Red Hat Directory Server 12

사용자 관리 및 인증

사용자, 그룹, 역할 및 인증 관련 설정 관리

Last Updated: 2024-08-15

Red Hat Directory Server 12 사용자 관리 및 인증

사용자, 그룹, 역할 및 인증 관련 설정 관리

법적 공지

초록

사용자 액세스 권한, 리소스 제한, 사용자 그룹 및 사용자 역할을 관리하는 방법을 알아봅니다. 암호 및 계정 잠금 정책을 구성하고, 사용자 그룹에 대한 액세스를 거부하며, 바인딩 고유 이름(바인드 DN)에 따라 시스템 리소스를 제한할 수 있습니다.

차례

RED HAT DIRECTORY SERVER에 대한 피드백 제공

Red Hat의 문서 및 제품에 대한 의견을 제공해 주셔서 감사합니다. Red Hat이 어떻게 이를 개선할 수 있는지 알려 주십시오. 이렇게 하려면 다음을 수행합니다.

- Jira (계정 필요)를 통해 Red Hat Directory Server 설명서에 피드백을 제출하려면 다음을 수행합니다.
  1. Red Hat 문제 추적기 로 이동하십시오.
  2. **요약** 필드에 설명 제목을 입력합니다.
  3. **설명** 필드에 개선을 위한 제안을 입력합니다. 문서의 관련 부분에 대한 링크를 포함합니다.
  4. 대화 상자 하단에서 **생성** 을 클릭합니다.
- Jira를 통해 Red Hat Directory Server 제품에 대한 피드백을 제출하기 위해 필요한 경우:
  1. Red Hat 문제 추적기 로 이동하십시오.
  2. **문제 생성** 페이지에서 **다음을** 클릭합니다.
  3. **Summary** 필드를 입력합니다.
  4. **Component** 필드에서 구성 요소를 선택합니다.
  5. 다음을 포함하여 **Description** 필드를 작성합니다.
     a. 선택한 구성 요소의 버전 번호입니다.
     b. 문제 또는 개선을 위한 제안을 재현하는 단계입니다.
  6. **생성**을 클릭합니다.

1장. DIRECTORY SERVER에서 그룹 사용

Directory Server의 그룹에 사용자를 추가할 수 있습니다. 그룹은 디렉터리 항목을 그룹화하는 메커니즘 중 하나로, 사용자 계정 관리를 단순화합니다.

그룹을 사용하는 경우 Directory Server는 이 그룹의 멤버인 사용자의 고유 이름(DN)을 그룹 항목의 멤버십 속성에 저장합니다. 이 특수 속성은 그룹 항목을 생성할 때 선택하는 오브젝트 클래스에 의해 정의됩니다. 그룹 유형에 대한 자세한 내용은 Directory Server의 그룹 유형을 참조하십시오.

그룹이 역할보다 빠릅니다. 그러나 그룹이 역할의 이점을 얻으려면 MemberOf 플러그인을 활성화해야 합니다. 기본적으로 MemberOf 플러그인은 이 사용자가 그룹의 멤버인 경우 **memberOf** 속성을 사용자 항목에 자동으로 추가합니다. 결과적으로 멤버십에 대한 정보는 그룹과 사용자 항목 모두에 저장됩니다. MemberOf 플러그인에 대한 자세한 내용은 사용자 항목의 Listing 그룹 멤버십을 참조하십시오.

1.1. DIRECTORY SERVER의 그룹 유형

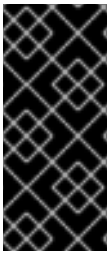
Directory Server에서는 **정적** 또는 **동적** 그룹에 멤버를 추가할 수 있습니다. 각 그룹 유형의 정의에 대한 자세한 내용은 Directory Server의 그룹 정보를 참조하십시오. 그룹 개체 클래스는 멤버십 속성을 정의하고 그룹에 멤버를 추가하려면 그룹 항목의 이 멤버십 속성에 값을 추가해야 합니다.

다음 표에는 그룹 오브젝트 클래스 및 해당 멤버십 속성이 나열되어 있습니다.

