UCS Manager &에서 LDAP 구성 Linux OpenLDAP 및 389-DS 서버를 사용하는 CIMC

목차

소개

이 문서에서는 Linux 기반 OpenLDAP 및 389 디렉토리 서버를 사용하여 UCS Manager 및 CIMC의 인증 방법으로 LDAP를 구성하는 다양한 옵션에 대해 설명합니다.

배경 정보

OpenLDAP 서버 컨피그레이션의 광범위한 가변성으로 인해 완전한 처리는 이 문서의 범위를 벗어납니다. 이 문서에서는 여러 Linux 배포, LDAP 서버 패키지 및 특성 스키마를 포괄하는 일반적으로 구현된 구성을 강조합니다. 이 문서에서는 명확성과 간소화를 위해 표준 LDAP 구성에 대해 설명합니다. LDAPS(Secure LDAP) 컨피그레이션은 이 문서에서 다루지 않습니다.

사전 요구 사항:

다음 항목에 대한 지식이 권장됩니다.

- UCS B 시리즈
- UCS C 시리즈
- Linux 서버 관리

사용되는 구성 요소

이 문서의 정보는 다음 소프트웨어 및 하드웨어 버전을 기반으로 합니다.

- UCS Manager 펌웨어 버전: 4.3(2c)
- 패브릭 인터커넥트 모델: UCS-FI-6454
- UCS C Series 독립형 서버 모델: UCSC-C240-M5
- UCS C Series 독립형 펌웨어 버전: 4.3(2.250045)
- Ubuntu 20.04
- CentOS 스트림 10

이 데모에 사용되는 설정:

- LDAP 서버 호스트 이름: 테스트
- 서버 도메인: xxxxxxxxx.com
- 서버 FQDN: test.xxxxxxxxx.com
- Linux 서버(Ubuntu 및 CentOS) IP 주소: X.X.X.19
- OpenLDAP 사용자 testuser1, testuser2

- LDAP 그룹 열기: IT
- OpenLDAP 바인드 사용자 계정: bind_user

참고: 이 실습에서는 linux Nano 텍스트 편집기를 사용했습니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

시나리오 1: 우분투 - 데비안

LDAP 서버 컨피그레이션은 관리 환경 설정 및 필수 제어 레벨에 따라 LDAP 계정 관리자 같은 그래픽 인터페이스 또는 명령줄 툴을 사용하여 수행할 수 있습니다. 이 시나리오에서는 Linux 기반 OpenLDAP를 사용하는 컨피그레이션을 검토하고, GUI 기반 구축부터 시작하여 명령줄 유틸리티로 전환하여 오버레이 플러그인(일반적으로 Cisco UCS Manager와의 통합에서 사용됨)을 비롯한 고급 기능을 살펴봅니다.

옵션 1: Ubuntu LDAP 계정 관리자(LAM)를 사용하여 OpenLDAP 구성

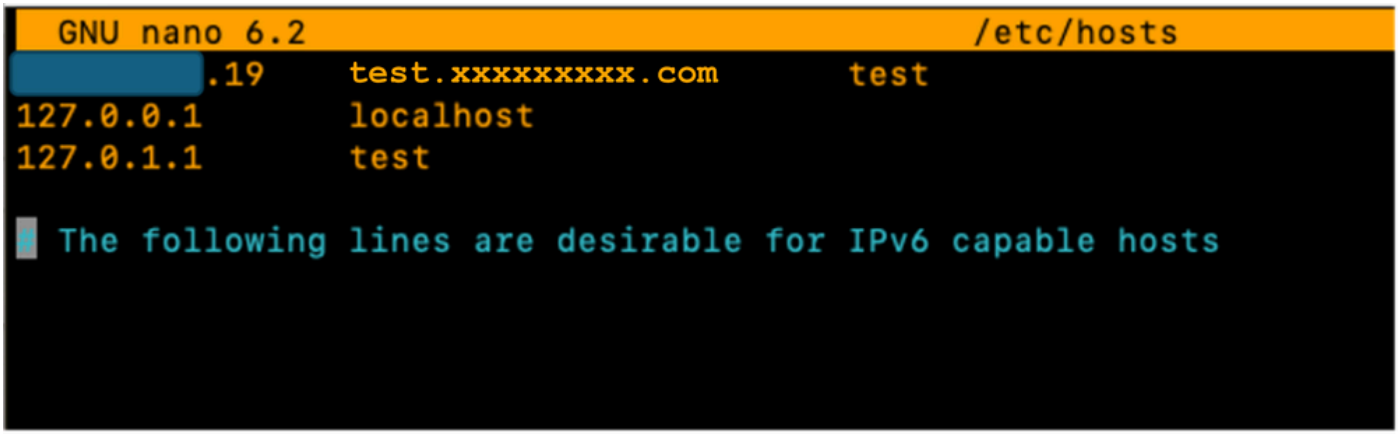
1단계: Linux 서버 호스트 이름 및 net-tools의 초기 구성

Ubuntu를 업데이트하고 ifconfig, netstat 등의 툴에 액세스할 수 있도록 net-tools 패키지를 설치합니다.

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

"ifconfig" 명령을 사용하여 서버 IP 주소를 확인한 다음 서버 도메인 이름과 함께 "/etc/hosts" 파일에 추가합니다(예: "test.xxxxxxxxx.com"(이 실습에 사용됨) 및 호스트 이름(예: "test")를 지정합니다.

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



또한 "/etc/hostname" 파일의 내용을 hostname(테스트)으로 바꾸어 업데이트합니다.

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

이러한 변경 사항을 적용하려면 서버를 재부팅해야 합니다.

```
sudo reboot
```

2단계: SLAPD, Apache, PHP 및 그 종속성 설치

다음으로, Apache, PHP 및 그 종속성을 설치합니다. 웹 페이지를 통해 GUI 상호 작용을 활성화하는 데 사용됩니다.

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

열려 있는 LDAP 서버 패키지 "slapd" 및 해당 종속성(ldap-utils) 설치

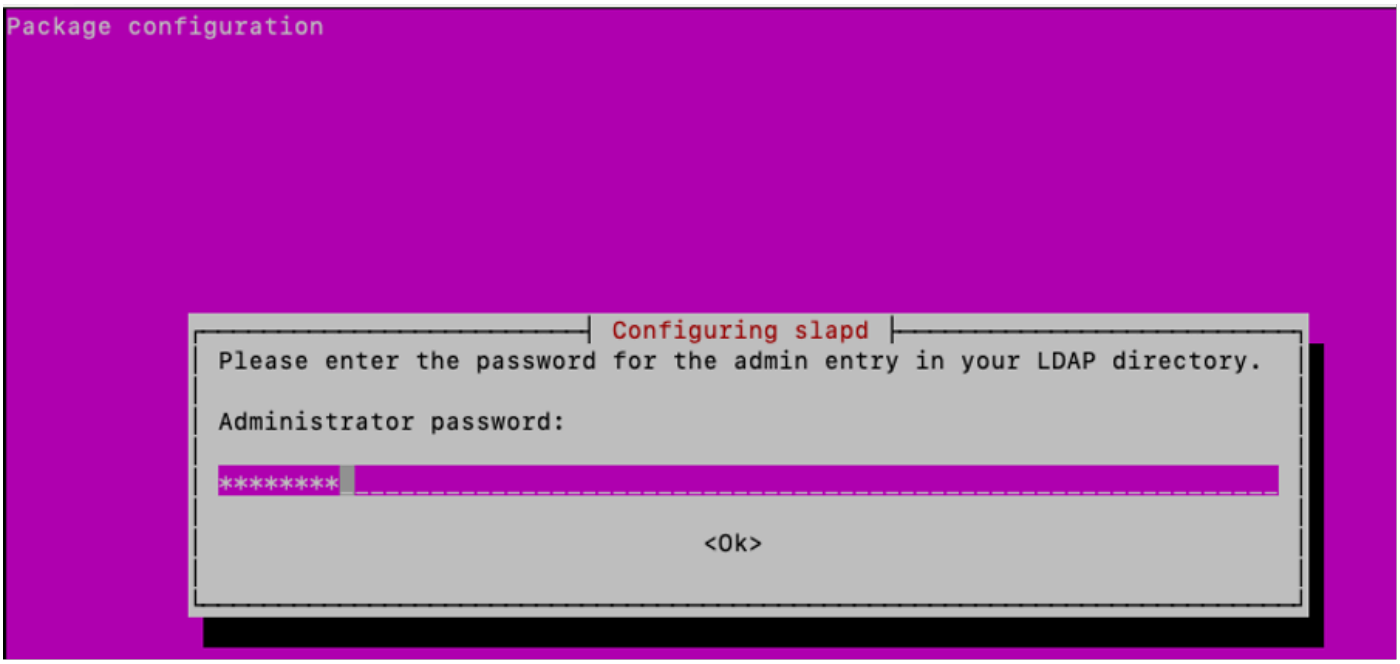
```
sudo apt install slapd ldap-utils -y
```

slapd 설치 중에 표시되는 GUI 팝업에서 추가 필수 SLAPD 패키지 컨피그레이션을 입력합니다.

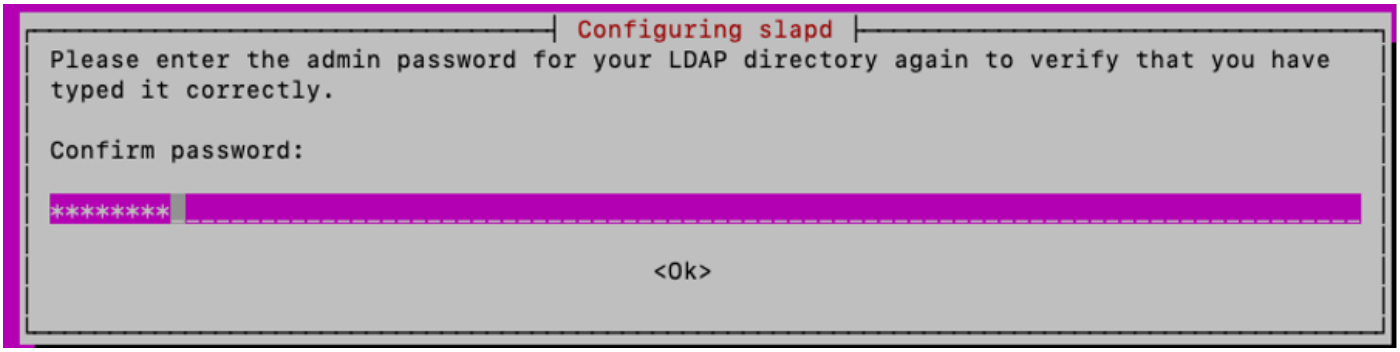
참고: 비밀번호를 분실하려면 LDAP 서버를 다시 설치해야 합니다.

이 컨텍스트에서 "관리자"(admin)는 OpenLDAP 서비스, 모듈 및 컨피그레이션을 관리하는 데 사용되는 계정입니다.

LDAP 패키지 "administrator" 비밀번호를 추가하고 키보드의 enter 키를 눌러 "OK"를 선택합니다.



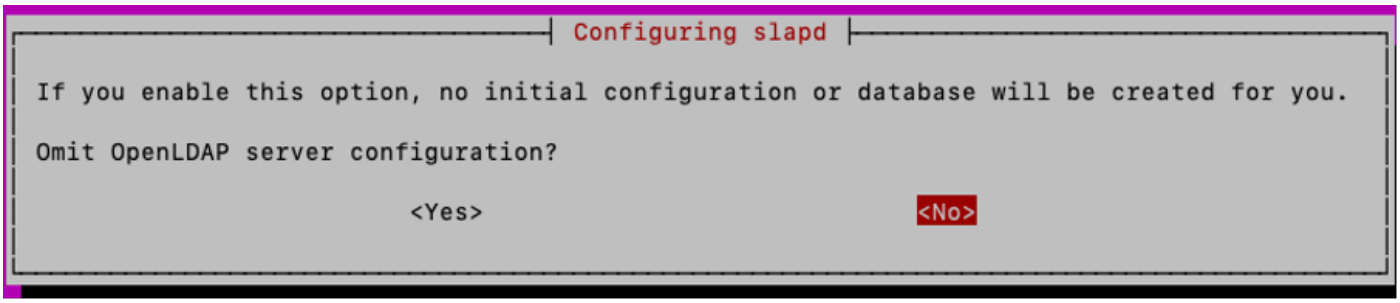
비밀번호를 확인합니다.



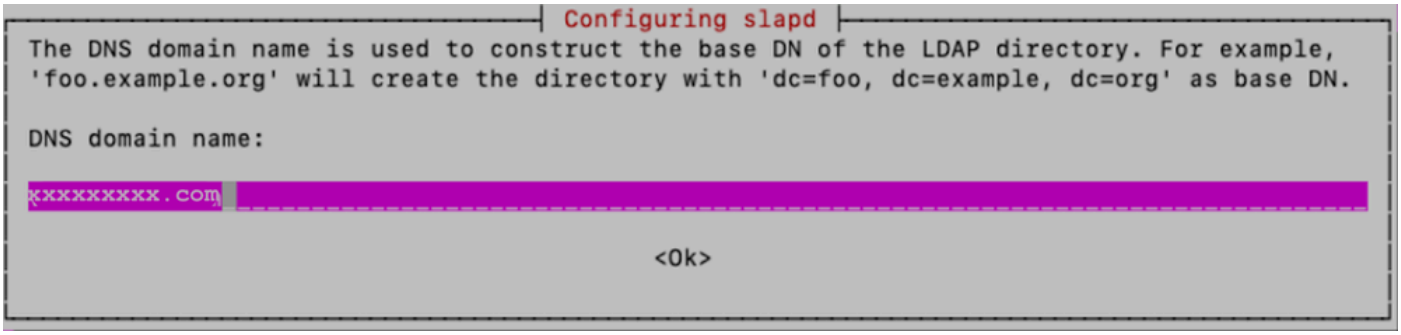
설치가 완료되면 지정된 명령을 사용하여 SLAPD 패키지를 재구성하고 도메인 정보를 추가할 수 있습니다.

```
sudo dpkg-reconfigure slapd
```

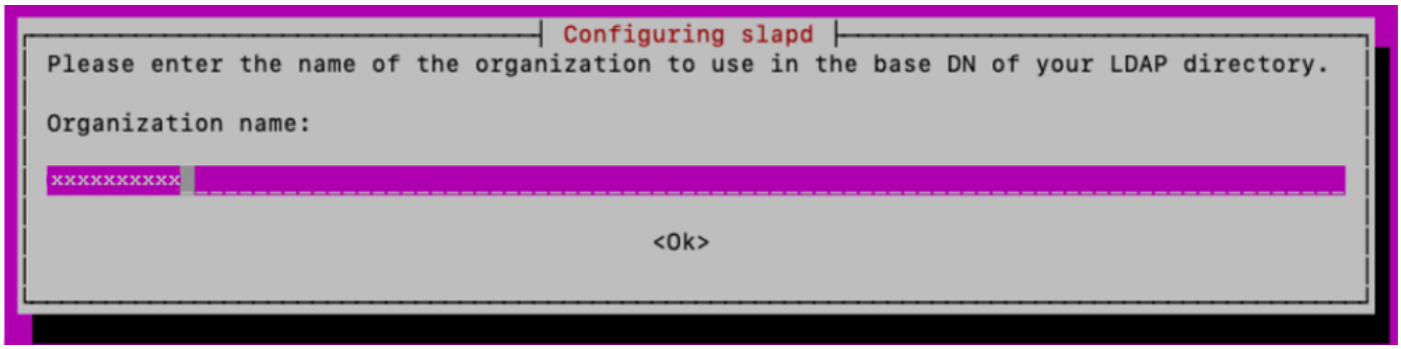
"OpenLDAP 서버 컨피그레이션 생략"에 대해 기본 "No" 옵션을 수락하고 enter 키를 누를 수 있습니다.



도메인 이름을 입력하고 Enter를 누릅니다.



이 실습에서는 "xxxxxxxxx"가 "Organization name"으로 사용됩니다.

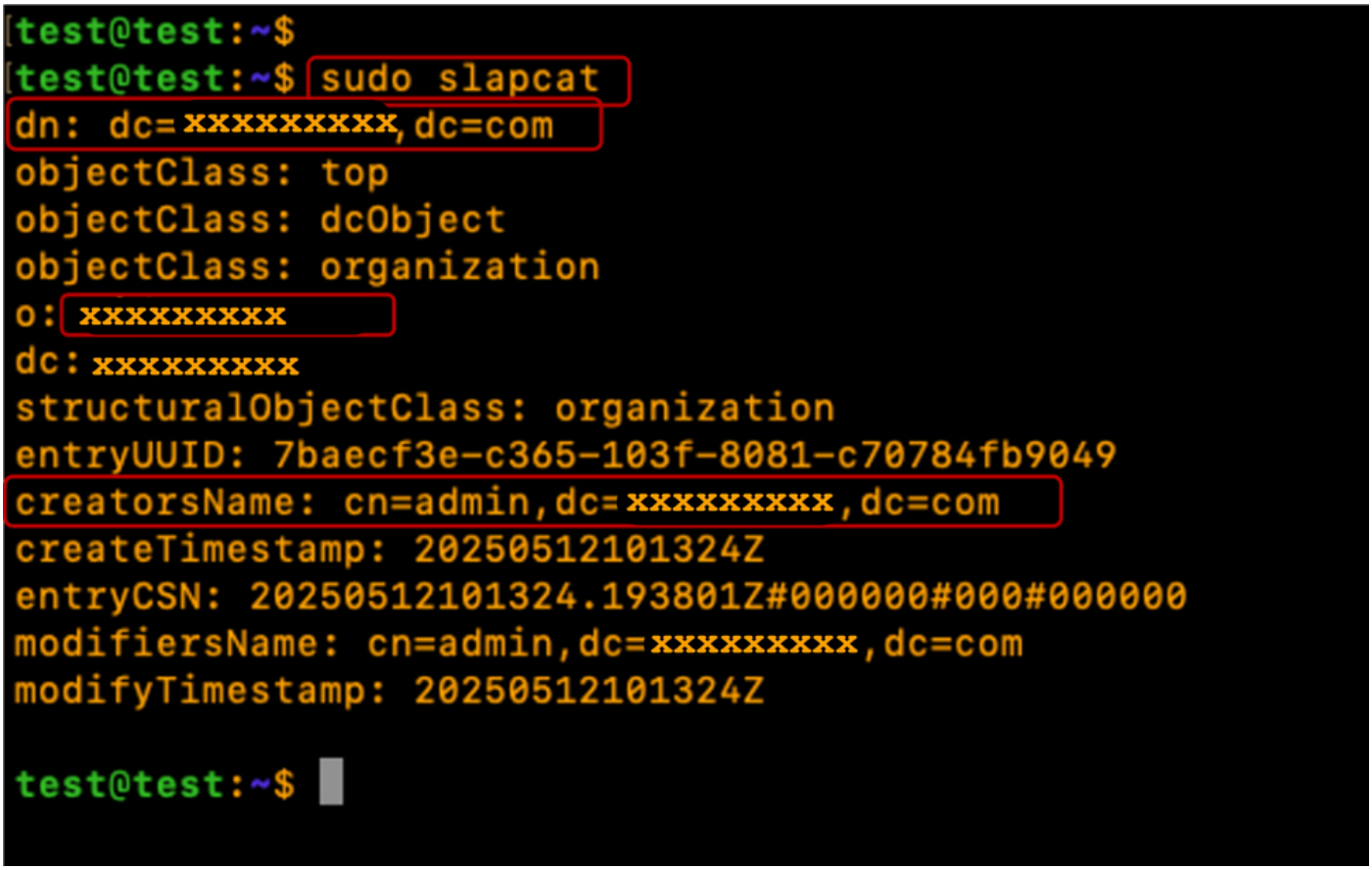


그런 다음 "Administrator password"를 입력하고 확인합니다

다른 컨피그레이션 옵션의 경우 기본값을 유지하고 키보드의 Enter 키를 눌러 컨피그레이션을 완료합니다.

