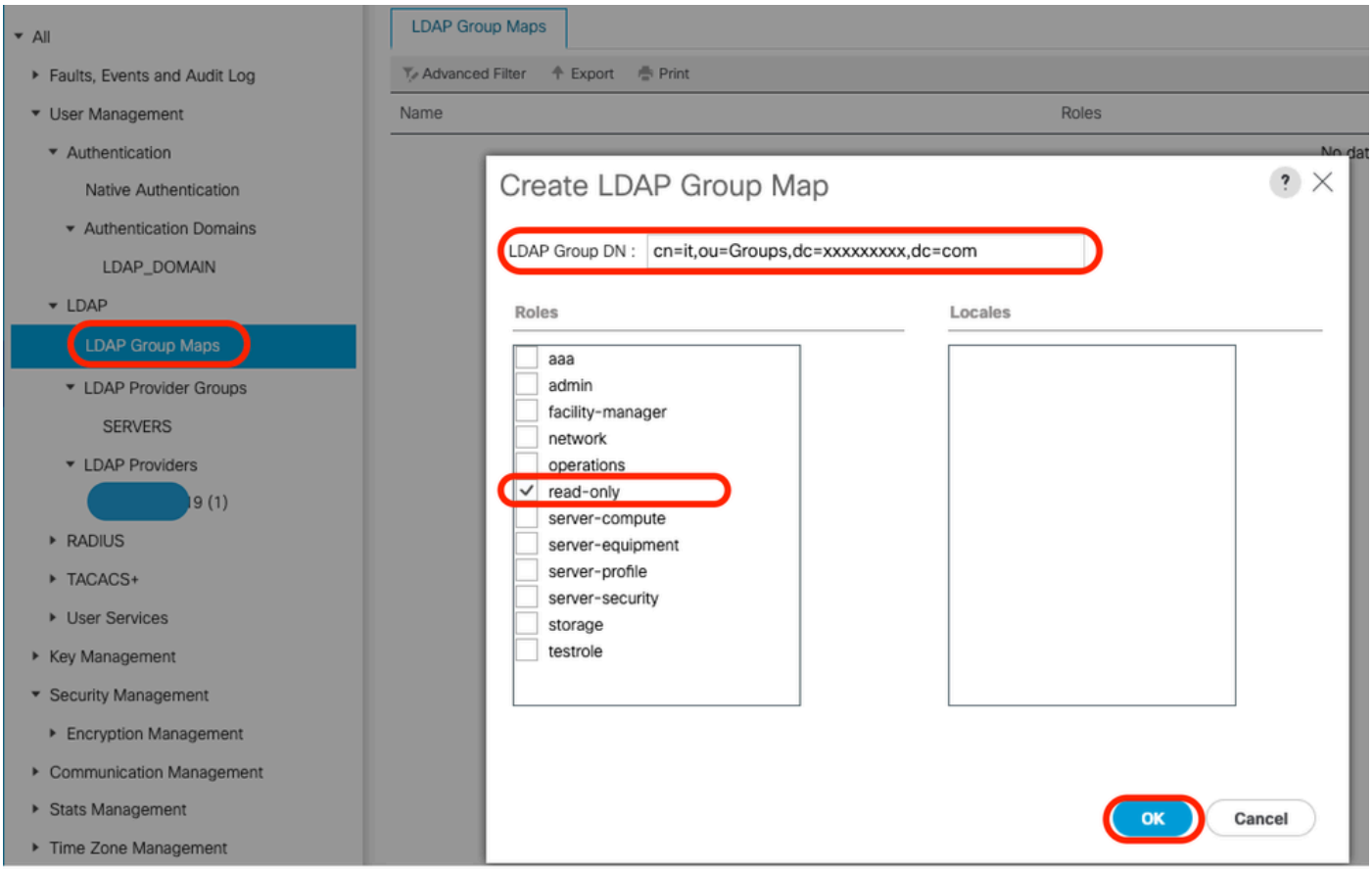
如下所示填充LDAP配置参数：

- LDAP提供程序：
  - 主机名：<LDAP服务器的FQDN或IP地址>
  - 绑定DN:uid=bind_user，ou=People，dc=xxxxxxxxx，dc=com
  - 基准 DN:dc=xxxxxxxxx，dc=com
  - 端口：389
  - 启用 SSL:禁用
  - 过滤器：uid=$userid
  - 组授权：启用
  - 组递归：递归
  - 目标属性：成员
- LDAP组映射：
  - LDAP组DN:cn=it，ou=Groups，dc=xxxxxxxxx，dc=com