그룹 유형	오브젝트 클래스	멤버십 속성
고정	**groupOfNames**	**멤버**
	groupOfUniqueNames	**uniqueMember**
Dynamic	**groupOfURLs**	**memberURL**
	groupOfCertificates	**memberCertificate**

그룹을 생성할 때 사용할 수 있는 오브젝트 클래스:

- **groupOfNames** 는 간단한 그룹입니다. 이 그룹에 항목을 추가할 수 있습니다. **members** 속성에 따라 그룹 멤버십이 결정됩니다. **members** 속성 값은 그룹의 멤버인 사용자 항목의 고유 이름(DN)입니다.
- **groupOfUniqueNames** 는 사용자 DN을 멤버로 나열하지만 DN은 고유해야 합니다. 이 그룹은 자체 참조 그룹 멤버십을 방지합니다. **uniqueMember** 특성에 따라 그룹 멤버십이 결정됩니다.
- **groupOfURLs** 는 LDAP URL 목록을 사용하여 멤버십 목록을 필터링하고 생성합니다. 모든 동적 그룹에는 이 오브젝트 클래스가 필요하며 **groupOfNames** 및 **groupOfUniqueNames** 와 함께 사용할 수 있습니다. **memberURL** 속성은 그룹 멤버십을 결정합니다.
- **groupOfCertificates** 는 LDAP 필터를 사용하여 인증서 이름을 검색하여 그룹 멤버를 식별합니다. 이 그룹에 대한 특수 액세스 권한을 부여할 수 있으므로 **groupOfCertificates** 오브젝트 클래스를 그룹 기반 액세스 제어에 사용합니다. **memberCertificateDescription** 속성에 따라 그룹 멤버십이 결정됩니다.



중요

정적 그룹의 개체 클래스를 동적 오브젝트 클래스 중 하나와 함께 사용하면 그룹이 동적이 됩니다.

MemberOf 플러그인 **은 동적 그룹을 지원하지 않습니다**. 따라서 사용자 항목이 동적 그룹의 필터와 일치하는 경우 플러그인은 **memberOf** 속성을 사용자 항목에 추가하지 않습니다.

1.2. 정적 그룹 생성

명령줄 또는 웹 콘솔을 사용하여 정적 그룹을 생성할 수 있습니다.

1.2.1. 명령줄을 사용하여 정적 그룹 생성

dsidm 유틸리티를 사용하여 **groupOfNames** 오브젝트 클래스로 정적 그룹을 생성합니다. **ldapmodify** 유틸리티를 사용하여 **groupOfUniqueNames** 오브젝트 클래스로 정적 그룹을 생성합니다.

다음 예제에서는 **ou=groups,dc=example,dc=com** 항목에 두 개의 정적 그룹을 생성합니다.

사전 요구 사항

- **ou=groups,dc=example,dc=com** 상위 항목이 있습니다.

절차

- **groupOfNames** 오브젝트 클래스를 사용하여 **cn=simple_group** 그룹을 생성하려면 다음을 실행합니다.

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" group create --cn "simple_group"
Successfully created simple_group
```

dsidm group create 명령은 **ou=group** 하위 항목에서만 그룹을 생성합니다. 다른 항목에서 그룹을 생성하려면 **ldapmodify** 유틸리티를 사용합니다.

- **groupOfUniqueNames** 오브젝트 클래스를 사용하여 **cn=unique_members_group** 그룹을 생성하려면 다음을 실행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x
dn: cn=unique_members_group,ou=groups,dc=example,dc=com
changetype: add
objectClass: top
objectClass: groupOfUniqueNames
cn: unique_members_group
description: A static group with unique members

adding new entry "cn=unique_members_group,ou=groups,dc=example,dc=com"
```

검증

- **dsidm group list** 명령을 사용하여 **groupOfNames** 오브젝트 클래스가 있는 그룹을 나열합니다.

```
# dsidm --basedn "dc=example,dc=com" instance_name group list
simple_group
```

- **dsidm uniquegroup list** 명령을 사용하여 고유한 멤버가 있는 그룹을 나열합니다.

```
# dsidm --basedn "dc=example,dc=com" instance_name uniquegroup list
unique_members_group
```

다음 단계

- 정적 그룹에 멤버 추가.

1.2.2. LDAP 브라우저에서 정적 그룹 생성

웹 콘솔을 사용하여 정적 그룹을 생성할 수 있습니다. 다음 예제에서는 **ou=groups,dc=example,dc=com** 상위 항목에 **static_group** 을 생성합니다.

사전 요구 사항

- **ou=groups,dc=example,dc=com** 상위 항목이 있습니다.
- 웹 콘솔에서 인스턴스에 로그인할 수 있는 권한이 있습니다. 웹 콘솔에 로그인하는 방법에 대한 자세한 내용은 웹 콘솔 을 사용하여 디렉터리 서버에 로그인을 참조하십시오.