다음 명령을 사용하여 SLAPD 설치를 확인합니다.

```
sudo slapcat
```



3단계: LDAP 계정 관리자 설치

LDAP 사용자 및 그룹의 생성 및 관리를 위해 LAM(LDAP Account Manager)을 설치합니다.

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

LAM에서 요구하는 PHP-CGI PHP 확장을 활성화합니다.

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

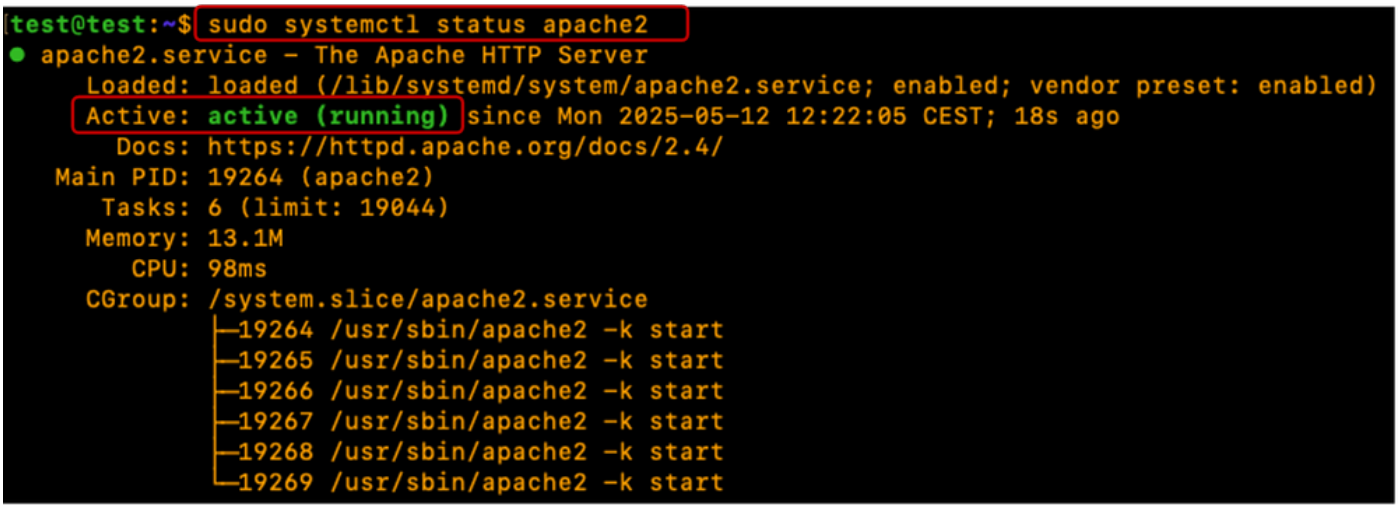
Apache를 다시 로드하여 새 컨피그레이션을 활성화합니다.

부팅 시 Apache 서비스를 다시 시작하고 자동 시작할 수 있도록 설정:

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

Apache Server 상태가 "Running(실행 중)" 및 "Active(활성)"인지 확인합니다.

```
sudo systemctl status apache2
```



포트 80(웹), 443(보안 웹), 389(LDAP) 및 636(필요한 경우 보안 LDAP)을 허용하도록 Ubuntu 방화벽 구성

```
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 방화벽 상태를 확인합니다.

```
sudo ufw status
```

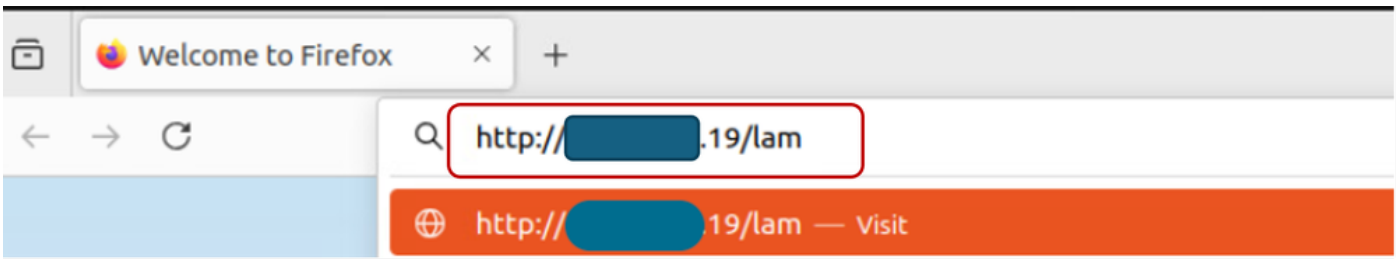
```
test@test:~$ sudo ufw status
Status: active