将已配置的LDAP提供程序添加到LDAP提供程序组。在本演示中，使用“SERVERS”LDAP提供程序组。

配置LDAP组映射，添加从LDAP服务器检索的“LDAP组DN”。



在引用LDAP提供程序组(SERVERS)的“All >> User Management >> Authentication >> Authentication Domains”中配置LDAP身份验证域(LDAP_DOMAIN)，并测试LDAP用户登录。

接下来，我们来了解一下在单独的Linux发行版(CentOS 10)中设置相同配置（使用重叠）

方案 2：CentOS流10 - Fedora

轻量级目录访问协议(LDAP)的配置过程因底层操作系统版本而异。本节重点介绍在CentOS Stream 10上实施LDAP。

虽然许多Linux发行版都使用OpenLDAP，但是CentOS Stream 10和基于Fedora的当代系统都使用389目录服务器(389 DS)作为默认LDAP提供程序。

注意：虽然389 DS被认为是CentOS和Red Hat生态系统中OpenLDAP的后继者，但是这两种解决方案不能直接互换。其各自的目录结构、配置文件和操作环境存在显着差异。

本指南提供在CentOS Stream 10环境中使用389 DS成功配置LDAP的必要步骤。

选项 1：在CentOS流10上使用389目录服务器配置LDAP

步骤 1：初始设置

在场景1的选项1中重复步骤1。

CentOS系统不使用APT包管理套件。要在CentOS Stream 10上执行必要的软件安装，请使用dnf(Dandified YUM)或yum包管理器

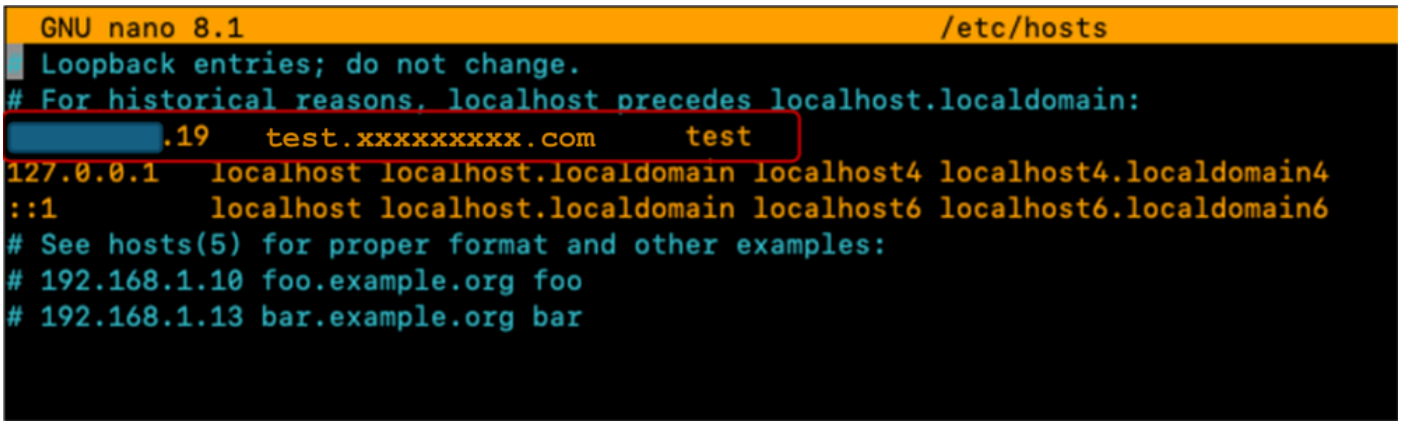
```
sudo yum update
sudo yum install net-tools
```