절차

1. **LDAP 브라우저** 메뉴로 이동합니다.
2. **트리** 또는 **테이블** 보기를 사용하여 그룹을 만들 상위 항목 **ou=groups,dc=example,dc=com** 을 확장합니다.
3. 옵션 메뉴(Cryostat)를 클릭하고 **새로** 만들기를 선택하여 마법사 창을 엽니다.
4. **Create a group** 을 선택하고 **Next** 를 클릭합니다.
5. groupe **의 기본** 그룹을 선택하고 **Next** 를 클릭합니다.
6. 그룹 이름, 그룹 설명을 추가하고 그룹의 멤버십 특성을 선택합니다.
   - **groupOfNames** 오브젝트 클래스가 있는 그룹의 **멤버입니다**.
   - **groupOfUniqueNames** 오브젝트 클래스를 사용하는 그룹의 **uniquemember** 입니다.
7. **Next**를 클릭합니다.
8. 선택 사항: 그룹에 멤버를 추가하고 **다음을** 클릭합니다.
9. 그룹 정보를 확인하고 **생성** 을 클릭하고 **마침** 을 클릭합니다.

검증

- 접미사 트리에서 새로 생성된 그룹 항목을 확장합니다.

1.3. 정적 그룹에 멤버 추가

웹 콘솔의 명령줄을 사용하여 그룹에 멤버를 추가할 수 있습니다.

1.3.1. 명령줄을 사용하여 정적 그룹에 멤버 추가

정적 그룹에 멤버를 추가하려면 **ldapmodify** 유틸리티를 사용합니다.

사전 요구 사항

- 그룹 항목이 있습니다.
- users 항목이 있습니다.

절차

- **groupOfNames** 오브젝트 클래스를 사용하여 정적 그룹에 멤버를 추가하려면 사용자 고유 이름(DN)을 그룹 항목의 **member** 속성에 대한 값으로 추가합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x
dn: cn=simple_group,ou=groups,dc=example,dc=com
changetype: modify
add: member
member: uid=jsmith,ou=people,dc=example,dc=com

modifying entry "cn=simple_group,ou=groups,dc=example,dc=com"
```

  명령은 **uid=jsmith** 사용자를 **cn=simple_group** 그룹에 추가합니다.

- **groupOfUniqueNames** 오브젝트 클래스를 사용하여 정적 그룹에 멤버를 추가하려면 사용자 고유 이름(DN)을 그룹 항목의 **uniqueMember** 속성에 대한 값으로 추가합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x
dn: cn=unique_members_group,ou=groups,dc=example,dc=com
changetype: modify
add: uniqueMember
uniqueMember: uid=ajonson,ou=people,dc=example,dc=com

modifying entry "cn=unique_members_group,ou=groups,dc=example,dc=com"
```

  명령은 **uid=ajonson** 사용자를 **cn=unique_members_group** 그룹에 추가합니다.

검증

- 그룹 멤버를 나열합니다.

```
# ldapsearch -xLL -D "cn=Directory Manager" -W -b dc=example,dc=com "
(cn=simple_group)"

dn: cn=simple_group,ou=Groups,dc=example,dc=com
objectClass: top
objectClass: groupOfNames
objectClass: nsMemberOf
cn: simple_group
member: uid=jsmith,ou=people,dc=example,dc=com
member: uid=mtomson,ou=people,dc=example,dc=com
```

1.3.2. LDAP 브라우저의 정적 그룹에 멤버 추가

LDAP 브라우저를 사용하여 웹 콘솔의 정적 그룹에 멤버를 추가할 수 있습니다.

사전 요구 사항

- 그룹 항목이 있습니다.
- 사용자 항목이 있습니다.
- 웹 콘솔에서 인스턴스에 로그인되어 있습니다. 웹 콘솔에 로그인하는 방법에 대한 자세한 내용은 웹 콘솔 을 사용하여 디렉터리 서버에 로그인을 참조하십시오.

절차

1. **LDAP 브라우저** 메뉴로 이동합니다.
2. **트리** 또는 **테이블** 보기를 사용하여 멤버를 추가할 그룹 항목을 확장합니다. 예를 들어 **cn=unique_members_group,ou=groups,dc=example,dc=com** 에 멤버를 추가하려고 합니다.
3. 옵션 메뉴(Cryostat)를 클릭하고 **편집을** 선택하여 마법사 창을 엽니다. 창에 현재 멤버 목록이 표시됩니다.
4. **새 멤버 찾기** 탭을 선택합니다.
5. 검색 창에 멤버의 **uid** 또는 **cn** 속성 값의 부분을 입력하고 **Enter** 를 누릅니다. **사용 가능한 멤버** 필드에는 그룹에 추가할 수 있는 사용자 고유 이름(DN)이 표시됩니다.
6. 멤버 DN을 선택하고 화살표(&gt **;)를 클릭하여 멤버의 멤버** 필드로 이동합니다.
7. **멤버 추가** 버튼을 클릭합니다.