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)
```

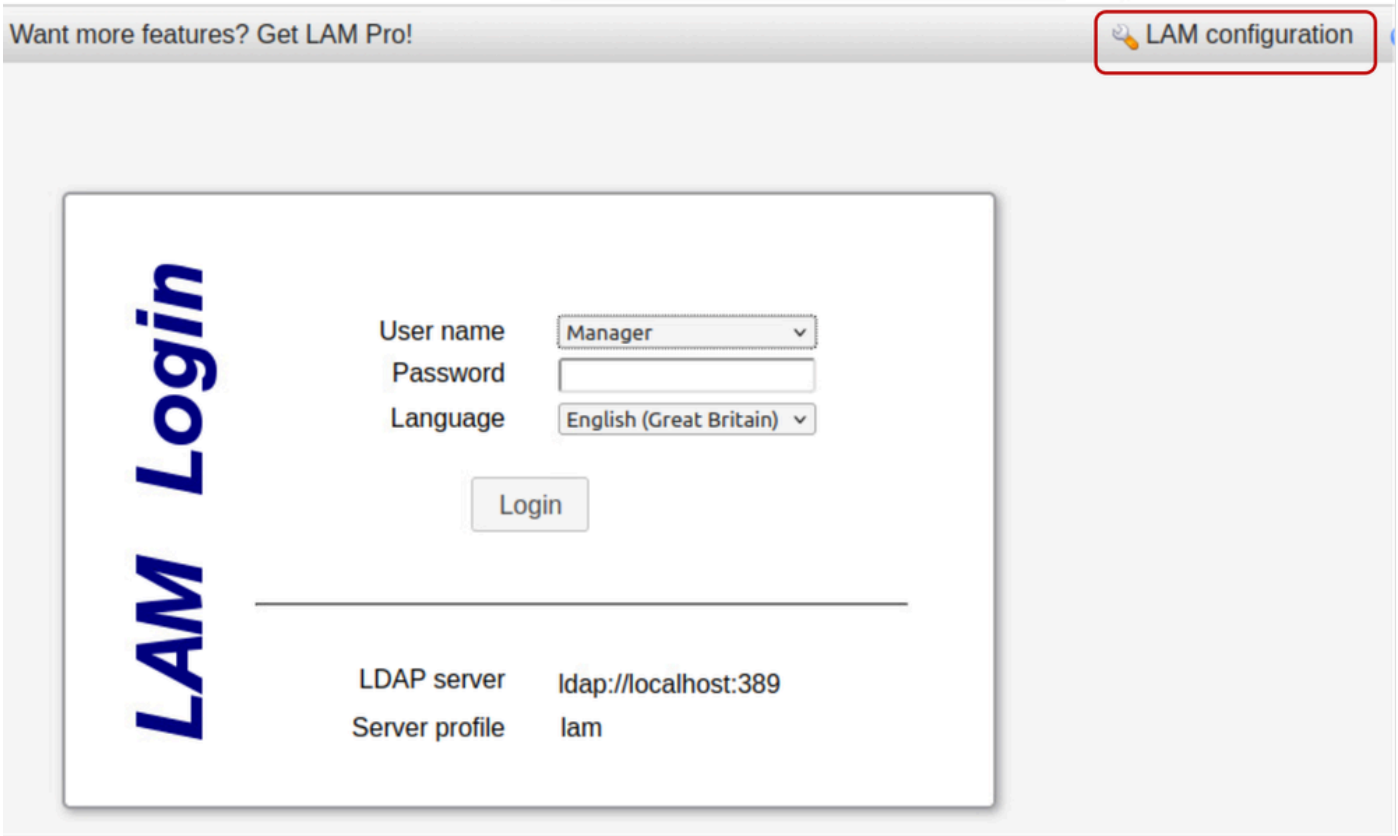
4단계: LDAP 계정 관리자 구성

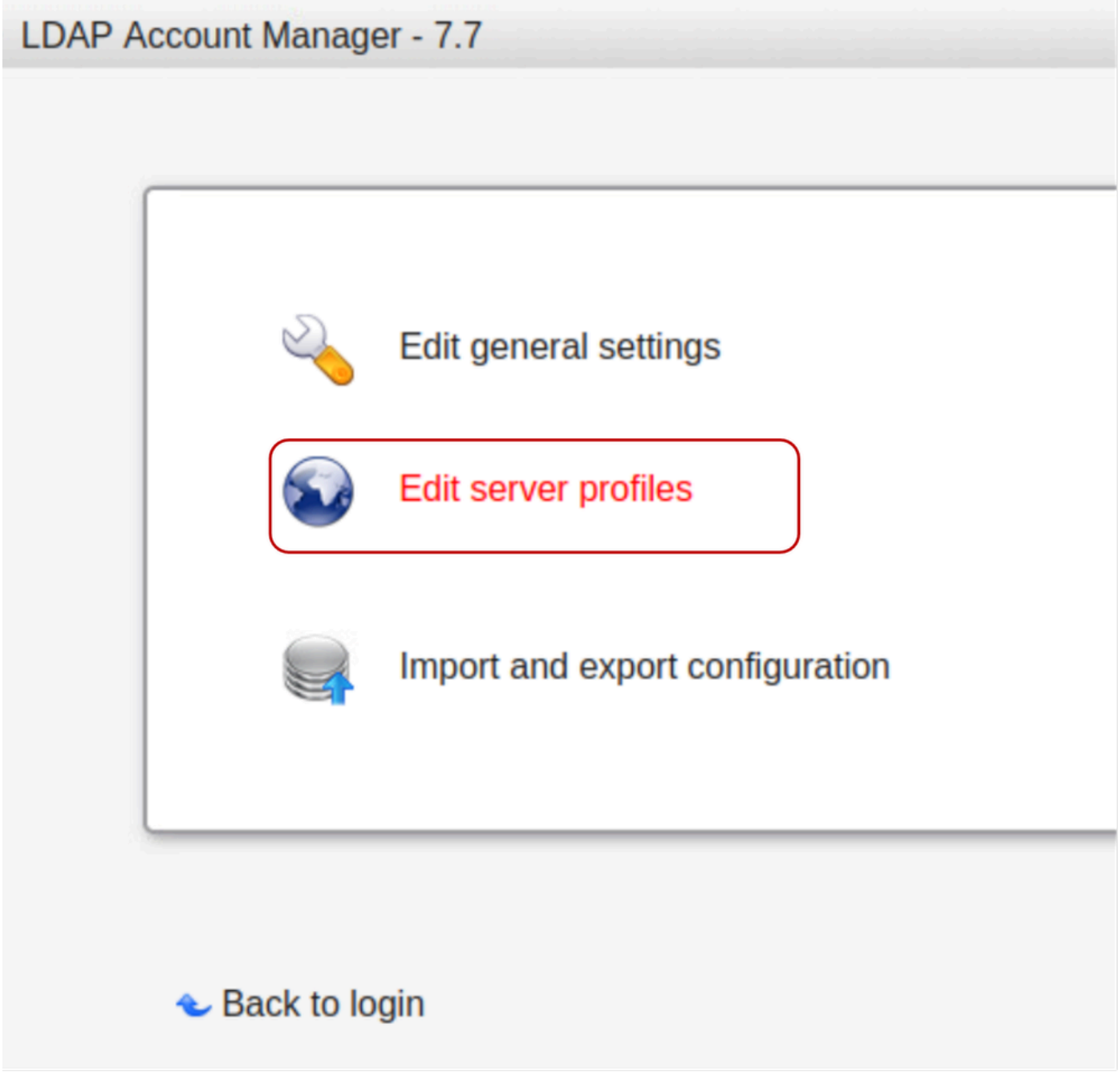
GUI에서 LAM(LDAP Account Manager)을 구성하려면 웹 브라우저를 열고 Linux 서버 IP 주소를 입력한 다음 표시된 대로 'lam' 경로를 추가합니다.

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

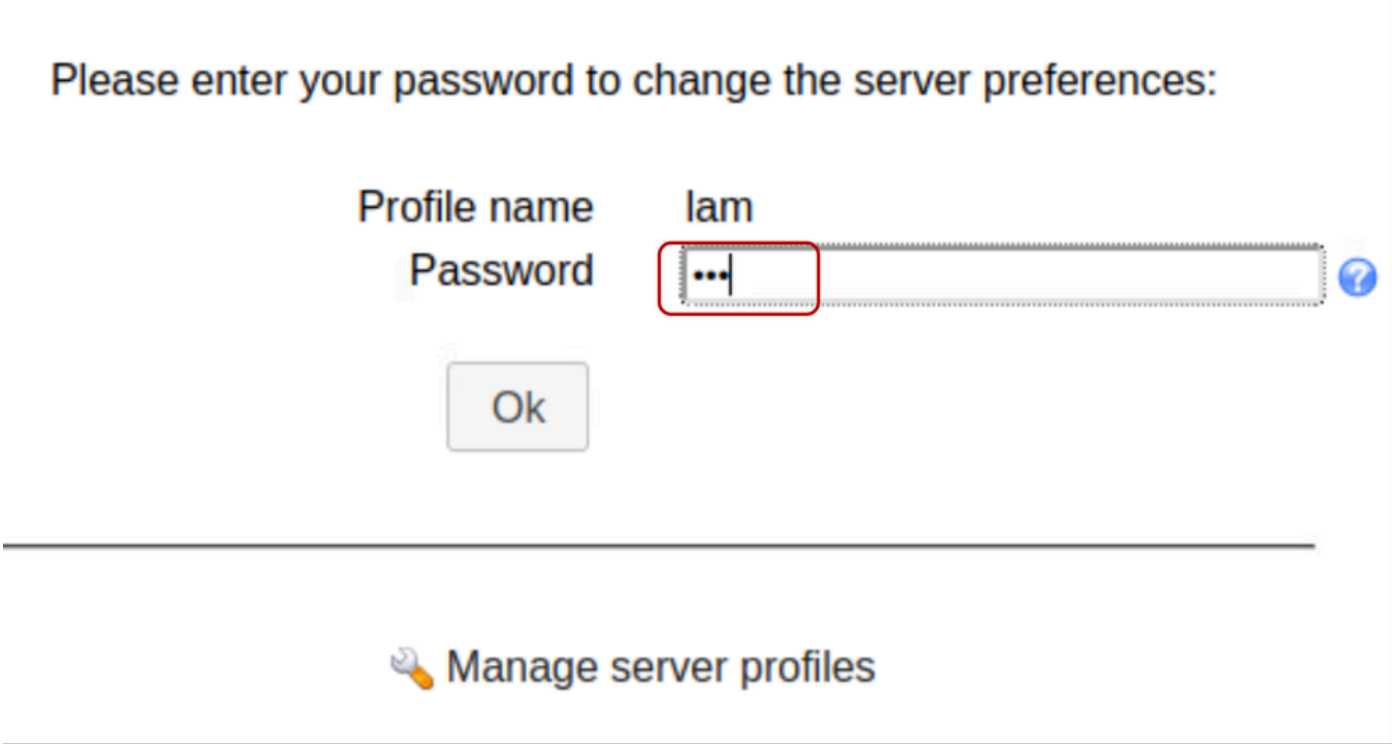


"LAM configuration(LAM 컨피그레이션)"을 클릭한 다음 "Edit server profiles(서버 프로필 수정)"를 선택합니다.





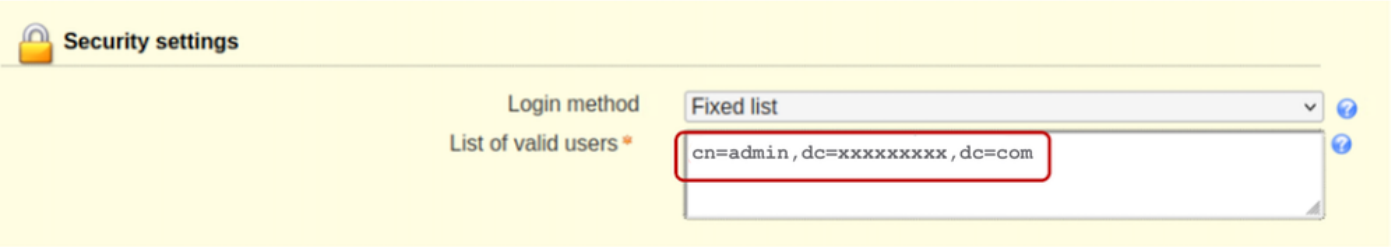
기본 lam 비밀번호 "lam"을 입력하여 로그인합니다.



General Settings(일반 설정) 탭에서 서버 설정, "Language(언어)" 및 "Timezone(시간대)"을 확인합니다.

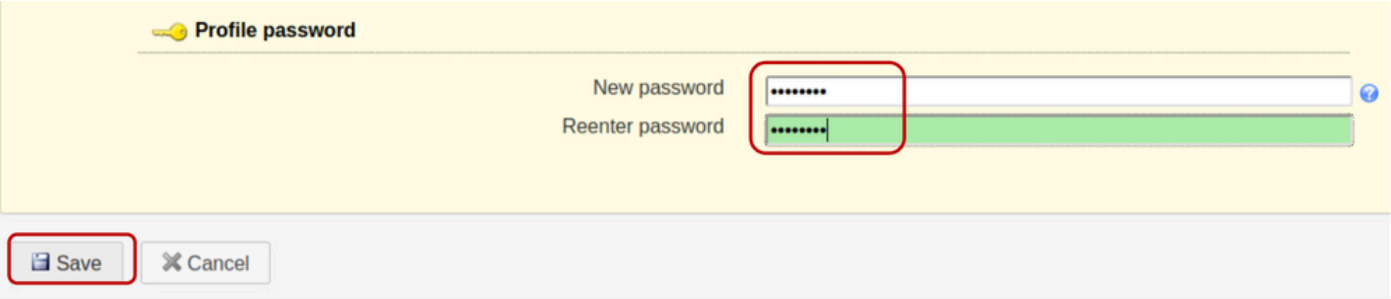
Tool settings(툴 설정) 섹션에서 아래와 같이 Tree suffix(트리 접미사) 필드에서 필요한 도메인 이름을 편집하여 추가합니다.

SLAPD 서비스를 관리하는 데 사용되는 "admin" 사용자를 포함하도록 Security settings(보안 설정) 섹션을 수정합니다.



"Profile Password(프로파일 비밀번호)"를 설정합니다. 이 비밀번호는 LAM 컨피그레이션 인터페이스에 대한 후속 로그인에 사용됩니다. 이 예에서는 기본 "lam" 비밀번호 대신 "cisco123"이 구성됩니다.

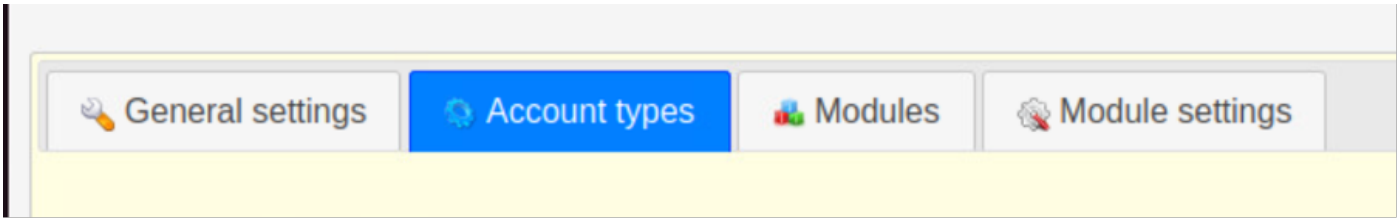
설정 저장:



그런 다음 LAM 컨피그레이션 GUI 인터페이스에서 세션이 다시 시작됩니다.

생성된 새 비밀번호를 사용하여 다시 로그인합니다(LAM configuration(LAM 컨피그레이션) >> Edit server profiles(서버 프로필 수정)).

"Account types(어카운트 유형)"를 클릭합니다.



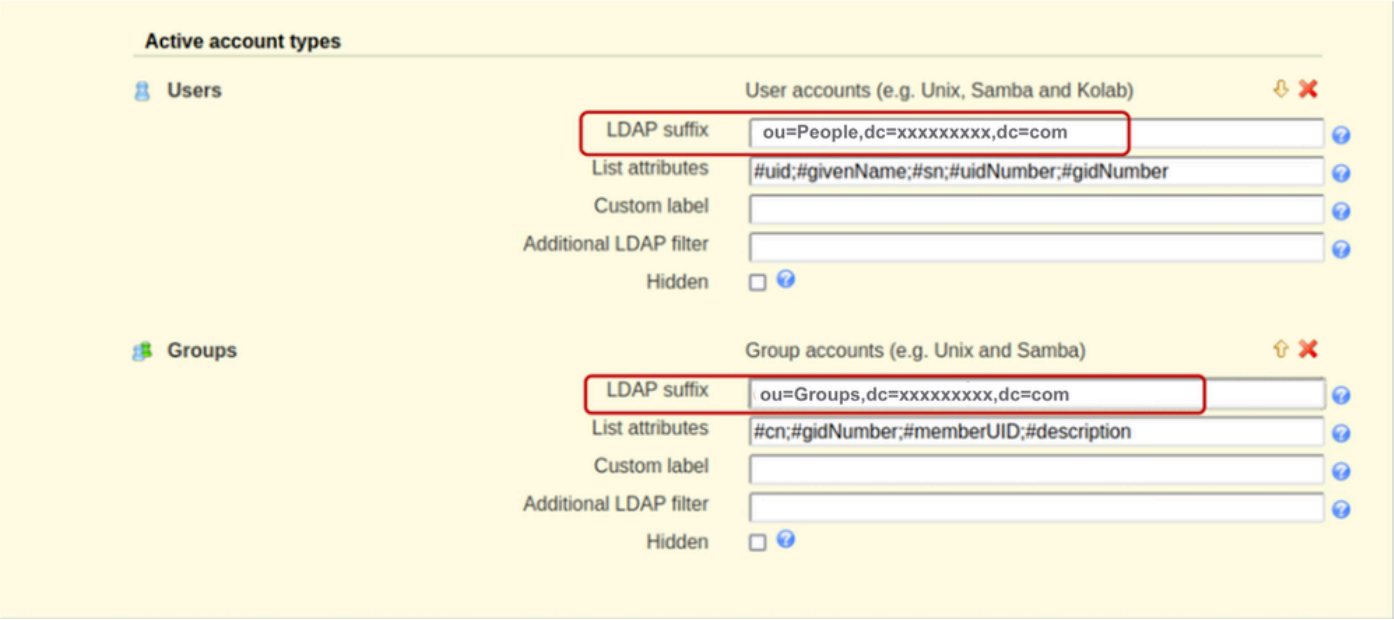
아래로 스크롤하여 LDAP 접미사 필드에 도메인 이름 정보가 있는 기본 Active(활성) 계정 유형을 수정합니다. 예를 들어 "LDAP 접미사" 필드의 기본 내용은 "ou=People,dc=my-domain,dc=com"과 같은 값을 표시합니다.

새 조직 구성 단위를 생성해야 하는 경우 조직 구성 단위의 이름을 포함하도록 "LDAP 접미사" 필드의 내용을 대체합니다.

형식은 "ou=<organizational_unit>,dc=xxxxxxxxx,dc=com"으로 표시됩니다.

이 데모에서 사용자의 OU는 "People"이고 그룹의 OU는 "Groups"입니다.

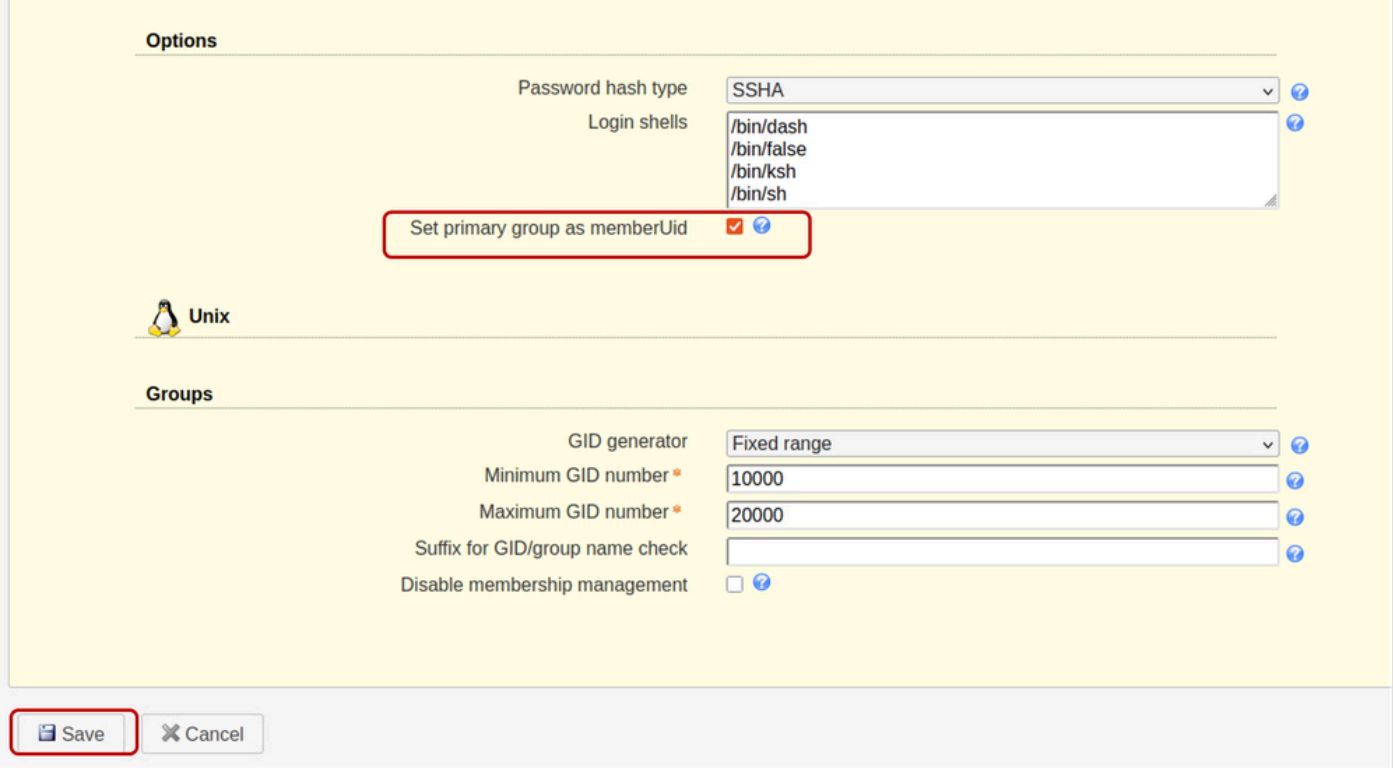
설정 저장.



아래로 스크롤하여 Options(옵션) 섹션으로 이동한 다음 "Set primary group as memberUid(기본 그룹을 memberUid로 설정)"를 확인합니다.

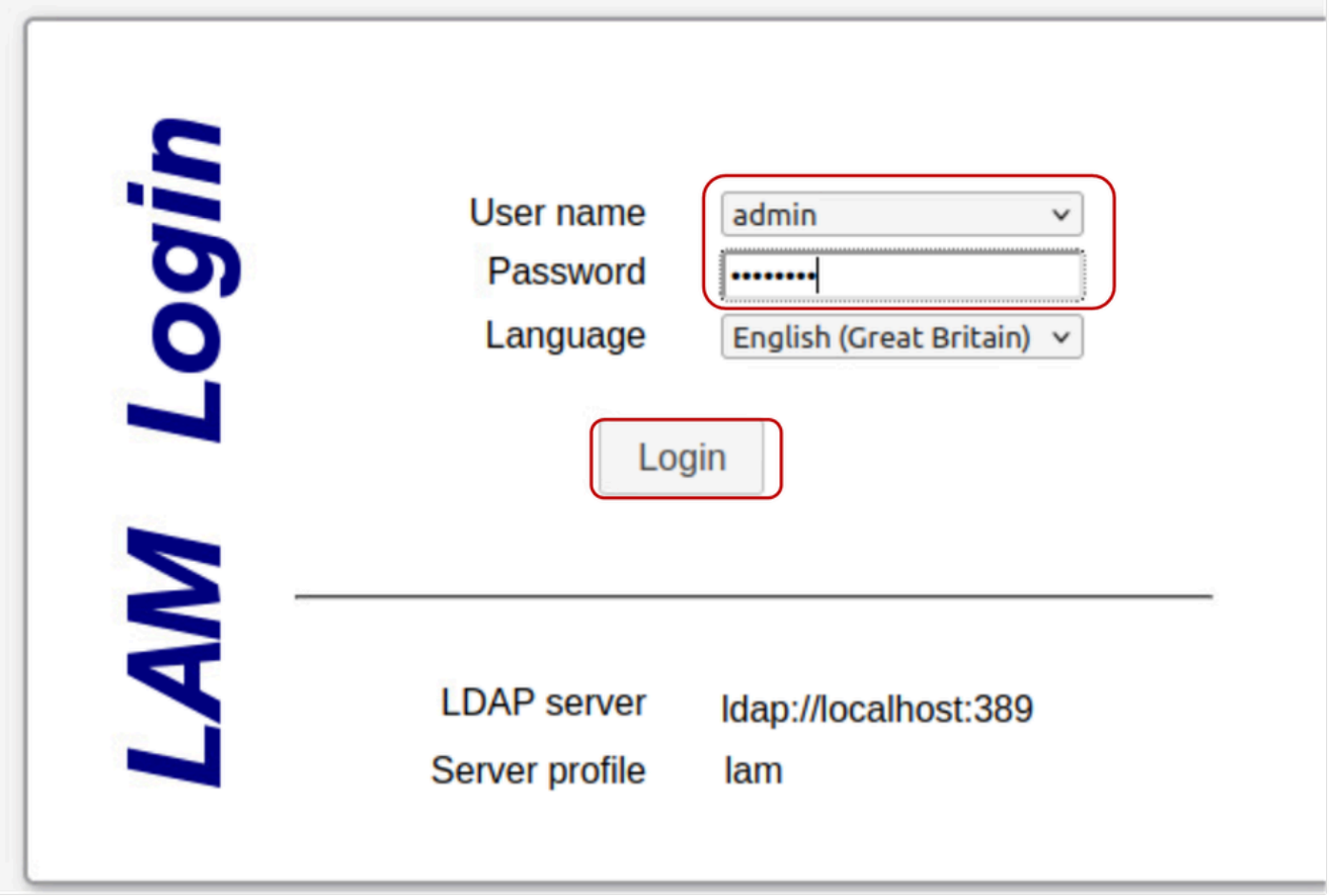
기본적으로 "Set primary group as memberUid(기본 그룹을 memberUid로 설정)" 옵션은 그룹 개체에 설정되어 있지 않습니다. 이를 활성화하면 표준 LDAP 그룹과 같이 OpenLDAP "기본 그룹"을 사용할 수 있습니다. 여기서 "memberUid"를 참조할 수 있습니다(예: UCS C Series 서버 컨피그레이션에서). 이 옵션을 선택하지 않으면 기본 그룹에 속한 사용자의 로그인이 실패합니다.

설정 저장.

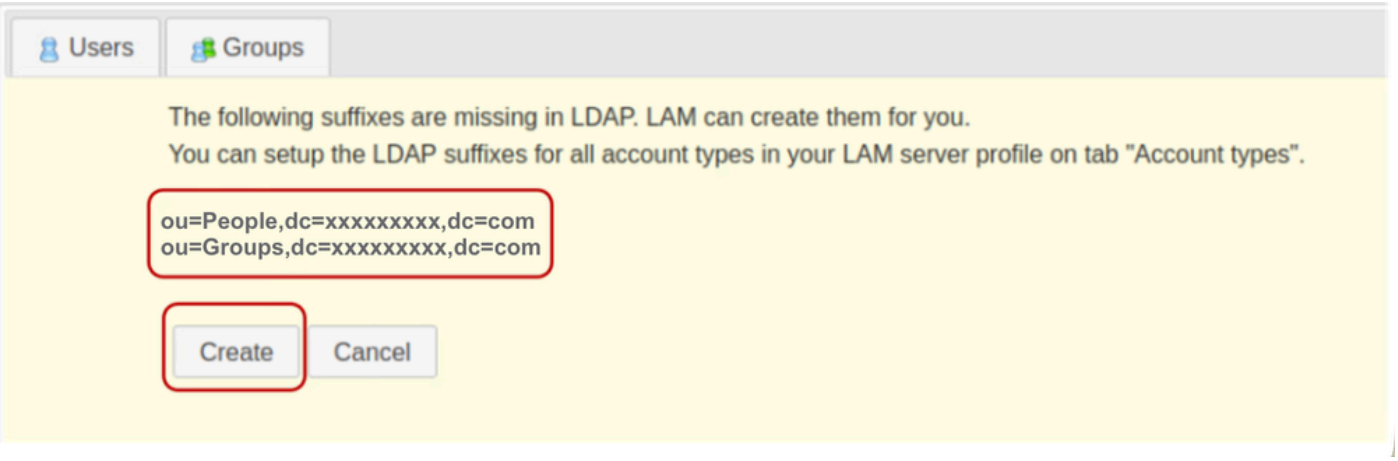


5단계: OU, 그룹 및 사용자 생성

LAM에 설치 중에 생성된 동일한 비밀번호를 사용하여 "admin" 사용자로 로그인하고, 앞서 생성된 OU(People 및 Groups)에 속한 Users 및 Groups를 각각 생성합니다.

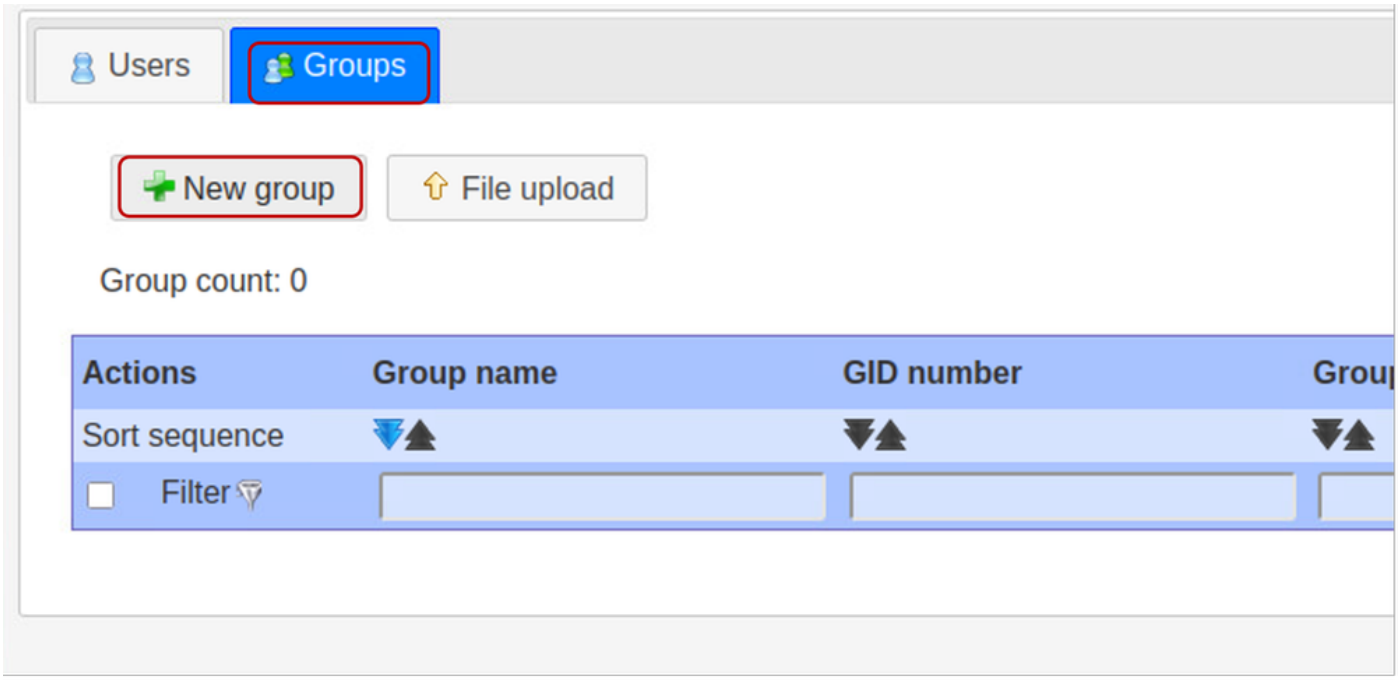


LAM Configuration(LAM 컨피그레이션) 섹션에서 이전에 지정된 OU를 생성합니다.
Create(생성)를 클릭합니다.