使用“ifconfig”命令检验服务器IP地址。

将服务器IP地址与服务器完全限定域名(例如：test.xxxxxxxxx.com，在本实验中使用)和主机名(例如

：test)一起添加到“/etc/hosts”文件中，其格式如下：

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



用主机名（测试）替换文件“/etc/hostname”的内容，以更新文件。

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

需要重新启动服务器才能使这些更改生效。

```
sudo reboot
```

步骤 2：安装EPEL回购和389服务器软件包

安装和更新EPEL存储库。

安装389目录服务器软件包。

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

创建包含所需LDAP服务器设置参数的目录模板文件：

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

验证创建的模板文件(ldapconfig.conf)的内容

```
sudo cat ldapconfig.conf
```

编辑ldapconfig.conf模板文件。

```
sudo nano ldapconfig.conf
```

将指定的配置条目插入文件并保存更改。

注意：根据每个环境的特定需求或要求，可能需要进行不同的修改。

本示例包括本演示的基线配置。

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

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

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

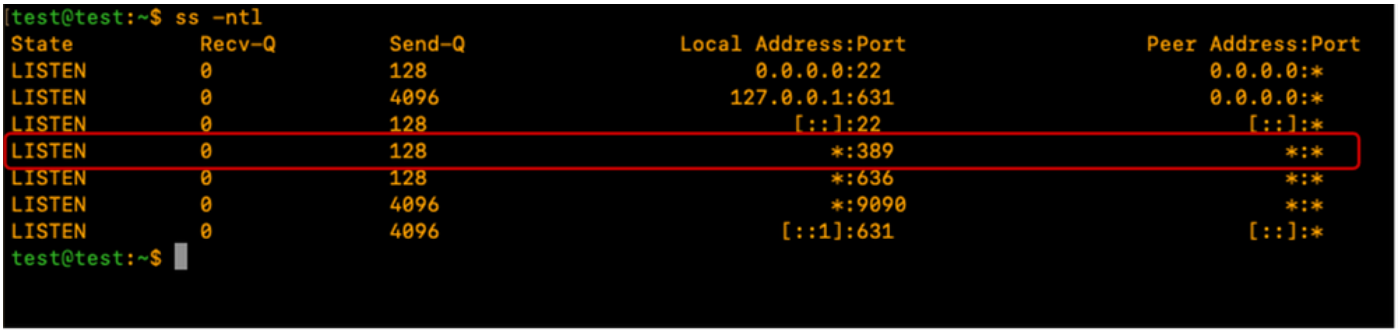
模板文件定义“localhost”目录实例的配置参数。这包括设置管理用户(“admin”)、相关密码和域上下文(“xxxxxxxxx.com”)。

使用之前编辑的模板创建“localhost”目录实例。指定的命令创建和启动LDAP目录服务器：

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

验证LDAP服务是否正在服务器上运行

```
ss -ntl
```



调整CentOS防火墙，以允许LDAP所需的端口（389和/或636）。

在本演示中，防火墙已关闭。

```
sudo systemctl stop firewalld
```

通过运行指定的命令验证LDAP在LDAP服务器上本地运行，并确保它返回LDAP输出，如下所示：

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

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

# xxxxxxxxx,com
dn: dc=xxxxxxxxx,dc=com

# groups, xxxxxxxxx,com
dn: ou=groups,dc=xxxxxxxxx,dc=com

# people, xxxxxxxxx,com
dn: ou=people,dc=xxxxxxxxx,dc=com

# permissions, xxxxxxxxx,com
dn: ou=permissions,dc=xxxxxxxxx,dc=com

# services, xxxxxxxxx,com
dn: ou=services,dc=xxxxxxxxx,dc=com

# demo_user, people, xxxxxxxxx,com
dn: uid=demo_user,ou=people,dc=xxxxxxxxx,dc=com

# demo_group, Groups, xxxxxxxxx,com
dn: cn=demo_group,ou=Groups,dc=xxxxxxxxx,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 8
# numEntries: 7
```

输出包含由389DS服务器创建的演示帐户。LDAP服务器会自动创建默认OU。

用户OU和组OU。可根据要求创建其他OU。

在本演示中，使用默认/自动创建的OU。

有关广泛使用389DS软件包的详细信息，请参阅389DS正式文档:

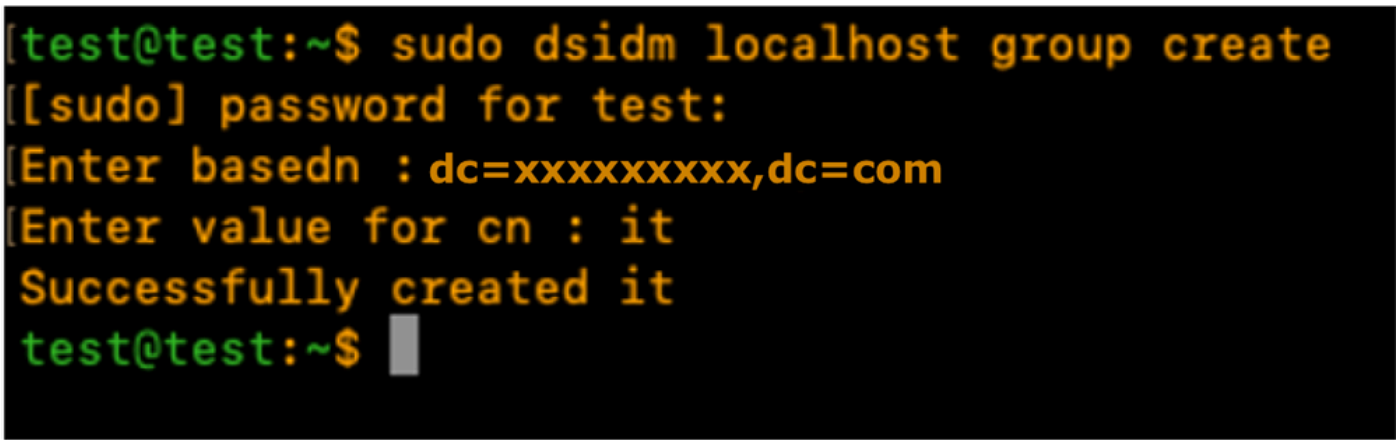
步骤 3：创建LDAP组和用户

使用指定的命令创建组：sudo dsidm <instance_name> group create。

在本演示中，实例名称为“localhost”。