검증

- **cn=unique_members_group,ou=groups,dc=example,dc=com** 그룹 항목을 확장하고 항목 세부 정보에서 추가된 사용자를 찾습니다.

1.4. 명령줄을 사용하여 동적 그룹 생성

Directory Server는 명령줄만 사용하여 동적 그룹 생성을 지원합니다. **ldapmodify** 유틸리티를 사용하여 **groupOfURLs** 및 **groupOfCertificates** 오브젝트 클래스로 동적 그룹을 생성합니다.

다음 예제에서는 **ou=groups,dc=example,dc=com** 항목에 두 개의 동적 그룹을 생성합니다.

사전 요구 사항

- **ou=groups,dc=example,dc=com** 상위 항목이 있습니다.

절차

- **groupOfURLs** 오브젝트 클래스를 사용하여 **cn=example_dynamic_group** 그룹을 생성하려면 다음을 실행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x
dn: cn=example_dynamic_group,ou=groups,dc=example,dc=com
changetype: add
objectClass: top
```

```
objectClass: groupOfURLs
cn: example_dynamic_group
description: Example dynamic group for user entries
memberURL: ldap:///dc=example,dc=com??sub?(&(objectclass=person)(cn=*sen))

adding new entry "cn=example_dynamic_group,ou=groups,dc=example,dc=com"
```

명령은 공통 이름(**cn**) 값의 오른쪽 부분에서 **person** object class 및 **sen** 하위 문자열로 멤버를 필터링하는 동적 그룹을 생성합니다.

- **groupOfCertificates** 오브젝트 클래스를 사용하여 **cn=example_certificates_group** 그룹을 생성하려면 다음을 실행합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x
dn: cn=example_certificates_group,ou=groups,dc=example,dc=com
changetype: add
objectClass: top
objectClass: groupOfCertificates
cn: example_certificates_group
description: Example dynamic group for certificate entries
memberCertificateDescription: {ou=people, l=USA, dc=example, dc=com}

adding new entry "cn=example_certificates_group,ou=groups,dc=example,dc=com"
```

이 명령은 인증서 제목 DN에 **ou=people,l=USA,dc=example,dc=com** 이 포함된 멤버를 필터링하는 동적 그룹을 생성합니다.

검증

- **groupOfURLs** 오브젝트 클래스를 사용하여 새로 생성된 그룹을 검색합니다.

```
# ldapsearch -xLLL -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"dc=example,dc=com" "objectClass=groupOfURLs" 1.1

dn: cn=example_dynamic_group,ou=groups,dc=example,dc=com
```

- **groupOfCertificates** 오브젝트 클래스를 사용하여 새로 생성된 그룹을 검색합니다.

```
# ldapsearch -xLLL -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"dc=example,dc=com" "objectClass=groupOfCertificates" 1.1

dn: cn=example_certificates_group,ou=groups,dc=example,dc=com
```

추가 리소스

- 명령줄을 사용하여 LDAP 항목 추가

1.5. 사용자 항목에 그룹 멤버십 나열

그룹은 멤버십 특성을 사용하여 이 그룹에 속하는 항목을 정의합니다. 그룹을 보고 멤버를 쉽게 찾을 수 있습니다. 예를 들어 **groupOfNames** 오브젝트 클래스가 있는 정적 그룹은 해당 멤버의 고유 이름(DN)을 **member** 속성 값으로 저장합니다. 그러나 단일 사용자가 속한 그룹을 빠르게 확인할 수 없습니다. 그룹을 사용하면 사용자 항목에 역할과 달리 사용자 멤버십을 나타내는 항목이 포함되어 있지 않습니다.

이 문제를 해결하려면 MemberOf 플러그인을 사용할 수 있습니다. MemberOf 플러그인은 그룹 항목의 멤버십 속성을 분석하고 그룹을 가리키는 사용자 항목에 **memberOf** 특성을 자동으로 씁니다. 기본적으로 플러그인은 그룹의 **member** 속성을 확인하지만 여러 특성을 사용하여 다른 그룹 유형을 지원할 수 있습니다.

그룹 멤버를 추가하거나 삭제하면 플러그인은 사용자 항목의 **memberOf** 특성을 업데이트합니다. MemberOf 플러그인을 사용하면 특정 사용자 항목에 대해 간단한 검색을 수행하여 사용자가 속한 모든 그룹을 찾을 수 있습니다. MemberOf 플러그인은 모든 그룹에 대한 직접 및 간접 멤버십을 보여줍니다.

중요

MemberOf 플러그인은 **정적** 그룹에 대한 멤버십 속성만 관리합니다.