다음으로, LDAP Account Manager에서 "it" 그룹을 생성합니다.

Groups(그룹) 탭을 선택하고 New group(새 그룹)을 클릭합니다

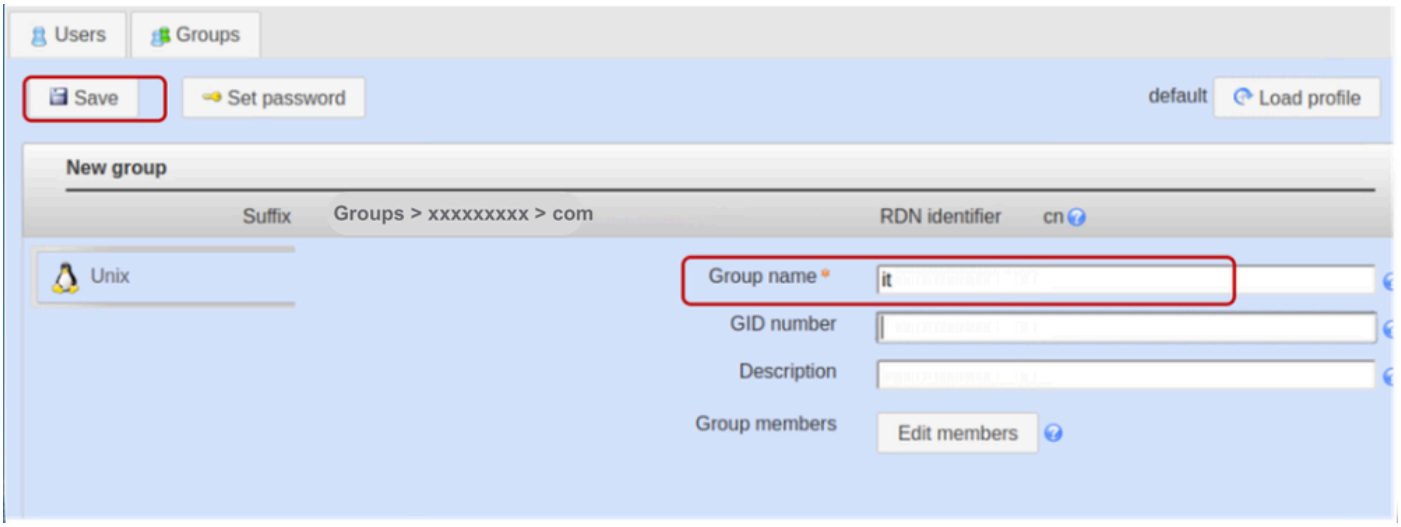


그룹 이름을 "it"로 설정합니다.

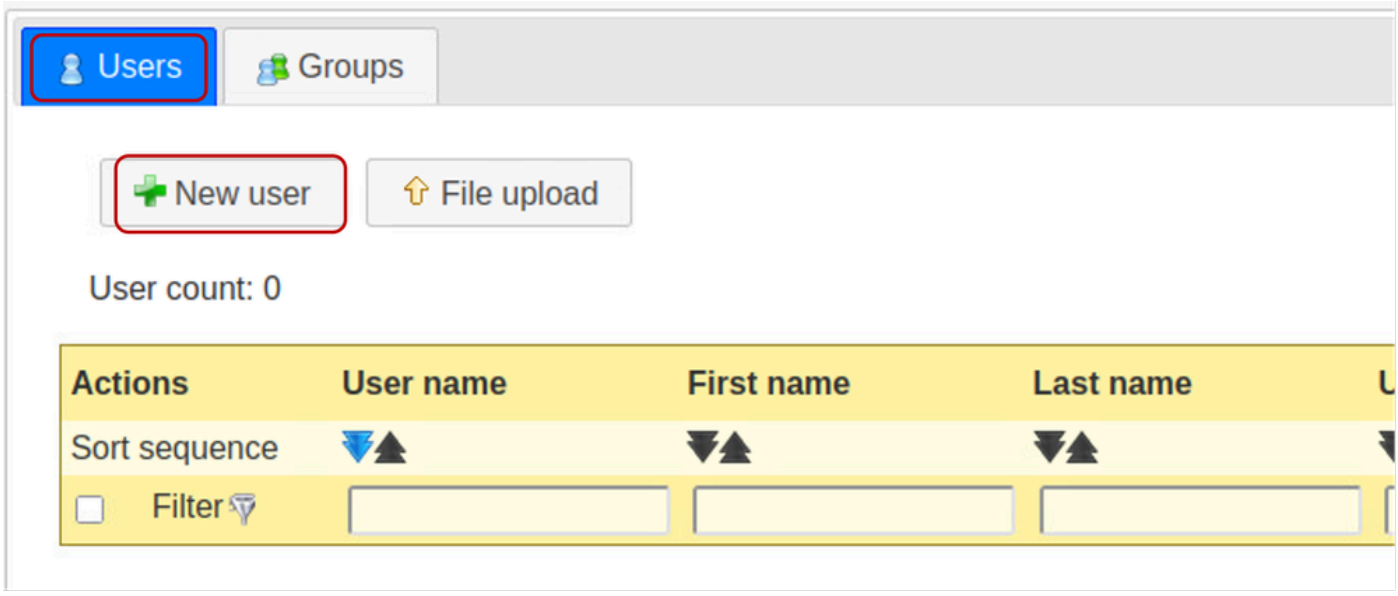
참고: Cisco UCS 시스템은 일반적으로 케이스 변형에 대한 복원력이 뛰어나지만 다양한 LDAP 서버 인프라 환경 전반에서 장기적인 상호 운용성을 보장하기 위해 소문자 명명 규칙을 유지하는 것이 모범 사례입니다.

GID 번호 필드를 비워 둡니다. LAM(LDAP Account Manager)은 이 필드에 사용 가능한 다음 값을 자동으로 채우도록 설계되었습니다.

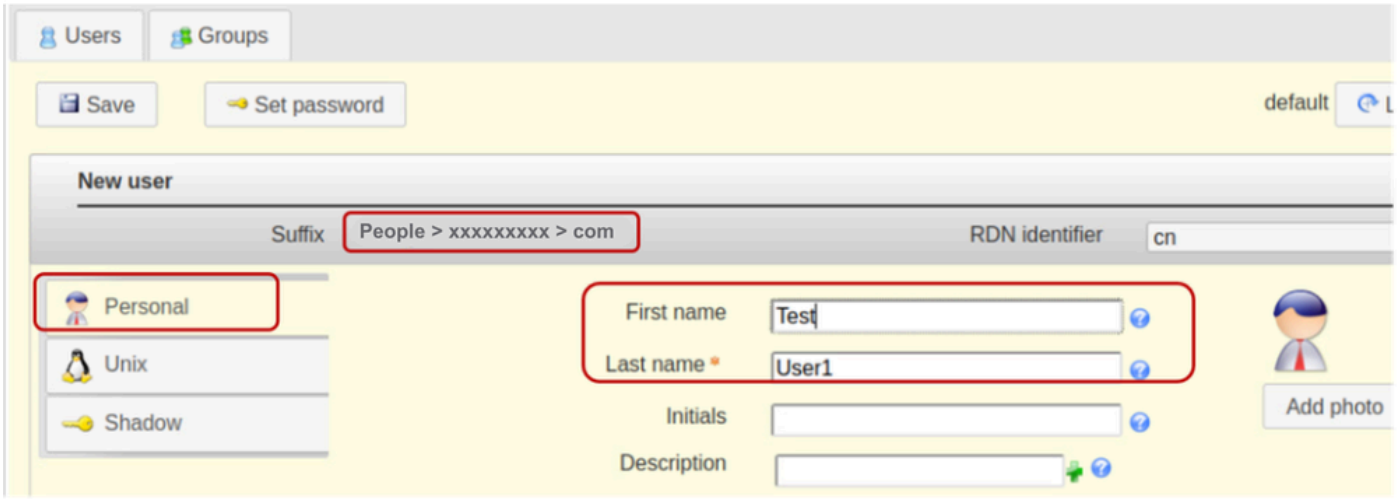
필요한 경우 설명을 입력하고 Save(저장)를 클릭합니다.



"Users(사용자)" 탭을 클릭하여 사용자 계정을 생성하고 "New user(새 사용자)"를 선택합니다.



Personal(개인) 탭에서 "testuser1" 사용자의 필수 필드를 입력합니다.

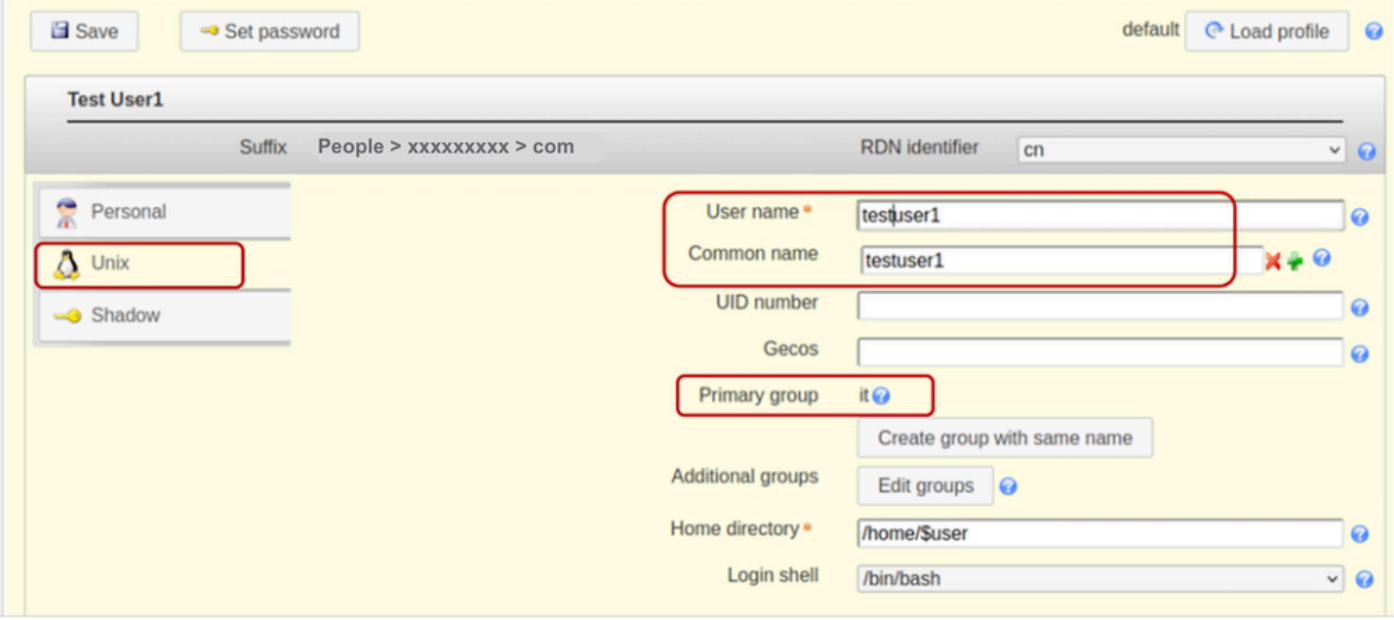


Unix 탭을 선택하고 사용자 이름 필드에 testuser1을 추가합니다. 사용자를 "it" 그룹에 포함시킵니다.

이 데모에서는 "it" 그룹만 존재하므로 이미 미리 채워져 있습니다.

RDN 식별자를 "Common Name"(cn)으로 유지합니다. 이렇게 하면 시스템에서 "User name(사용자 이름)" 필드에 지정된 값을 사용하여 "Common name(일반 이름)" 필드를 자동으로 채울 수 있습니다.

LAM이 사용 가능한 값으로 필드를 자동으로 채우므로 UID Number(UID 번호) 필드를 비워 둡니다.



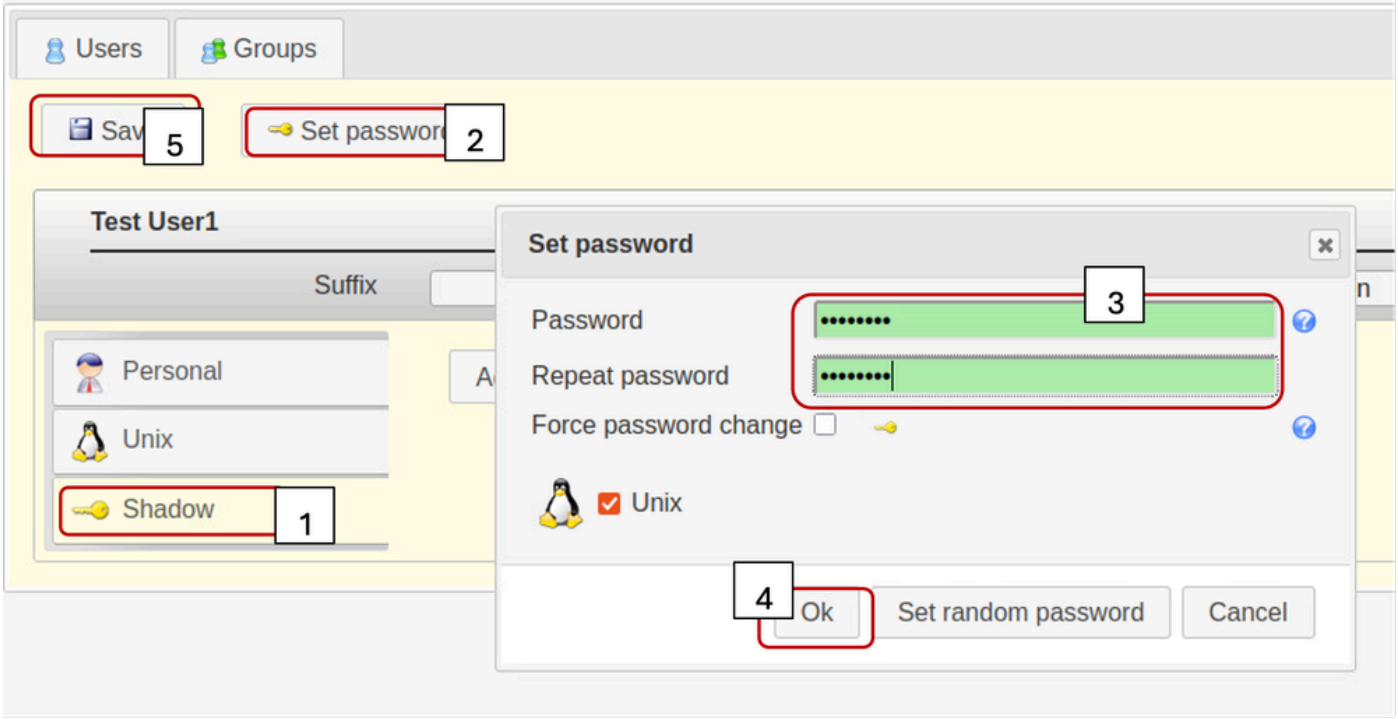
Shadow(그림자) 탭을 선택하고

섀도 계정 확장이 사용되지 않습니다.

"Set Password(비밀번호 설정)"를 클릭합니다.

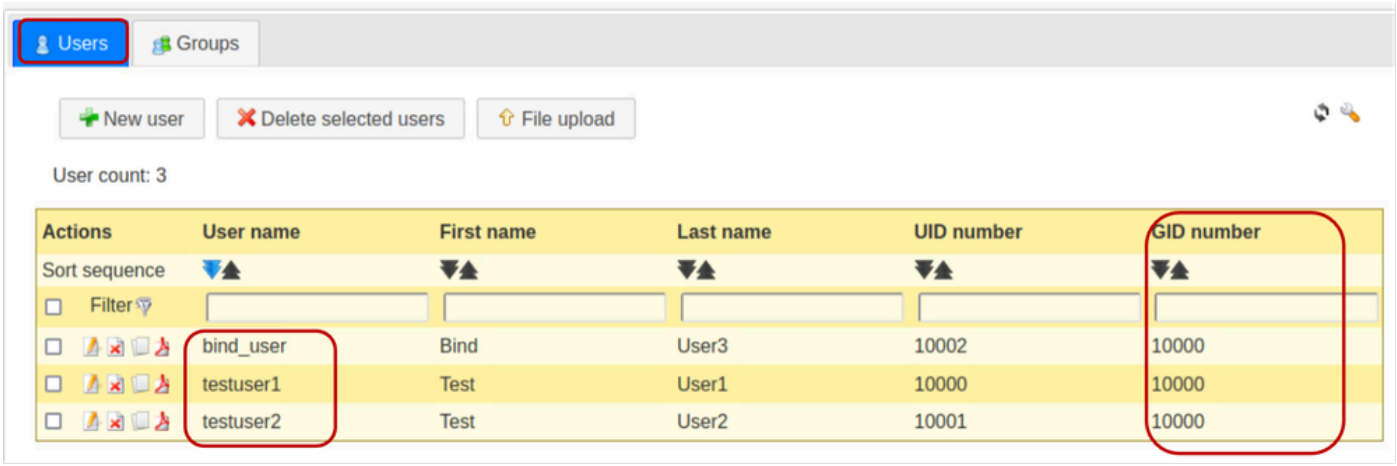
사용자 암호 설정

OK(확인) 및 Save(저장)를 클릭합니다.



"testuser2" 사용자 어카운트 및 "bind_user" 어카운트를 생성하기 위해 앞에서 설명한 지정된 단계를 반복합니다.

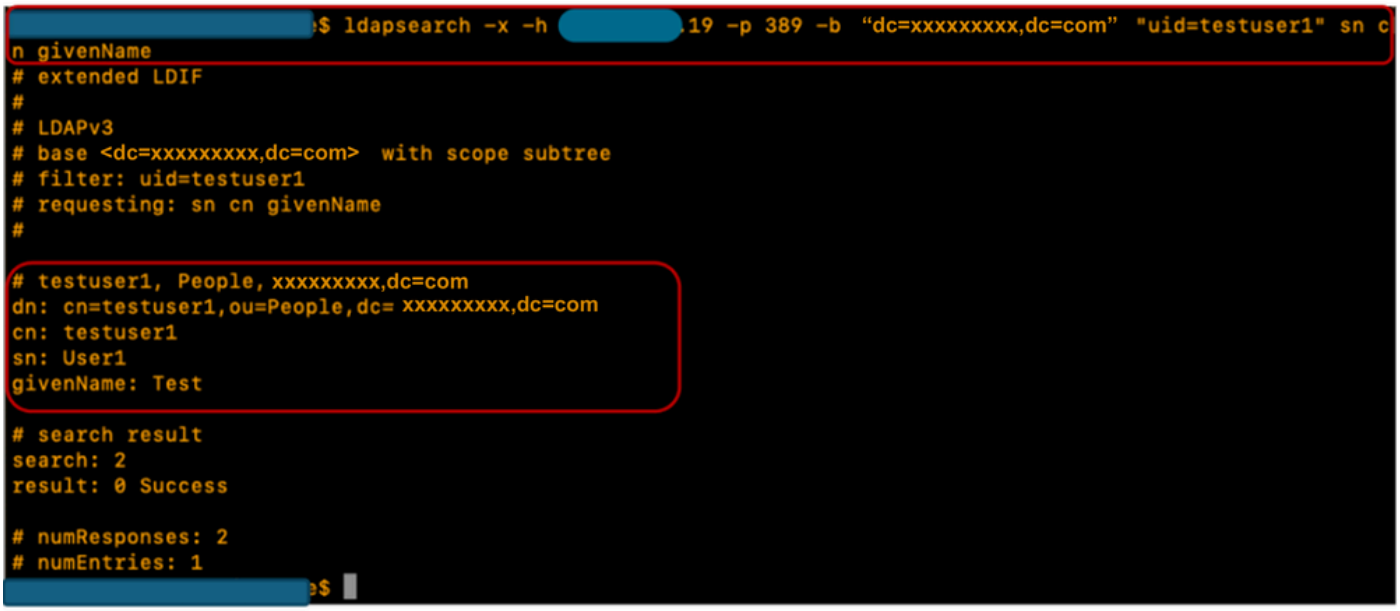
"Users(사용자)" 탭을 클릭하여 원하는 모든 사용자 생성을 확인합니다. (gidNumber 열에 동일한 값이 있으면 생성된 사용자가 동일한 그룹(해당 그룹)에 속해 있음을 확인합니다.)



6단계: 로컬 LDAP 로그인 테스트

OpenLDAP 서버에 연결할 수 있는 다른 Linux 기반 시스템에 로그인합니다.
지정된 ldapsearch 명령을 실행하여 LDAP가 작동하는지 확인합니다.

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



CIMC의 컨피그레이션 매개변수

CIMC에 로그인합니다.

Navigation(탐색) 창에서 Admin(관리), User Management(사용자 관리) 및 LDAP를 선택합니다.

아래와 같이 LDAP 컨피그레이션 매개변수를 채웁니다.

- LDAP 활성화: 선택
- 기본 DN: dc=xxxxxxxxx,dc=com

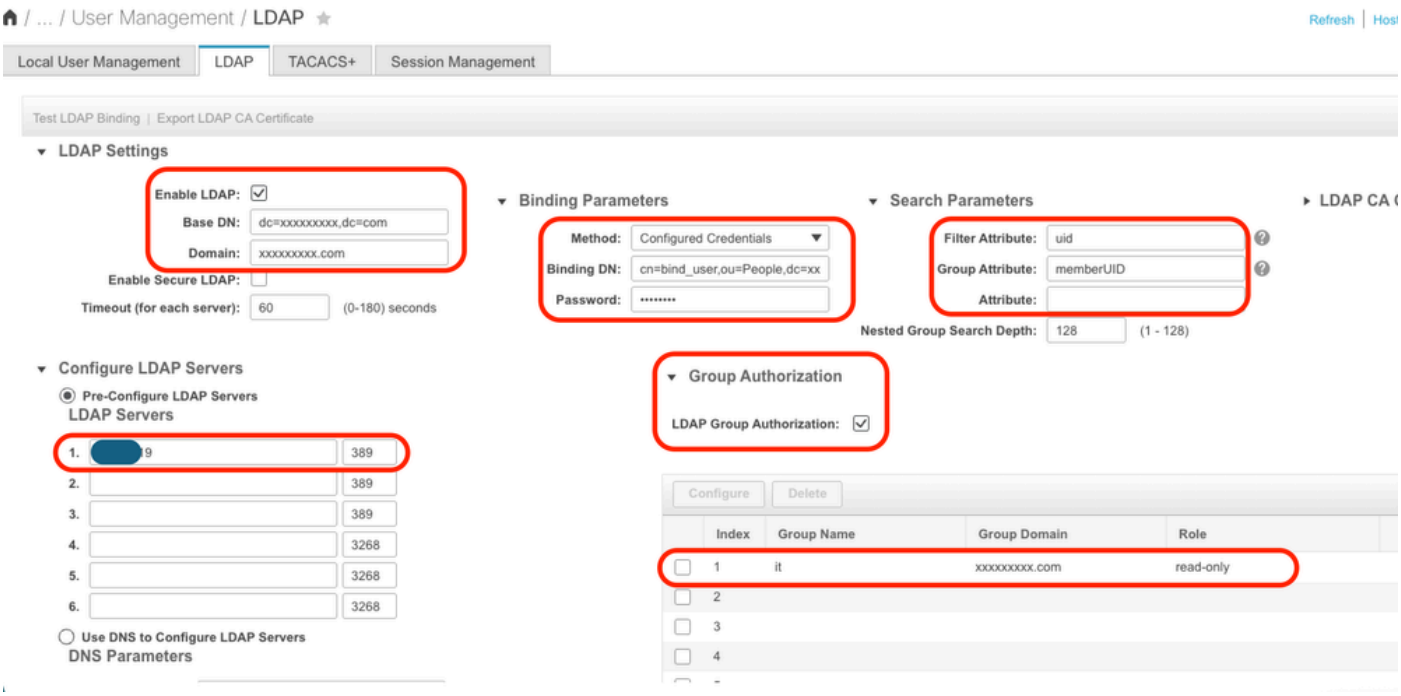
- 도메인: xxxxxxxxx.com

- LDAP 서버: <ldap_server_IP 또는 FQDN> X.X.X.19

- 바인드 매개 변수: "Login Credentials(로그인 자격 증명)" 또는 "Configured Credentials(구성된 자격 증명)"
    - Configured Credentials(구성된 자격 증명)를 사용할 때 LDAP 서버에 구성된 것과 동일하게 bind_user DN을 추가합니다.
        - 예: cn=bind_user,ou=People,dc=xxxxxxxxx,dc=com

- 검색 매개 변수:
    - 필터 특성: "cn" 또는 "uid"
    - 그룹 특성: 멤버UID

- LDAP 그룹 권한 부여 - 선택됨
    - 그룹 이름: IT
    - 그룹 도메인: xxxxxxxxx.com
    - 역할: 읽기 전용(원하는 역할)



컨피그레이션을 저장하고 LDAP 사용자 로그인을 테스트합니다.

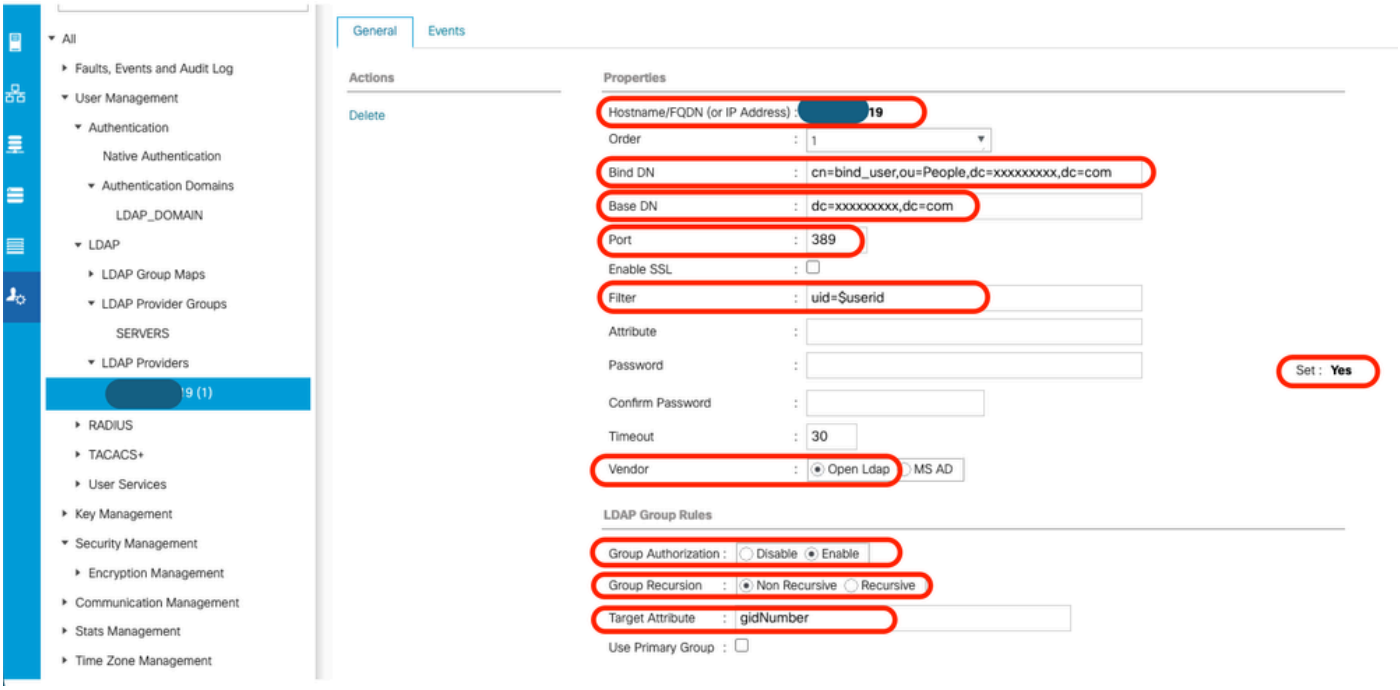