```
sudo dsidm localhost group create
```

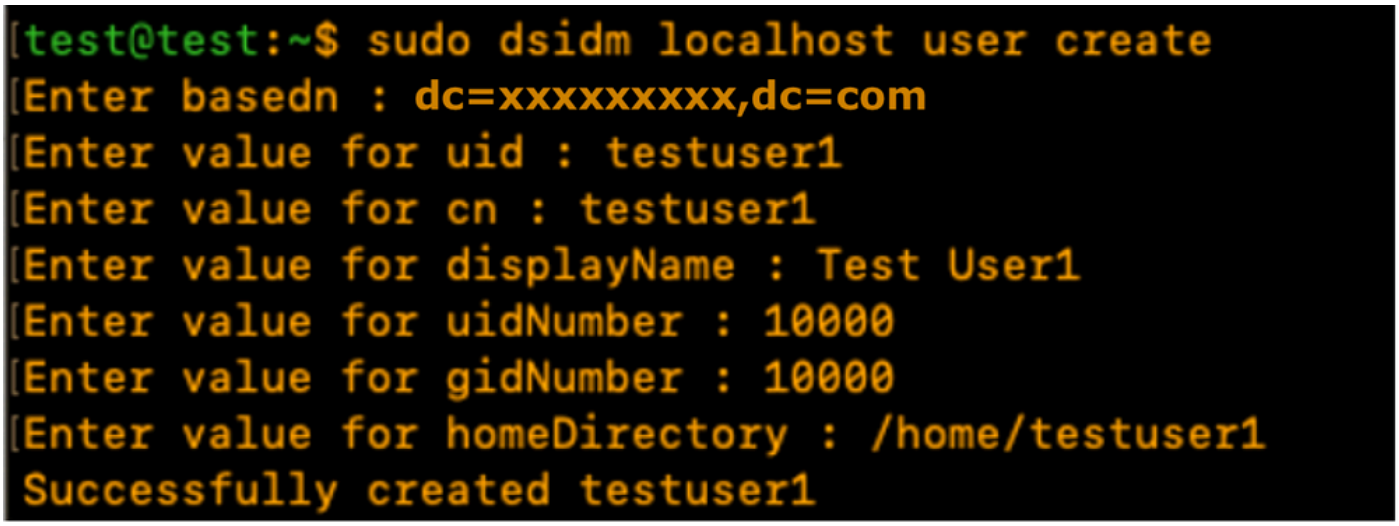
输入终端提示以填充组详细信息，如下所示：



使用命令创建testuser1用户帐户：

```
sudo dsidm localhost user create
```

输入终端提示以填充用户详细信息，如下所示



使用指定的命令为testuser1创建口令，然后输入CLI提示符：

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

使用指定的命令将用户添加到组："sudo dsidm <directory_instance> group add_member <group_cn> <user_dn>"

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

重复用户创建步骤以创建testuser2和bind_user。

注意：确保将每个用户显式添加到其目标组。

忽略此步骤可能会导致访问受限或授权失败。

bind_user帐户不需要是特定组的成员，因为它可以配置为独立帐户，从而灵活地管理目录环境中的管理和服务级别访问。

重新启动Directory实例：

```
sudo dsctl localhost restart
```

步骤 4：安装memberOf重叠

安装“memberOf”插件并重新启动Directory实例：