추가 리소스

- Directory Server의 그룹 유형

1.5.1. MemberOf 플러그인을 사용할 때의 고려 사항

MemberOf 플러그인을 사용하는 경우 다음을 고려하십시오.

- 복제 토폴로지의 MemberOf 플러그인
  복제 토폴로지에서는 다음 두 가지 방법으로 MemberOf 플러그인을 관리할 수 있습니다.
  - 토폴로지의 모든 공급자 및 소비자 서버에서 MemberOf 플러그인을 활성화합니다. 이 경우 모든 복제 계약에서 사용자 항목의 **memberOf** 특성을 복제에서 제외해야 합니다.
  - 토폴로지의 모든 공급자 서버에서만 MemberOf 플러그인을 활성화합니다. 이렇게 하려면 다음을 수행합니다.
    - 복제 계약의 모든 쓰기 사용 공급업체에 **memberOf** 속성 복제를 비활성화해야 합니다.
    - 복제 계약의 모든 소비자 복제본에 **memberOf** 특성 복제를 활성화해야 합니다.
    - 소비자 복제본에서 MemberOf 플러그인을 비활성화해야 합니다.
- 분산 데이터베이스가 있는 MemberOf 플러그인
  데이터베이스 생성 및 유지 관리에 설명된 대로 디렉터리의 하위 트리를 별도의 데이터베이스에 저장할 수 있습니다. 기본적으로 MemberOf 플러그인은 그룹과 동일한 데이터베이스에 저장된 사용자 항목만 업데이트합니다. 모든 데이터베이스에서 사용자를 업데이트하려면 **memberOfAllBackends** 매개변수를 **on** 으로 설정해야 합니다. **memberOfAllBackends** 매개변수 설정에 대한 자세한 내용은 웹 콘솔을 사용하여 각 서버에서 MemberOf 플러그인 구성을 참조하십시오.

1.5.2. MemberOf 플러그인의 필수 오브젝트 클래스

기본적으로 MemberOf 플러그인은 **nsMemberOf** 오브젝트 클래스를 사용자 항목에 추가하여 **memberOf** 특성을 제공합니다. **nsMemberOf** 오브젝트 클래스는 플러그인이 올바르게 작동하기에 충분합니다.

또는 **inetUser**,**inetAdmin**,**inetOrgPerson** 개체 클래스를 포함하는 사용자 항목을 만들 수 있습니다. 이러한 오브젝트 클래스는 **memberOf** 특성을 지원합니다.

중첩 그룹을 구성하려면 그룹이 **extensibleObject** 개체 클래스를 사용해야 합니다.

참고

디렉터리 항목에 필수 특성 작업을 지원하는 오브젝트 클래스가 포함되어 있지 않으면 다음 오류와 함께 실패합니다.

```
LDAP: error code 65 - Object Class Violation
```

1.5.3. MemberOf 플러그인 구문

MemberOf 플러그인을 구성할 때 기본 두 속성을 설정합니다.

- **memberOfGroupAttr**. 그룹 항목에서 폴링할 멤버십 속성을 정의합니다. **memberOfGroupAttr** 속성은 multi-valued입니다. 따라서 플러그인은 여러 유형의 그룹을 관리할 수 있습니다. 기본적으로 플러그인은 **member** 속성을 폴링합니다.
- **memberOfAttr**. 멤버의 사용자 항목에서 생성 및 관리할 멤버십 속성을 정의합니다. 기본적으로 플러그인은 **memberOf** 속성을 사용자 항목에 추가합니다.

또한 플러그인 구문은 플러그인 경로, MemberOf 플러그인, 플러그인 상태 및 기타 구성 매개 변수를 식별하는 함수를 제공합니다.

다음 예제에서는 기본 MemberOf 플러그인 항목 구성을 보여줍니다.

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
cn: MemberOf Plugin
memberofallbackends: off
memberofattr: memberOf
memberofentryscope: dc=example,dc=com
memberofgroupattr: member
memberofskipnested: off
nsslapd-plugin-depends-on-type: database
nsslapd-pluginDescription: memberof plugin
nsslapd-pluginEnabled: off
nsslapd-pluginId: memberof
nsslapd-pluginInitfunc: memberof_postop_init
nsslapd-pluginPath: libmemberof-plugin
nsslapd-pluginType: betxnpostoperation
nsslapd-pluginVendor: 389 Project
nsslapd-pluginVersion: 2.4.5
objectClass: top
objectClass: nsSlapdPlugin
objectClass: extensibleObject
```

설정할 수 있는 예제 및 기타 매개변수의 매개변수에 대한 자세한 내용은 "구성 및 스키마 참조" 설명서 의 MemberOf 플러그인 섹션을 참조하십시오.

1.5.4. MemberOf 플러그인 활성화

명령줄 또는 웹 콘솔을 사용하여 MemberOf 플러그인을 활성화할 수 있습니다.

1.5.4.1. 명령줄을 사용하여 MemberOf 플러그인 활성화

dsconf 유틸리티를 사용하여 MemberOf 플러그인을 활성화합니다.

절차

1. 플러그인을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof enable
```