UCS Manager의 컨피그레이션 매개변수

UCS Manager에 로그인합니다.

Navigation(탐색) 창에서 Admin(관리), User Management(사용자 관리) 및 LDAP를 선택합니다.

아래와 같이 LDAP 컨피그레이션 매개변수를 채웁니다.

- LDAP 제공자:
  - 호스트 이름: <LDAP 서버의 FQDN 또는 IP 주소>
  - 바인드 DN: cn=bind_user,ou=People,dc=xxxxxxxxx,dc=com
  - 기본 DN: dc=xxxxxxxxx,dc=com
  - 포트: 389
  - SSL을 활성화합니다. 비활성화됨
  - 필터: uid=$userid
  - 그룹 권한 부여: 활성화됨
  - 그룹 재귀: 비재귀
  - 대상 특성: gid번호
- LDAP 그룹 맵:
  - LDAP 그룹 DN: 10000it" 그룹에 대한 <gidNumber> 입력

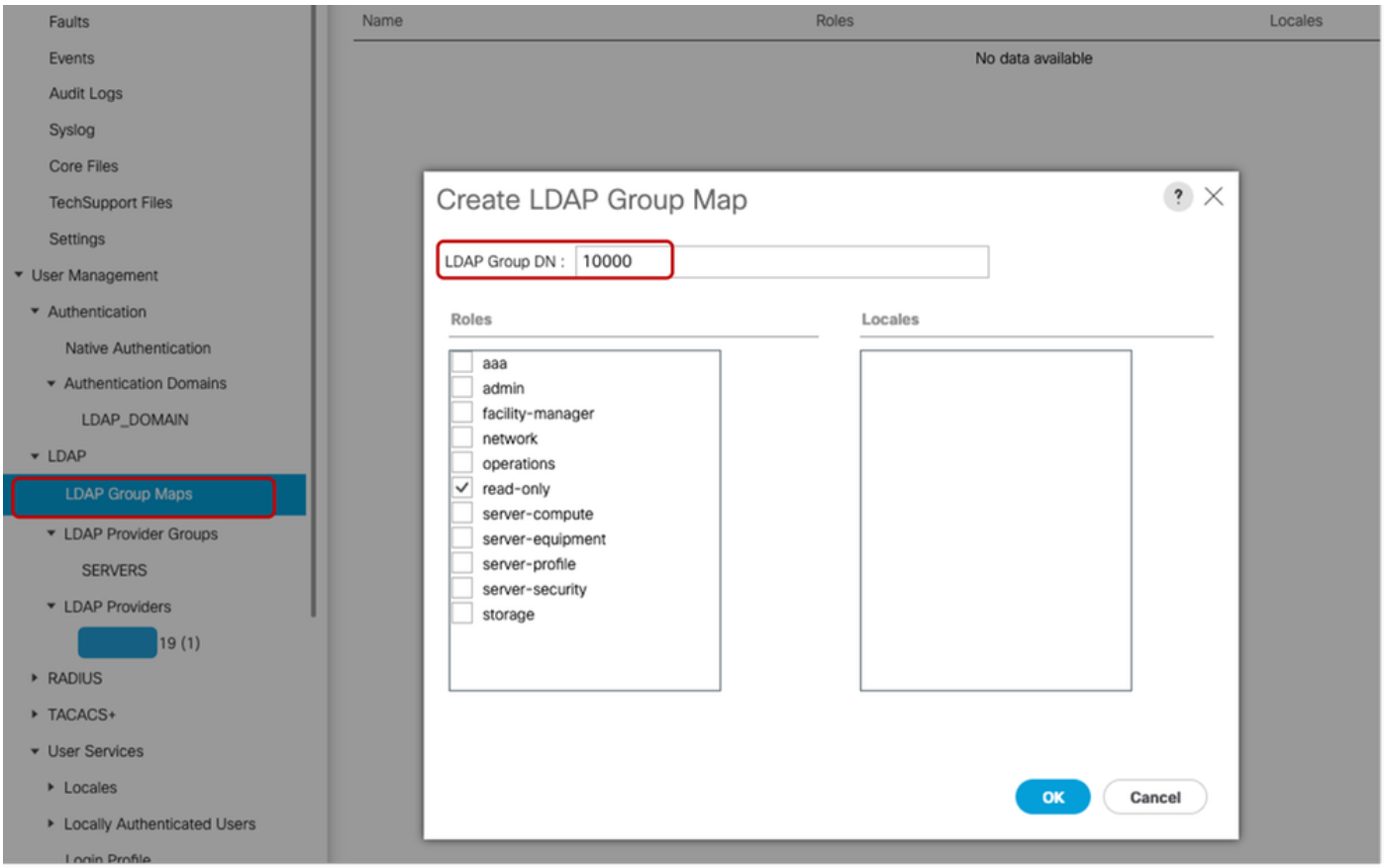


All(모두) >> User Management(사용자 관리) >> LDAP >> LDAP Providers(LDAP 제공자) >> LDAP Group Rules(LDAP 그룹 규칙)에서 UCS Manager의 기본 Target Attribute(대상 특성)는 "memberOf"입니다. 기본적으로 OpenLDAP 서버는 해당 특성을 활성화하지 않으므로 OpenLDAP 서버가 요청한 특성 값을 인식하지 못하므로 Target Attribute 값을 "memberOf"로 설정하거나 비워 두면 사용자 로그인이 실패합니다.

이 예에서는 "Target Attribute" 값이 "gidNumber"로 설정되었습니다.

구성된 LDAP 제공자를 LDAP 제공자 그룹에 추가합니다. 이 데모에서는 "SERVERS" LDAP 제공자 그룹이 생성되었습니다.

"All(모두) >> User Management(사용자 관리) >> LDAP >> LDAP Group Maps(LDAP 그룹 맵)>>"에서 "LDAP 그룹 맵"을 구성할 때 gidNumber 값(이 경우 "10000")이 다음과 같이 "Group DN Map(그룹 DN 맵)"으로 사용됩니다.



LDAP 제공자 그룹을 참조하여 "All(모두) >> User Management(사용자 관리) >> Authentication(인증) >> Authentication Domains(인증 도메인)"에서 LDAP 인증 도메인(LDAP_DOMAIN)을 구성하고 LDAP 사용자 로그인을 테스트합니다.

참고: memberOf 특성이 특정 환경 요구 사항을 충족하거나 "그룹 재귀" 기능을 구현해야 하는 경우, 아래 두 번째 구성 옵션을 사용하는 것이 좋습니다. 이 경우 오버레이 확장이 활성화된 LDAP가 필요합니다.

LDAP 계정 관리자(LAM)는 오버레이 구성을 지원하지만, 이 기능을 사용하려면 적절한 라이센스가 필요합니다.

LAM을 사용하여 LDAP를 구성하는 방법에 대한 자세한 내용은 공식 LDAP 어카운트 매니저 설명서를 참조하십시오.

옵션 2: Ubuntu CLI 도구 및 오버레이를 사용하여 OpenLDAP 구성

UCS Manager 인증에 OpenLDAP를 사용하려면 UCS 시스템(UCS Manager 및 CIMC)이 이해할 수 있는 방식으로 그룹이 사용자와 연결되도록 두 개의 오버레이가 필요합니다.

OpenLDAP 측의 컨피그레이션에는 다음이 필요합니다.

- "memberof" 오버레이: 이 오버레이는 사용자 DN을 쿼리할 경우 그 쿼리의 일부로 memberOf 특성을 요청할 수 있도록 사용자와 그룹 간의 매핑을 생성합니다. Memberof 오버레이가 openLDAP에 추가되지 않는 한 기본적으로 그룹 멤버십에 대한 사용자의 특성은 없습니다
- "참조" 오버레이: 이 오버레이는 그룹 개체의 멤버 특성의 항목이 사용자 개체의 memberOf 특성과 동기화된 상태로 유지되는지 확인하도록 구성됩니다. 이 서비스를 사용하지 않으면 그룹을 수정하지 않고 사용자를 삭제할 경우 고아 DN이 그룹 객체에 남아 있을 수 있습니다. 참조 서비스는 양방향의 일관성을 보장합니다.

1단계: 초기 net-tools 및 Linux 서버 호스트 이름 구성

옵션 1 내에서 1단계를 반복합니다.

2단계: SLAPD 설치

옵션 1에서 2단계를 반복합니다. (옵션 2는 PHP 및 아파치를 설치할 필요가 없으므로 LAM이 없음)

Ubuntu 방화벽을 통과하는 필수 포트를 허용해야 합니다.

3단계: LDAP 서버에 'memberOf' 오버레이 설치

"memberOf" 오버레이가 설치되어 있는지 확인합니다.