```
sudo dsconf localhost plugin memberof status
```

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

使用指定的命令配置“memberOf”插件："sudo dsconf <directory_instance> plugin memberof set —scope <base_dn>"

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

使用指定的命令将用户标记为有效的“memberOf”目标："sudo dsidm <directory_instance> user modify <uid> add:objectclass:nsmemberof"

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

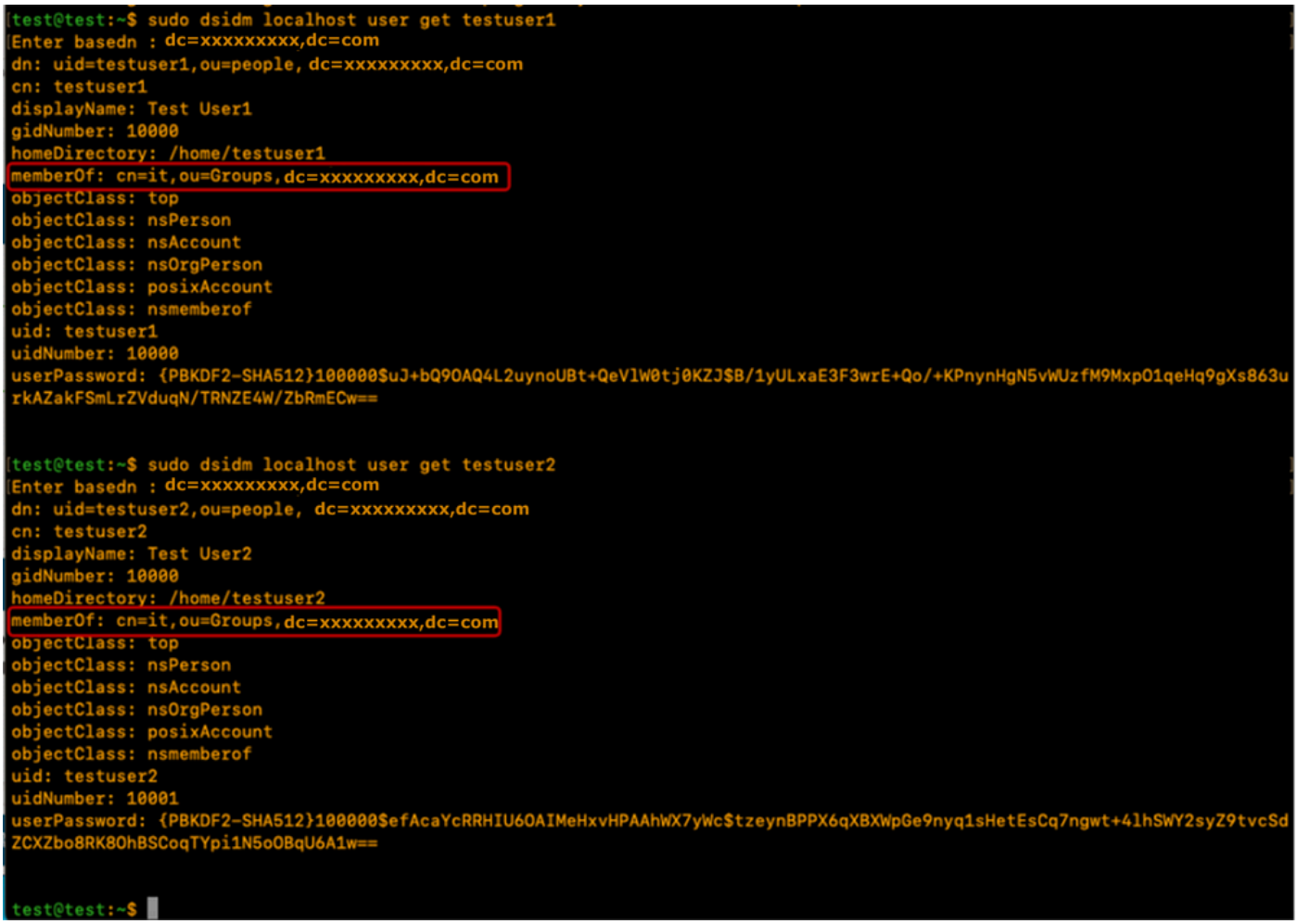
为基本DN生成“memberOf”修正："sudo dsconf <directory_instance> plugin memberof fixup <base_dn>"

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