2. 인스턴스를 다시 시작하십시오.

```
# dsctl instance_name restart
```

검증

- 플러그인 구성 세부 정보를 확인합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof show

dn: cn=MemberOf Plugin,cn=plugins,cn=config
...
nsslapd-pluginEnabled: on
...
```

추가 리소스

- 각 서버에서 MemberOf 플러그인 구성

1.5.4.2. 웹 콘솔을 사용하여 MemberOf 플러그인 활성화

웹 콘솔을 사용하여 MemberOf 플러그인을 활성화할 수 있습니다.

사전 요구 사항

- 웹 콘솔에서 인스턴스에 로그인되어 있습니다. 웹 콘솔에 로그인하는 방법에 대한 자세한 내용은 웹 콘솔 을 사용하여 디렉터리 서버에 로그인을 참조하십시오.

절차

1. **Plugins** 메뉴로 이동합니다.
2. 플러그인 목록에서 **MemberOf** 플러그인을 선택합니다.
3. 플러그인을 활성화하려면 상태를 **ON** 으로 변경합니다.
4. 인스턴스를 다시 시작합니다. 인스턴스를 다시 시작하는 방법은 웹 콘솔을 사용하여 Directory Server 인스턴스 시작 및 중지를 참조하십시오.

추가 리소스

- 웹 콘솔을 사용하여 각 서버에서 MemberOf 플러그인 구성

1.5.5. 각 서버에서 MemberOf 플러그인 구성

MemberOf 플러그인의 구성을 복제하지 않으려면 명령줄 또는 웹 콘솔을 사용하여 각 서버에서 수동으로 플러그인을 구성합니다.

1.5.5.1. 명령줄을 사용하여 각 서버에서 MemberOf 플러그인 구성

기본적으로 MemberOf 플러그인은 그룹 항목에서 **member** membership 속성을 읽고 **memberOf** 속성을 사용자 항목에 추가합니다. 그러나 그룹의 다른 멤버십 속성을 읽고, 사용자 항목에 다른 특성을 추가하고, 중첩된 그룹을 건너뛰고, 모든 데이터베이스 및 기타 설정에서 작동하도록 플러그인을 구성할 수 있습니다.

예를 들어 MemberOf 플러그인에서 다음을 수행하도록 합니다.

- 그룹 항목에서 **uniqueMember** 특성을 읽고 멤버십을 식별합니다.
- 중첩된 그룹을 건너뜁니다.
- 모든 데이터베이스에서 사용자 항목을 검색합니다.

사전 요구 사항

- MemberOf 플러그인을 활성화했습니다. 자세한 내용은 MemberOf 플러그인 활성화를 참조하십시오.

절차

1. 선택 사항: MemberOf 플러그인 구성을 표시하여 플러그인이 현재 그룹 항목에서 읽고 있는 멤버십 특성을 확인합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof show

...
memberofgroupattr: member
...
```

   플러그인은 현재 그룹 항목에서 **member** 속성을 읽고 멤버를 검색합니다.

2. 플러그인 구성에서 **uniqueMember** 특성을 **memberOfGroupAttr** 매개변수 값으로 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof set --groupattr uniqueMember
```

   memberOfGroupAttr 매개변수는 multi-valued이며 모두 **--groupattr** 매개변수에 전달하여 여러 값을 설정할 수 있습니다. 예를 들면 다음과 같습니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof set --groupattr member uniqueMember
```

3. 분산 데이터베이스를 사용하는 환경에서 로컬 데이터베이스 대신 모든 데이터베이스의 사용자 항목을 검색하도록 플러그인을 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof set --allbackends on
```

   명령은 **memberOfAllBackends** 매개변수를 설정합니다.

4. 중첩 그룹을 건너뛰도록 플러그인을 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof set --skipnested on
```

이 명령은 **memberOfSkipNested** 매개변수를 설정합니다.

5. 선택 사항: 기본적으로 플러그인은 **memberOf** 특성을 허용하는 오브젝트 클래스가 없는 경우 사용자 항목에 **nsMemberOf** 오브젝트 클래스를 추가합니다. **nsMemberOf** 대신 **inetUser** 오브젝트 클래스를 사용자 항목에 추가하도록 플러그인을 구성하려면 다음을 실행합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof set --autoaddoc inetUser
```

이 명령은 **memberOfAutoAddOC** 매개변수를 설정합니다.

6. 인스턴스를 다시 시작하십시오.