```
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" 오버레이를 설치하려면 ldap.memberof.load.ldif라는 .ldif 파일을 만들고(원하는 명명 규칙 사용) 지정된 컨피그레이션을 추가합니다.

```
cat <

    ./ldap.memberof.load.ldif
```

```
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module olcModuleLoad: memberof
EOF
```

지정된 명령을 사용하여 ldap.memberof.load.ldif 파일의 컨피그레이션을 LDAP 프로파일에 추가합니다.

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

linux 배포에 따라 배포 요구 사항에 맞게 memberOf 모듈 및 olcDatabase 항목을 구성합니다.

두 가지 필수 특성 값은 아래와 같이 "olcDatabase={1}mdb" 및 "groupOfNames"입니다.

ldap.memberof.config.ldif 파일을 만들고 특성을 채우고 LDAP 프로필로 내용을 가져옵니다.

```
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
```

4단계: LDAP 서버에 'refint' 오버레이 설치

다음으로 openldap에 대한 참조를 설치합니다.

ldap.refint.load.ldif라는 .ldif 파일을 만들고(원하는 명명 규칙 사용) 지정된 컨피그레이션을 추가합니다.

```
cat <


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

지정된 명령을 사용하여 ldap.refint.load.ldif 파일의 컨피그레이션을 LDAP 프로필로 가져옵니다.

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

그룹과 사용자 간의 참조 무결성을 유지하는 참조 구성

배포 요구 사항에 맞게 참조 모듈 및 해당 olcDatabase 항목을 구성합니다.

ldap.refint.config.ldif 파일을 만들고 그 내용을 LDAP 프로필로 가져옵니다.

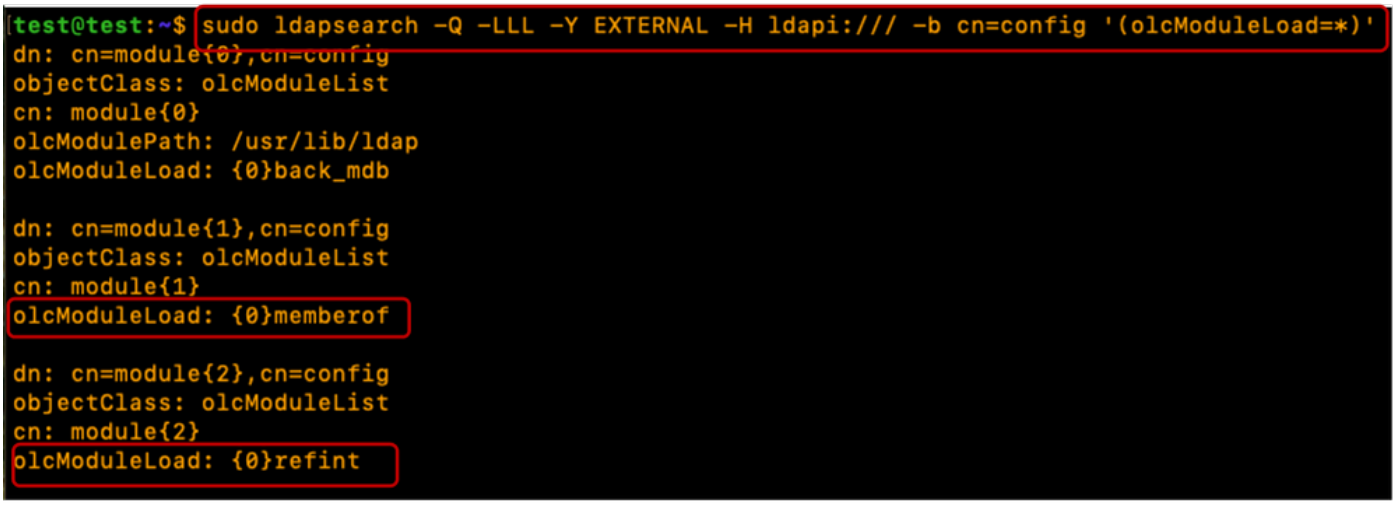
```
cat <


      ./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
```

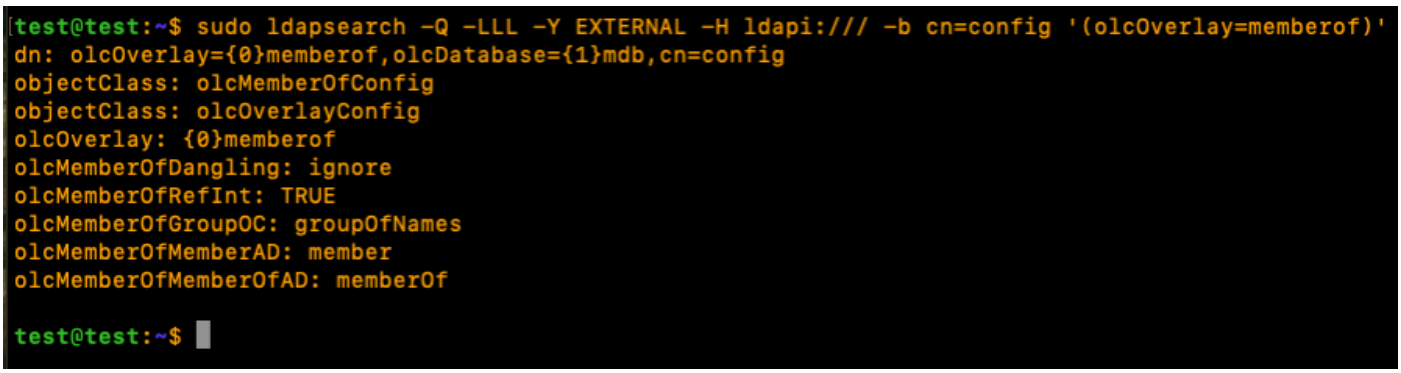
두 플러그인/확장을 모두 설치할 때 지정된 ldapsearch 명령의 출력은 아래에 표시된 출력과 유사합니다.

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



두 플러그인/확장을 모두 구성할 경우 지정된 ldapsearch 명령의 출력은 다음과 같은 출력과 유사합니다.

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




```
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
```

새로 설치된 플러그인/모듈을 사용할 수 있도록 slapd 서비스를 다시 시작합니다.

```
sudo systemctl restart slapd
```

5단계: OU, 사용자 및 그룹 생성

조직 구성 단위(사용자 및 그룹용), 사용자 및 그룹을 생성합니다.

사용자(사람) 및 그룹(그룹) OU를 생성하고 LDAP 프로필로 가져옵니다. "admin" 계정 비밀번호가 필요합니다.

```
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
```

사용자(testuser1, testuser2 및 bind_user)를 생성하고, 사용자를 해당 OU(People)에 매핑하고, gidNumbers를 사용하여 그룹에 추가하고(권장 방법), 사용자를 LDAP 프로필로 가져옵니다.

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:~$
```

그룹(it)을 생성하고 해당 OU(그룹)에 매핑하고 그룹 멤버(testuser1, testuser2)를 연결한 다음 LDAP 프로필로 가져옵니다.

```
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
```

참고: 사용자 또는 그룹을 만드는 동안 memberOf 특성이 명시적으로 정의되지 않은 경우에도 시스템은 자동으로 이 참조를 생성하고 유지 관리합니다. 사용자가 그룹에 연결되면 memberOf 속성은 이러한 멤버십을 자동으로 반영하므로 디렉토리가 현재 액세스 구조와 동기화된 상태로 유지됩니다.

6단계: 로컬 LDAP 로그인 테스트

지정된 명령을 사용하여 LDAP 서버에 대한 사용자 로그인을 확인합니다(환경에 따라 로그인 매개변수 대체).

```
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:~$
```

CIMC의 컨피그레이션 매개변수

CIMC에 로그인합니다.

Navigation(탐색) 창에서 Admin(관리), User Management(사용자 관리) 및 LDAP를 선택합니다.

아래와 같이 LDAP 컨피그레이션 매개변수를 채웁니다.

- LDAP 활성화: 선택
- 기본 DN: dc=xxxxxxxxx,dc=com

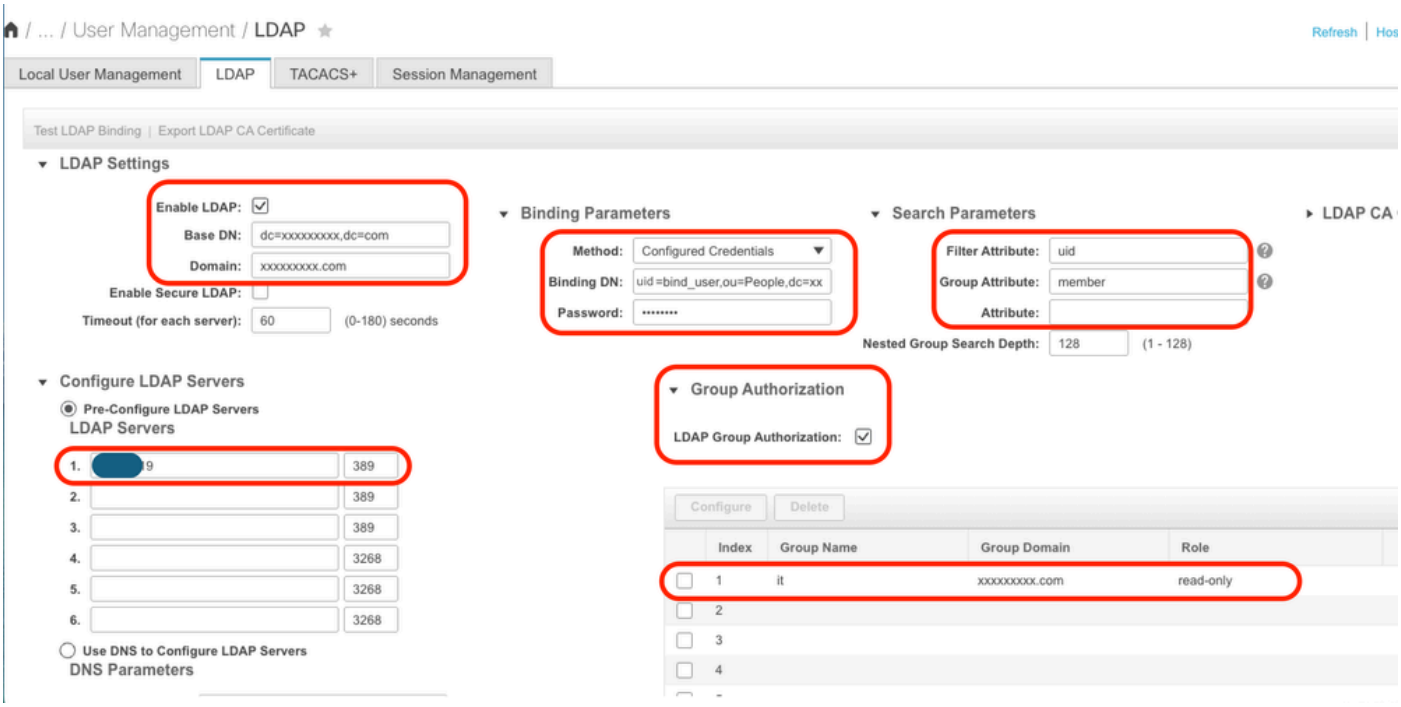
- 도메인: xxxxxxxxx.com

- LDAP 서버: <ldap_server_IP 또는 FQDN> X.X.X.19

- 바인드 매개 변수: "Login Credentials(로그인 자격 증명)" 또는 "Configured Credentials(구성된 자격 증명)"일 수 있습니다.
  - Configured Credentials(구성된 자격 증명)를 사용할 때 LDAP 서버에 구성된 것과 동일하게 bind_user DN을 추가합니다.
  - 예: "cn=bind_user,ou=People,dc=xxxxxxx,dc=com" 또는 "uid=bind_user,ou=People,dc=xxxxxxx,dc=com"

- 검색 매개 변수:
  - 필터 특성: "cn" 또는 "uid"
  - 그룹 특성: 멤버

- LDAP 그룹 권한 부여 - 선택됨
  - 그룹 이름: IT
  - 그룹 도메인: xxxxxxxxx.com
  - 역할: 읽기 전용(기본 설정 역할)



컨피그레이션을 저장하고 LDAP 사용자 로그인을 테스트합니다.

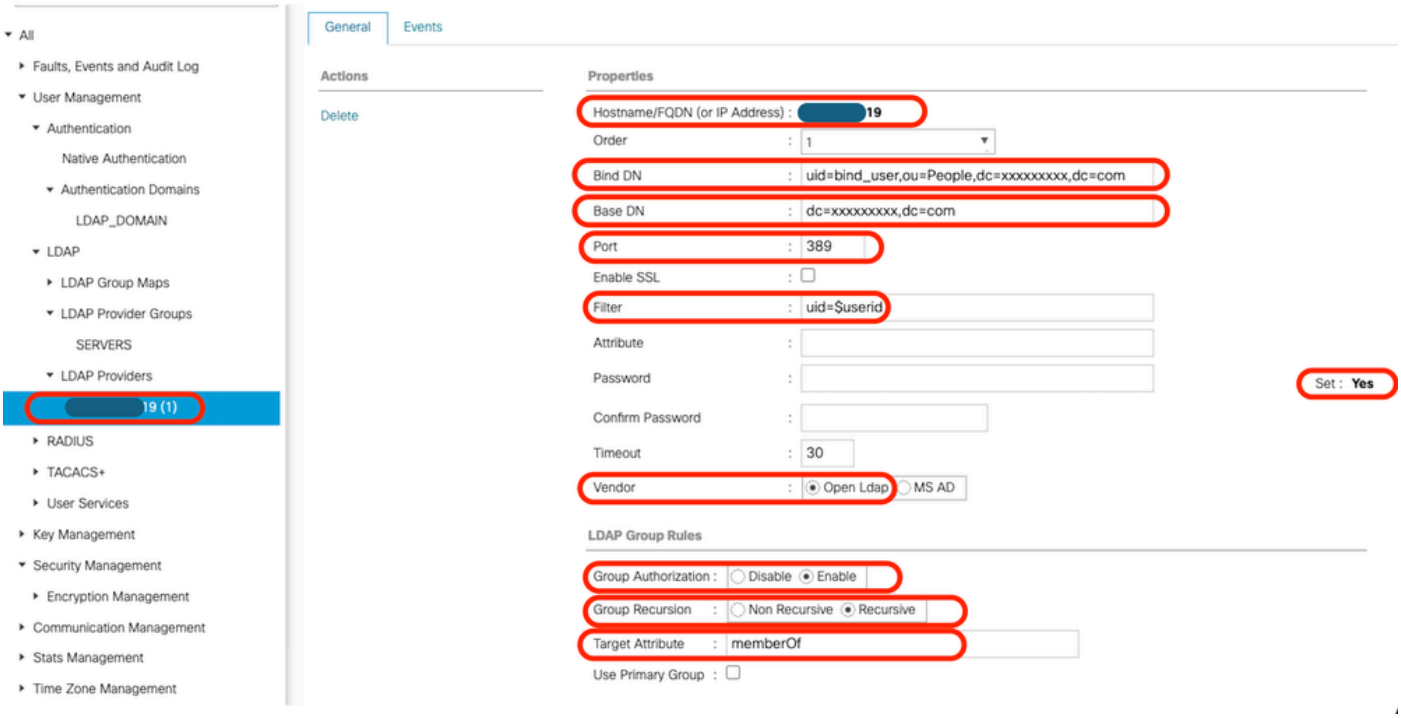
UCS Manager의 컨피그레이션 매개변수

UCS Manager에 로그인합니다.

Navigation(탐색) 창에서 Admin(관리), User Management(사용자 관리) 및 LDAP를 선택합니다.