```
test@test:~$ sudo dsconf localhost plugin memberof fixup dc=xxxxxxxxx,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". Th
is task is running in the background. To track its progress you can use the "fixup-status" command.
test@test:~$
```

验证用户配置：

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



389DS LDAP服务器配置有memberOf插件以支持memberOf属性。

CIMC上的配置参数

登录CIMC。

在“导航”(Navigation)窗格中，依次选择管理(Admin)、用户管理(User Management)和LDAP。

如下所示填充LDAP配置参数：

- 启用LDAP：选中
- 基准 DN:dc=xxxxxxxxx，dc=com

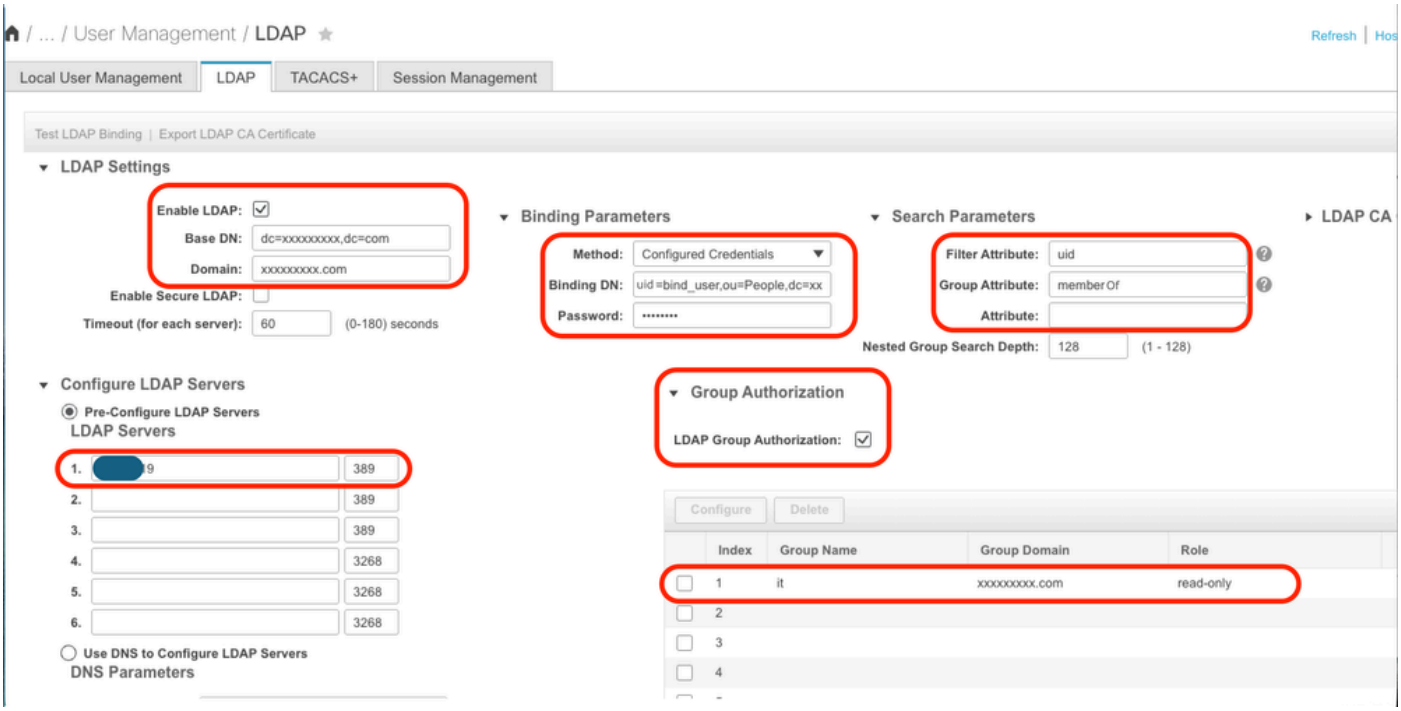
- 域名：xxxxxxxxx.com

- LDAP服务器：<ldap_server_IP或FQDN> X.X.X.19