```
# dsctl instance_name restart
```

검증

- MemberOf 플러그인 구성을 확인합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof show
dn: cn=MemberOf Plugin,cn=plugins,cn=config
cn: MemberOf Plugin
memberofallbackends: on
memberofattr: memberOf
memberofautoaddoc: inetuser
memberofentryscope: dc=example,dc=com
memberofgroupattr: uniqueMember
memberofskipnested: on
...
nsslapd-pluginEnabled: on
...
```

추가 리소스

- **수정** 작업을 사용하여 사용자 항목에서 **memberOf** 속성 값 업데이트
- 서버 간 MemberOf 플러그인 구성 공유
- MemberOf 플러그인의 범위 설정
- MemberOf 플러그인의 필수 오브젝트 클래스

1.5.5.2. 웹 콘솔을 사용하여 각 서버에서 MemberOf 플러그인 구성

기본적으로 MemberOf 플러그인은 그룹 항목에서 **member** membership 속성을 읽고 **memberOf** 속성을 사용자 항목에 추가합니다. 그러나 그룹에서 다른 멤버십 속성을 읽고, 중첩된 그룹을 건너뛰고, 웹 콘솔을 사용하여 모든 데이터베이스 및 기타 설정에서 작업하도록 플러그인을 구성할 수 있습니다.

예를 들어 MemberOf 플러그인에서 다음을 수행하도록 합니다.

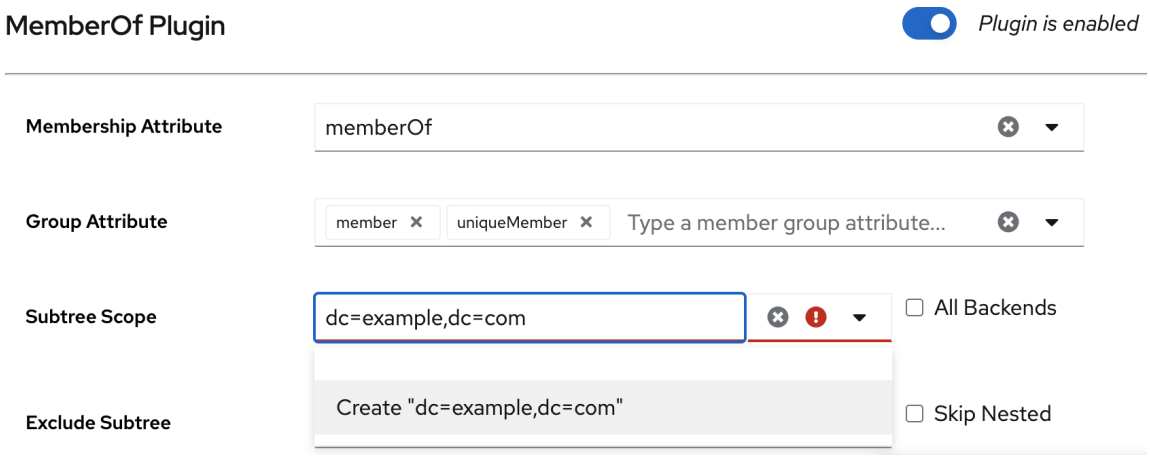
- 멤버십을 식별하기 위해 그룹 항목에서 **멤버** 및 **uniqueMember** 특성을 읽습니다.
- 플러그인의 범위를 **dc=example,dc=com** 으로 설정합니다.
- 중첩된 그룹을 건너뜁니다.
- 모든 데이터베이스에서 사용자 항목을 검색합니다.

사전 요구 사항

- 웹 콘솔에서 인스턴스에 로그인되어 있습니다. 웹 콘솔에 로그인하는 방법에 대한 자세한 내용은 웹 콘솔 을 사용하여 디렉터리 서버에 로그인을 참조하십시오.
- MemberOf 플러그인을 활성화했습니다. 자세한 내용은 MemberOf 플러그인 활성화를 참조하십시오.

절차

1. **LDAP 브라우저** 메뉴로 이동합니다.
2. 플러그인 목록에서 **MemberOf** 플러그인을 선택합니다.
3. **Group Attribute** 필드에 **uniqueMember** 특성을 추가합니다.
4. 플러그인의 범위를 **dc=example,dc=com**:으로 설정합니다.
   a. **dc=example,dc=com** 을 **Subtree Scope** 필드에 입력합니다.
   b. 드롭다운 목록에서 **Create "dc=example,dc=com"** 을 클릭합니다.