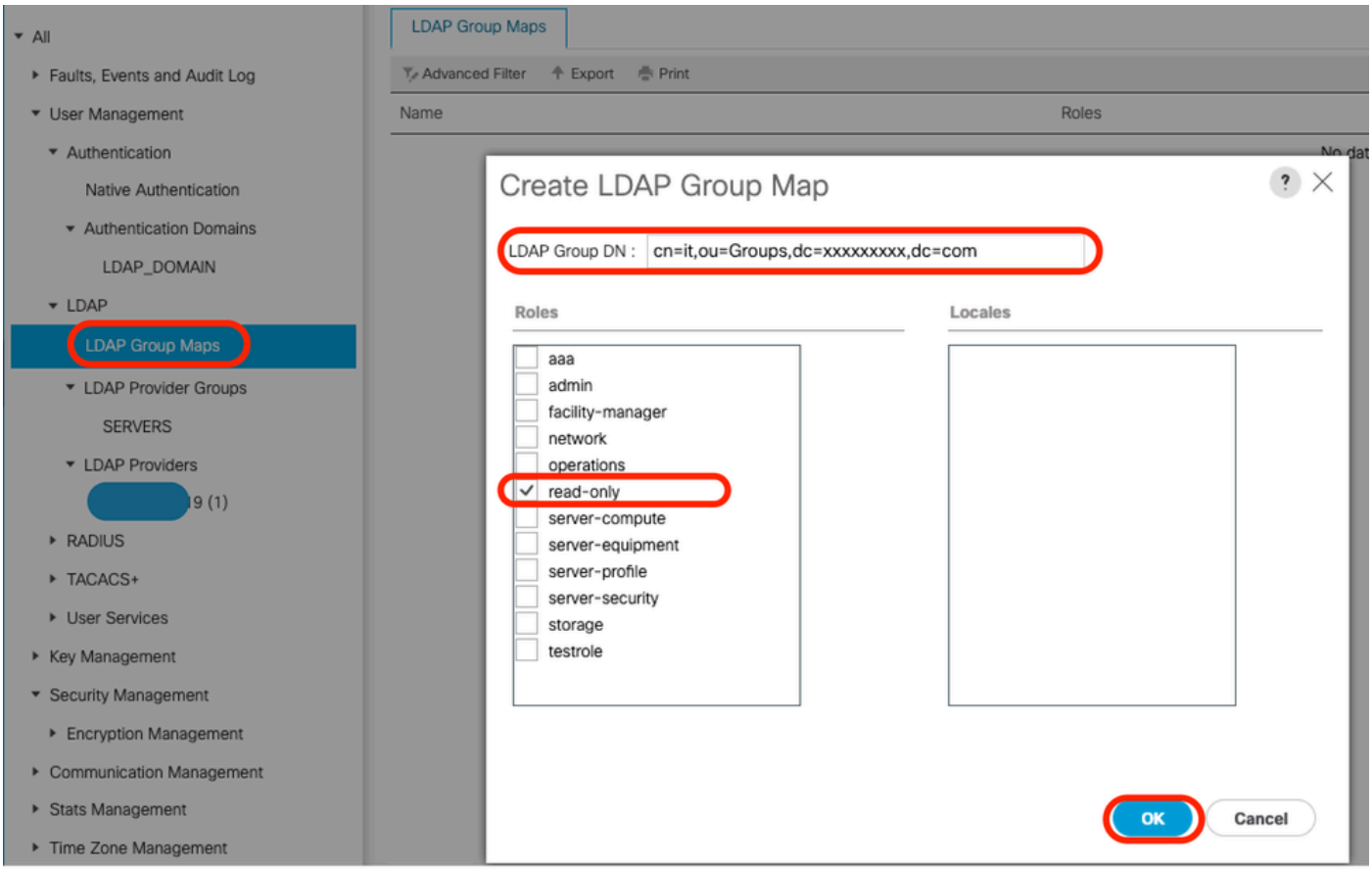
아래와 같이 LDAP 컨피그레이션 매개변수를 채웁니다.

- LDAP 제공자:
  - 호스트 이름: <LDAP 서버의 FQDN 또는 IP 주소>
  - 바인드 DN: uid=bind_user,ou=People,dc=xxxxxxx,dc=com
  - 기본 DN: dc=xxxxxxxxx,dc=com
  - 포트: 389
  - SSL을 활성화합니다. 비활성화됨
  - 필터: uid=$userid
  - 그룹 권한 부여: 활성화됨
  - 그룹 재귀: 재귀
  - 대상 특성: 구성원
- LDAP 그룹 맵:
  - LDAP 그룹 DN: cn=it,ou=Groups,dc=xxxxxxxxx,dc=com



구성된 LDAP 제공자를 LDAP 제공자 그룹에 추가합니다. 이 데모에서는 "SERVERS" LDAP 제공자 그룹이 사용됩니다.

LDAP 서버에서 검색된 "LDAP 그룹 DN"을 추가하는 LDAP 그룹 맵을 구성합니다.



LDAP Provider Groups(SERVERS)를 참조하여 "All(모두) >> User Management(사용자 관리) >> Authentication(인증) >> Authentication Domains(인증 도메인)"에서 LDAP 인증 도메인(LDAP_DOMAIN)을 구성하고 LDAP 사용자 로그인을 테스트합니다.

다음은 별도의 Linux 배포판(CentOS 10)에서 동일한(오버레이 사용) 설정을 살펴보겠습니다

시나리오 2: CentOS Stream 10 - Fedora

LDAP(Lightweight Directory Access Protocol)에 대한 컨피그레이션 절차는 기본 운영 체제 버전에 따라 다릅니다. 이 섹션에서는 CentOS Stream 10에서 LDAP를 구현하는 방법을 중점적으로 살펴봅니다.

많은 Linux 배포에서 OpenLDAP를 사용하지만 CentOS Stream 10과 최신 Fedora 기반 시스템에서는 389 Directory Server(389 DS)를 기본 LDAP 제공자로 사용합니다.

참고: 389 DS는 CentOS 및 Red Hat 에코시스템 내에서 OpenLDAP의 후속 솔루션으로 간주되지만, 두 솔루션은 직접 상호 교환 가능하지 않습니다. 각 디렉토리 구조, 구성 파일 및 운영 환경이 크게 다릅니다.

이 가이드에서는 CentOS Stream 10 환경 내에서 389 DS를 사용하여 LDAP를 성공적으로 구성하는 데 필요한 단계를 제공합니다.

옵션 1: CentOS 스트림 10에서 389 디렉토리 서버를 사용하여 LDAP 구성

1단계: 초기 설정

시나리오 1, 옵션 1에서 단계 1을 반복합니다.

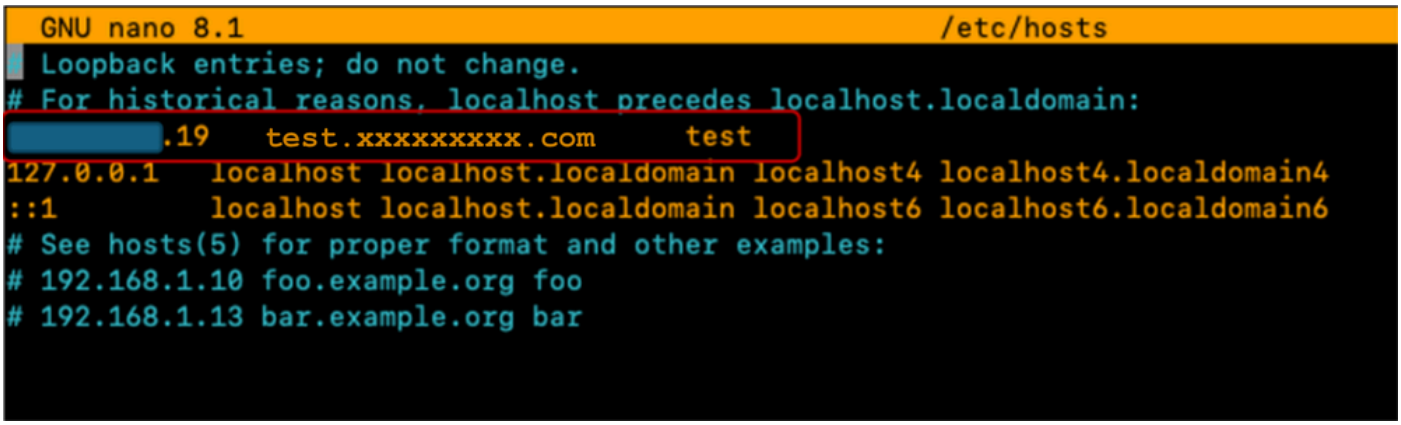
CentOS 시스템은 APT 패키지 관리 제품군을 활용하지 않습니다. CentOS Stream 10에서 필요한 소프트웨어 설치를 수행하려면 dnf(Dandified YUM) 또는 yum 패키지 관리자를 사용합니다

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

"ifconfig" 명령을 사용하여 서버 IP 주소를 확인합니다.

"/etc/hosts" 파일에 서버 IP 주소를 아래 지정된 형식으로 서버 정규화된 도메인 이름(예: 이 실습에서 사용되는 test.xxxxxxx.com) 및 호스트 이름(예: test)과 함께 추가합니다.

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



"/etc/hostname" 파일의 내용을 hostname(테스트)으로 바꾸어 업데이트합니다.

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

이러한 변경 사항을 적용하려면 서버를 재부팅해야 합니다.

```
sudo reboot
```

2단계: EPEL repo 및 389 서버 패키지 설치

EPEL 저장소를 설치하고 업데이트합니다.

389 Directory Server 패키지를 설치합니다.

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

원하는 LDAP 서버 설정 매개변수를 포함하는 디렉토리 템플릿 파일을 생성합니다.

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

생성된 템플릿 파일(ldapconfig.conf)의 내용을 확인합니다.

```
sudo cat ldapconfig.conf
```

ldapconfig.conf 템플릿 파일을 편집합니다.

```
sudo nano ldapconfig.conf
```

지정된 컨피그레이션 항목을 파일에 삽입하고 변경 사항을 저장합니다.

참고:각 환경의 특정 요구 사항이나 요구 사항에 따라 다른 수정이 필요할 수 있습니다.

이 예에서는 이 데모의 기본 컨피그레이션을 다룹니다.

```
[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
```

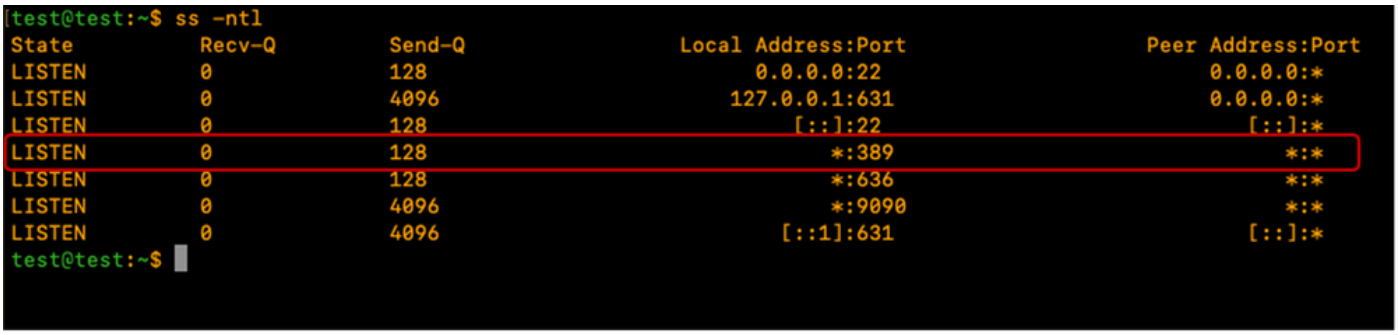
템플릿 파일은 "localhost" 디렉토리 인스턴스의 컨피그레이션 매개변수를 정의합니다. 여기에는 관리 사용자("admin"), 연결된 비밀번호, 도메인 컨텍스트("xxxxxxx.com") 설정이 포함됩니다.

이전에 편집한 템플릿을 사용하여 "localhost" 디렉토리 인스턴스를 생성합니다. 지정된 명령은 LDAP 디렉토리 서버를 생성하고 시작합니다.

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

LDAP 서비스가 서버에서 실행 중인지 확인합니다

```
ss -ntl
```



LDAP(389 및/또는 636)에 필요한 포트를 허용하도록 CentOS 방화벽을 조정합니다.

이 데모에서는 방화벽이 꺼져 있습니다.

```
sudo systemctl stop firewalld
```

지정된 명령을 실행하여 LDAP가 LDAP 서버에서 로컬로 작동하는지 확인하고 표시된 대로 LDAP 출력을 반환하는지 확인합니다.

```
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
```

출력에는 389DS 서버에서 생성한 데모 계정이 포함되어 있습니다. LDAP 서버에서 자동으로 기본 OU를 생성했습니다.

사용자용 사람 OU 및 그룹용 그룹 OU 요구 사항에 따라 추가 OU를 생성할 수 있습니다.

이 데모에서는 기본/자동 생성된 OU가 사용됩니다.

389DS 패키지의 광범위한 사용에 대한 자세한 내용은 공식 389DS 설명서를 참조하십시오.

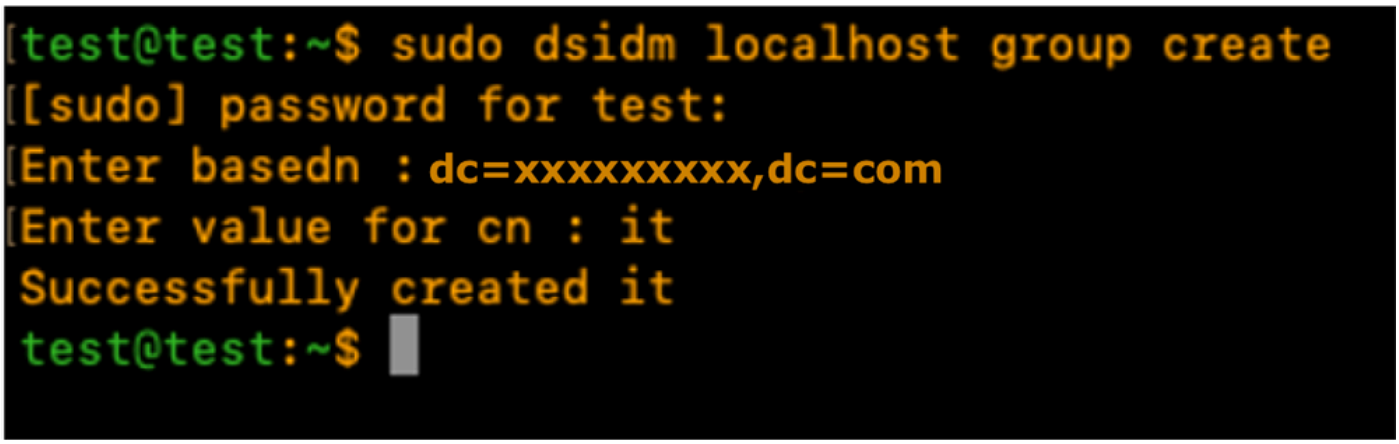
3단계: LDAP 그룹 및 사용자 생성

지정된 sudo dsidm <instance_name> group create 명령을 사용하여 그룹을 생성합니다.

이 데모에서 인스턴스 이름은 "localhost"입니다.

```
sudo dsidm localhost group create
```

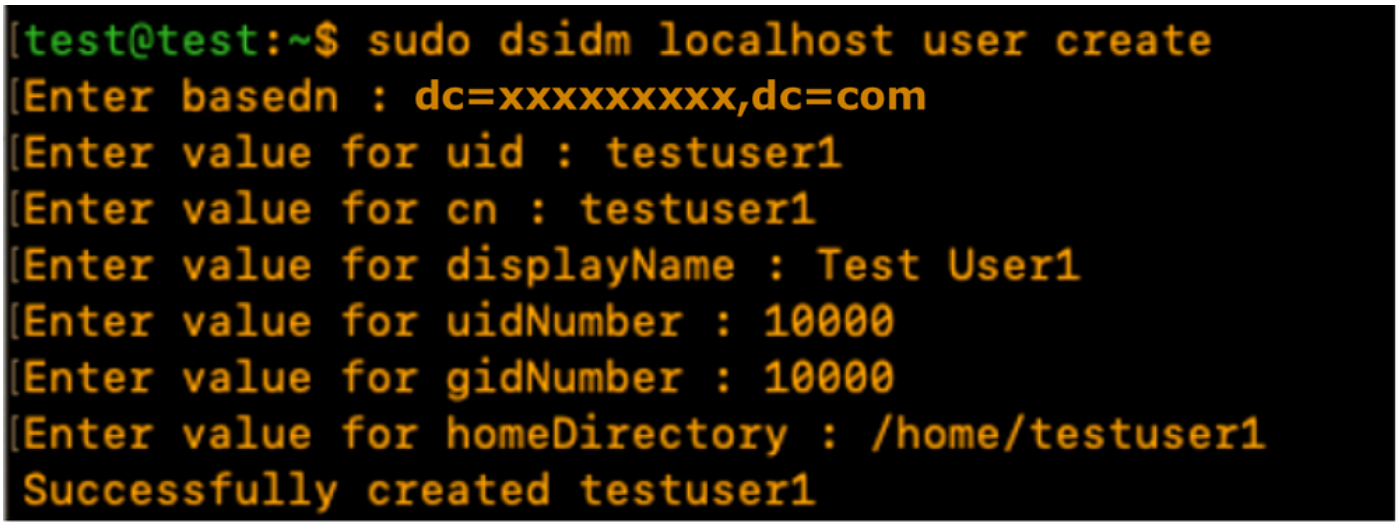
다음과 같이 그룹 세부사항을 채우려면 터미널 프롬프트를 입력합니다.



다음 명령을 사용하여 testuser1 사용자 계정을 생성합니다.

```
sudo dsidm localhost user create
```

터미널 프롬프트를 입력하여 표시된 대로 사용자 세부사항을 채웁니다



지정된 명령을 사용하여 testuser1의 비밀번호를 만들고 CLI 프롬프트를 입력합니다.

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

지정된 명령을 사용하여 그룹에 사용자를 추가합니다. "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
```

사용자 생성 단계를 반복하여 testuser2 및 bind_user를 생성합니다.

참고:각 사용자가 원하는 그룹에 명시적으로 추가되었는지 확인합니다.

이 단계를 생략하면 액세스가 제한되거나 권한 부여가 실패할 수 있습니다.

bind_user 계정은 특정 그룹의 멤버가 될 필요가 없습니다. 독립 실행형 계정으로 구성할 수 있으므로 디렉터리 환경 내에서 관리 및 서비스 수준 액세스를 유연하게 관리할 수 있습니다.

디렉토리 인스턴스를 재시작합니다.