- 绑定参数：可以是“登录凭证”或“配置的凭证”
  - 使用配置的凭证时，请完全按照在LDAP服务器上配置的步骤添加bind_user DN:
    - 例如："cn=bind_user，ou=People，dc=xxxxxxxxx，dc=com"或"uid=bind_user，ou=People，dc=xxxxxxxxx，dc=com"

- 搜索参数：
  - 过滤器属性："cn"或"uid"
  - 组属性：成员

- LDAP组授权 — 已选中
  - 组名称:it
  - 组域：xxxxxxxxx.com
  - 角色：只读（任何首选角色）



保存配置并测试LDAP用户登录。

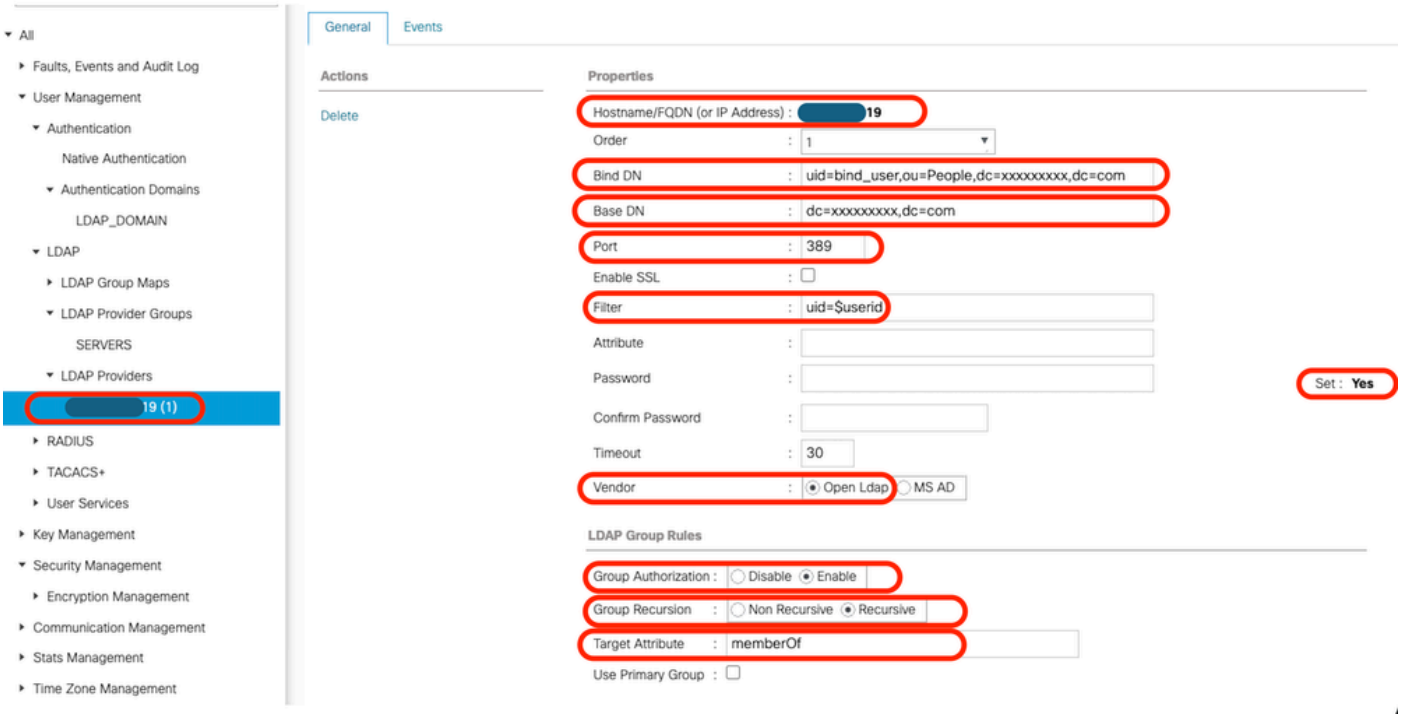
UCS Manager上的配置参数

登录UCS Manager。

在“导航”(Navigation)窗格中，依次选择管理(Admin)、用户管理(User Management)和LDAP。

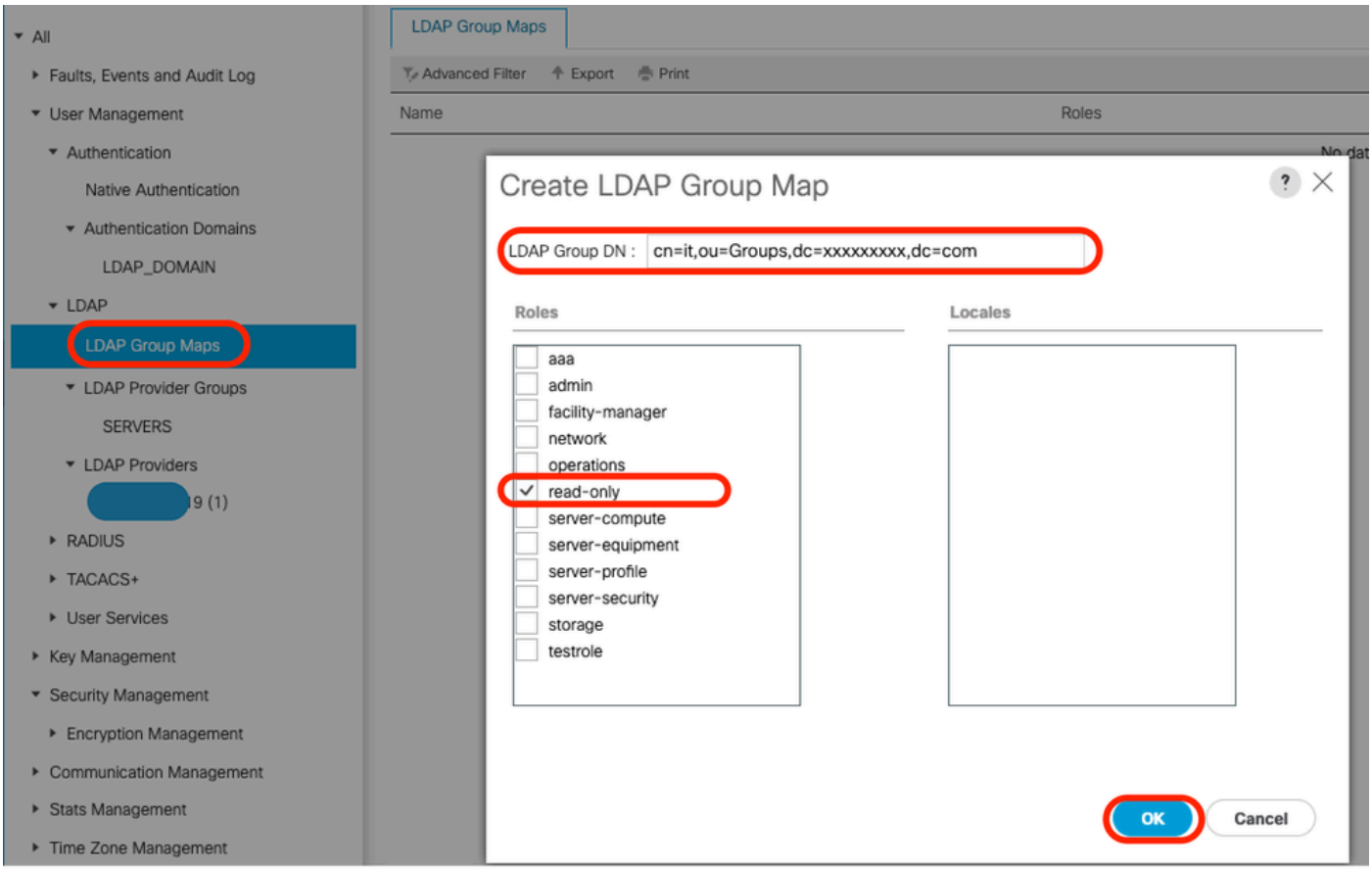
如下所示填充LDAP配置参数：

- LDAP提供程序：
  - 主机名：<LDAP服务器的FQDN或IP地址>
  - 绑定DN:uid=bind_user，ou=people，dc=xxxxxxxxx，dc=com
  - 基准 DN:dc=xxxxxxxxx，dc=com
  - 端口：389
  - 启用 SSL:禁用
  - 过滤器：uid=$userid
  - 组授权：启用
  - 组递归：递归
  - 目标属性：成员
- LDAP组映射：
  - LDAP组DN:cn=it，ou=Groups，dc=xxxxxxxxx，dc=com



将已配置的LDAP提供程序添加到LDAP提供程序组。在本演示中，使用“SERVERS”LDAP提供程序组。

配置LDAP组映射，添加从LDAP服务器检索的“LDAP组DN”。



在引用LDAP提供程序组的All >> User Management >> Authentication >> Authentication Domains中配置LDAP身份验证域(LDAP_DOMAIN)并测试LDAP用户登录。

结论

虽然本指南涵盖基本部署场景，但进一步探索LDAP功能可以显着增强目录性能和安全。

有关其他信息、最佳实践和高级配置详细信息，请参阅指定的资源：

- OpenLDAP官方文档
- LDAP客户经理 — 手动
- 389目录服务器文档
- 在UCS Manager上配置LDAP
- 在UCS C系列服务器上配置安全LDAP

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。