```
sudo dsctl localhost restart
```

4단계: 멤버Of 오버레이 설치

"memberOf" 플러그인을 설치하고 디렉터리 인스턴스를 다시 시작합니다.

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

지정된 명령을 사용하여 "memberOf" 플러그인을 구성합니다. "sudo dsconf <directory_instance> plugin memberof set —scope <base_dn>"

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

지정된 명령을 사용하여 사용자를 유효한 "memberOf" 대상으로 표시: "sudo dsidm <directory_instance> 사용자가 <uid> add:objectclass:nsmemberof를 수정합니다."

```
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:~$
```

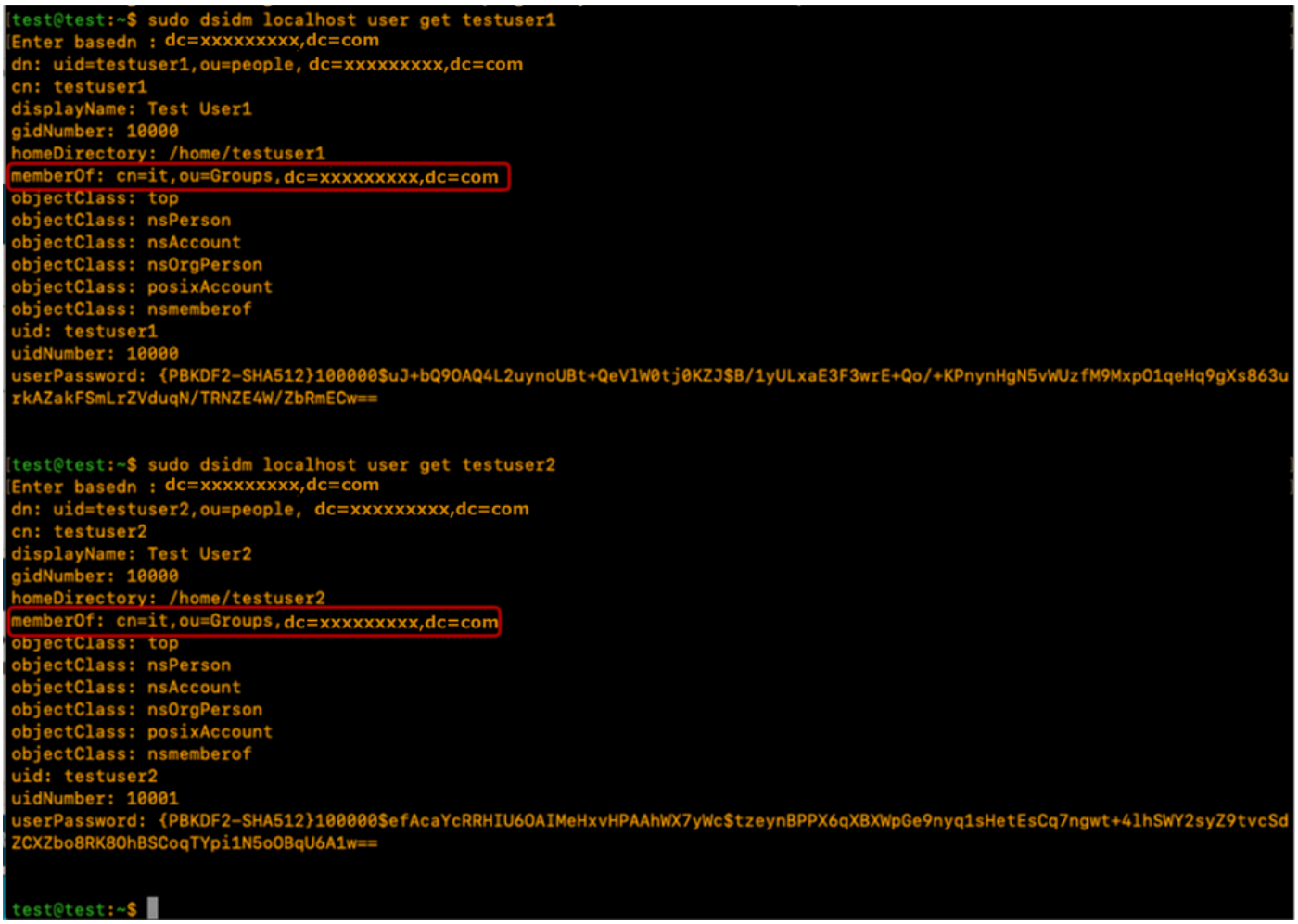
기본 DN에 대한 "memberOf" 수정을 생성합니다. "sudo dsconf <directory_instance> plugin memberof fixup <base_dn>"

```
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:~$
```

사용자 컨피그레이션을 확인합니다.

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



389DS LDAP 서버는 memberOf 특성을 지원하도록 memberOf 플러그인으로 구성됩니다.

CIMC의 컨피그레이션 매개변수

CIMC에 로그인합니다.

Navigation(탐색) 창에서 Admin(관리), User Management(사용자 관리) 및 LDAP를 선택합니다.

아래와 같이 LDAP 컨피그레이션 매개변수를 채웁니다.

- LDAP 활성화: 선택
- 기본 DN: dc=xxxxxxxxx,dc=com

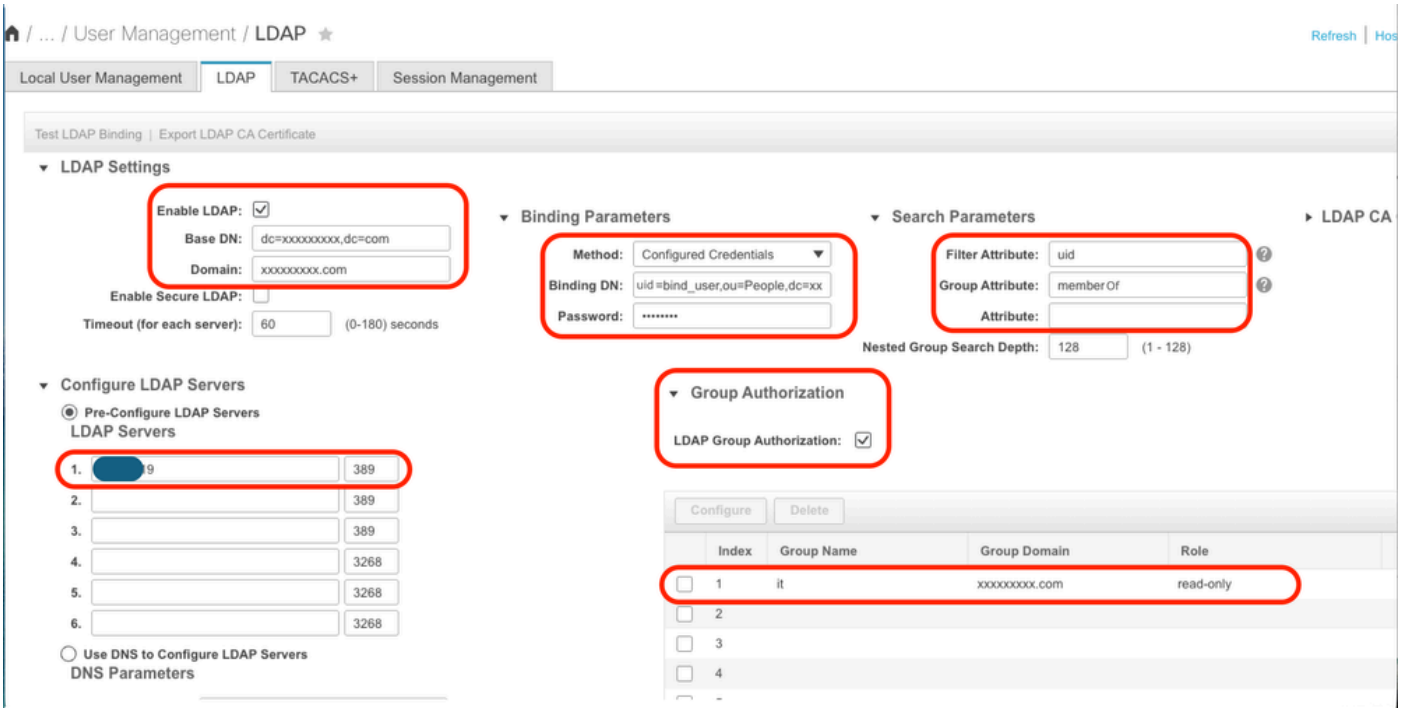
- 도메인: xxxxxxxxx.com

- LDAP 서버: <ldap_server_IP 또는 FQDN> X.X.X.19

- 바인드 매개 변수: "Login Credentials(로그인 자격 증명)" 또는 "Configured Credentials(구성된 자격 증명)"일 수 있습니다.
  - Configured Credentials(구성된 자격 증명)를 사용할 때 LDAP 서버에 구성된 것과 동일하게 bind_user DN을 추가합니다.
    - 예: "cn=bind_user,ou=People,dc=xxxxxxx,dc=com" 또는 "uid=bind_user,ou=People,dc=xxxxxxx,dc=com"

- 검색 매개 변수:
  - 필터 특성: "cn" 또는 "uid"
  - 그룹 특성: 구성원

- LDAP 그룹 권한 부여 - 선택됨
  - 그룹 이름: IT
  - 그룹 도메인: xxxxxxxxx.com
  - 역할: 읽기 전용(기본 설정 역할)



컨피그레이션을 저장하고 LDAP 사용자 로그인을 테스트합니다.

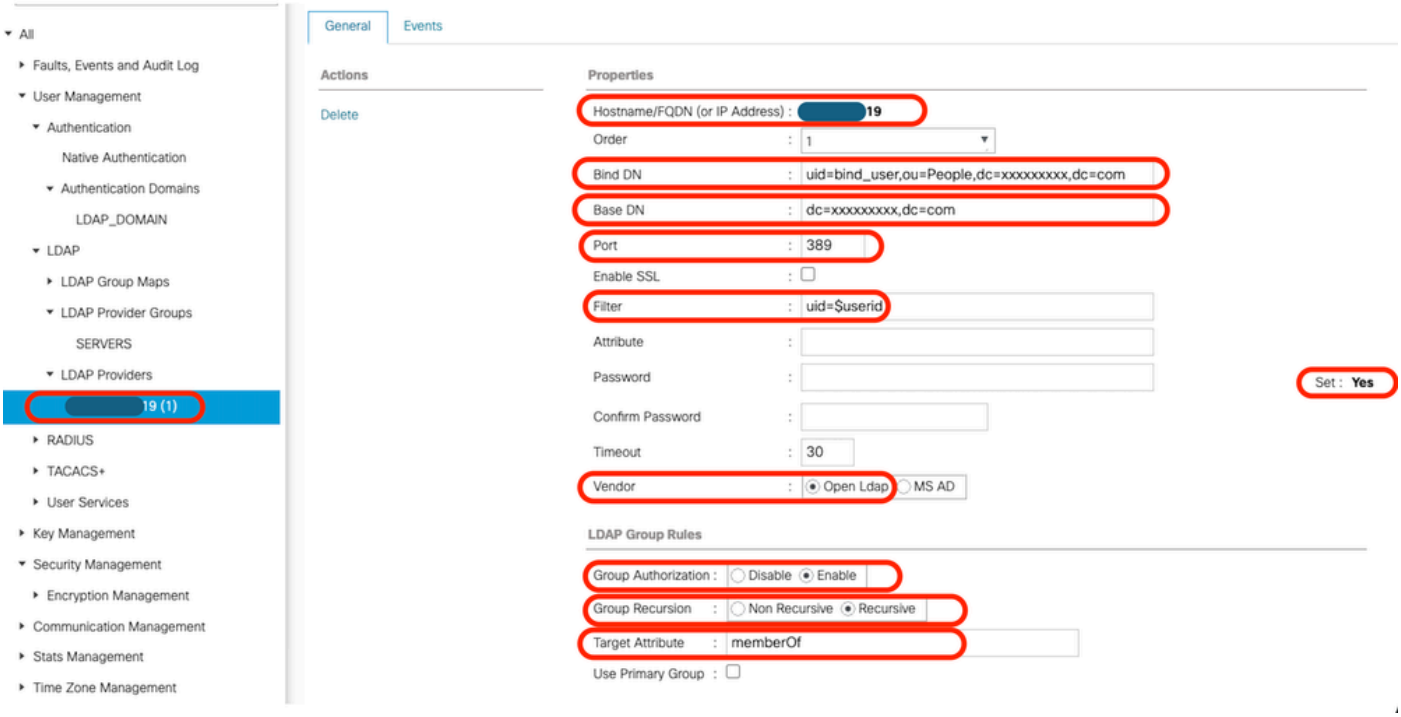
UCS Manager의 컨피그레이션 매개변수

UCS Manager에 로그인합니다.

Navigation(탐색) 창에서 Admin(관리), User Management(사용자 관리) 및 LDAP를 선택합니다.

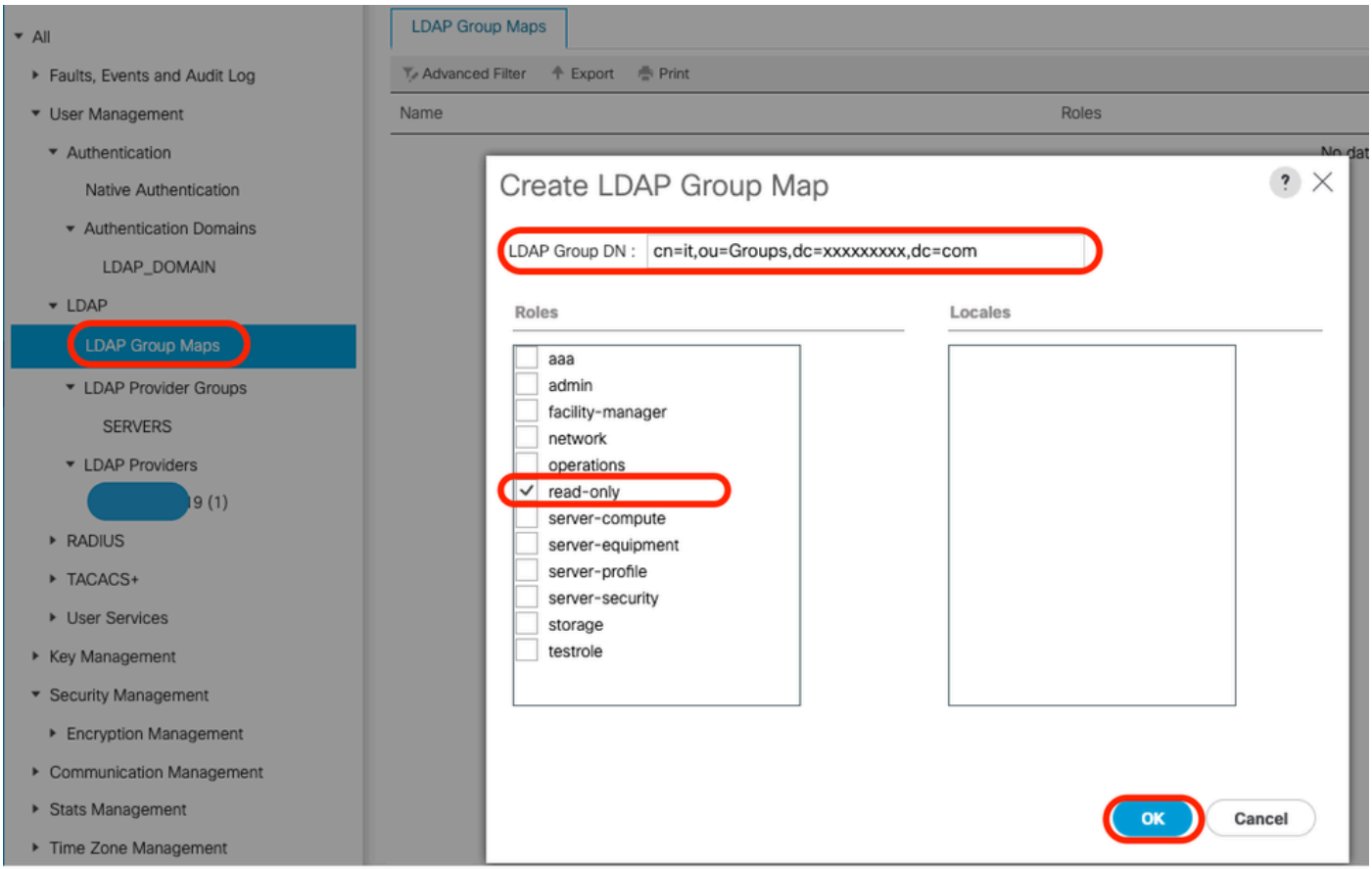
아래와 같이 LDAP 컨피그레이션 매개변수를 채웁니다.

- LDAP 제공자:
  - 호스트 이름: <LDAP 서버의 FQDN 또는 IP 주소>
  - 바인드 DN: uid=bind_user,ou=people,dc=xxxxxxx,dc=com
  - 기본 DN: dc=xxxxxxxxx,dc=com
  - 포트: 389
  - SSL을 활성화합니다. 비활성화됨
  - 필터: uid=$userid
  - 그룹 권한 부여: 활성화됨
  - 그룹 재귀: 재귀
  - 대상 특성: 구성원
- LDAP 그룹 맵:
  - LDAP 그룹 DN: cn=it,ou=Groups,dc=xxxxxxxxx,dc=com



구성된 LDAP 제공자를 LDAP 제공자 그룹에 추가합니다. 이 데모에서는 "SERVERS" LDAP 제공자 그룹이 사용됩니다.

LDAP 서버에서 검색된 "LDAP 그룹 DN"을 추가하는 LDAP 그룹 맵을 구성합니다.



LDAP 제공자 그룹을 참조하여 "All(모두) >> User Management(사용자 관리) >> Authentication(인증) >> Authentication Domains(인증 도메인)"에서 LDAP 인증 도메인(LDAP_DOMAIN)을 구성하고 LDAP 사용자 로그인을 테스트합니다.

결론

이 가이드에서는 필수 구축 시나리오를 다루지만, LDAP 기능을 추가로 탐색하면 디렉토리 성능과 보안을 크게 향상시킬 수 있습니다.

추가 정보, 모범 사례 및 고급 컨피그레이션 세부사항은 지정된 리소스를 참조하십시오.

- OpenLDAP 공식 문서
- LDAP 계정 관리자 - 수동
- 389 디렉토리 서버 설명서
- UCS Manager에서 LDAP 구성
- UCS C Series 서버에서 보안 LDAP 구성

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.