5. 선택 사항: 제외하도록 하위 트리를 설정합니다. 예를 들어, 플러그인이 **ou=private,dc=example,dc=com** 하위 트리에서 작동하지 않도록 합니다.
   a. **Exclude Subtree** 필드에 **ou=private,dc=example,dc=com** 을 입력합니다.
   b. 드롭다운 목록에서 **Create "ou=private,dc=example,dc=com"** 을 클릭합니다.
6. **모든 백엔드를** 선택하여 로컬 데이터베이스뿐만 아니라 모든 데이터베이스의 사용자 항목을 검색하도록 플러그인을 구성합니다.
7. 중첩 그룹을 건너뛰도록 플러그인을 구성하려면 **Skip Nested** 를 선택합니다.
8. **Save Config** 를 클릭합니다.

추가 리소스

- 서버 간 MemberOf 플러그인 구성 공유
- 명령줄을 사용하여 MemberOf 플러그인의 범위 설정
- memberOf 플러그인
- **수정** 작업을 사용하여 사용자 항목에서 **memberOf** 속성 값 업데이트

1.5.6. 서버 간 MemberOf 플러그인 구성 공유

기본적으로 각 서버는 MemberOf 플러그인의 자체 구성을 저장합니다. 플러그인의 공유 구성을 사용하면 각 서버에서 수동으로 플러그인을 구성하지 않고 동일한 설정을 사용할 수 있습니다. Directory Server는 **cn=config** 접미사 외부에 공유 구성을 저장하고 복제합니다.

예를 들어 플러그인 공유 구성을 **cn=shared_MemberOf_config,dc=example,dc=com** 항목에 저장하려고 합니다.

중요

공유 구성을 활성화하면 플러그인은 **cn=MemberOf Plugin,cn=plugins,cn=config** 플러그인 항목에 설정된 모든 매개변수를 무시하고 공유 구성 항목의 설정만 사용합니다.

사전 요구 사항

- 복제 토폴로지의 모든 서버에서 MemberOf 플러그인을 활성화했습니다. 자세한 내용은 MemberOf 플러그인 활성화를 참조하십시오.

절차

1. 서버에서 공유 구성 항목을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof
config-entry add "cn=shared_MemberOf_config,dc=example,dc=com" --attr memberOf
--groupattr member

Successfully created the cn=shared_MemberOf_config,dc=example,dc=com
MemberOf attribute nsslapd-pluginConfigArea (config-entry) was set in the main plugin
config
```

   이 명령은 **nsslapd-pluginConfigArea** 속성 값을 **cn=shared_MemberOf_config,dc=example,dc=com** 으로 설정합니다.

2. 인스턴스를 다시 시작하십시오.

```
# dsctl instance_name restart
```

3. 공유 구성을 사용해야 하는 복제 토폴로지의 다른 서버에서 공유 구성을 활성화합니다.

   a. 공유 구성을 저장하는 구성 항목의 DN(고유 이름)을 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server2.example.com plugin memberof
set --config-entry cn=shared_MemberOf_config,dc=example,dc=com
```

b. 인스턴스를 다시 시작하십시오.

```
# dsctl instance_name restart
```

검증

1. MemberOf 플러그인이 공유 구성을 사용하는지 확인합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server1.example.com plugin memberof show

dn: cn=MemberOf Plugin,cn=plugins,cn=config
cn: MemberOf Plugin
...
nsslapd-pluginConfigArea: cn=shared_MemberOf_config,dc=example,dc=com
...
```

2. 선택 사항: 공유 구성 설정을 확인합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server1.example.com plugin memberof config-entry show "cn=shared_MemberOf_config,dc=example,dc=com"

dn: cn=shared_MemberOf_config,dc=example,dc=com
cn: shared_MemberOf_config
memberofattr: memberOf
memberofgroupattr: member
objectClass: top
objectClass: extensibleObject
```

추가 리소스

- nsslapd-pluginConfigArea

1.5.7. MemberOf 플러그인의 범위 설정

여러 백엔드 또는 다중 유형 접미사를 구성한 경우 **memberOfEntryScope** 및 **memberOfEntryScopeExcludeSubtree** 매개변수를 사용하여 **MemberOf** 플러그인의 접미사를 설정할 수 있습니다.

사용자를 그룹에 추가하는 경우, MemberOf 플러그인은 사용자와 그룹이 플러그인 범위에 있는 경우에만 **memberOf** 특성을 그룹에 추가합니다.

예를 들어 다음 절차에서는 **dc=example,dc=com** 의 모든 항목에서 작동하도록 MemberOf 플러그인을 구성하지만 **ou=private,dc=example,dc=com** 의 항목을 제외합니다.

사전 요구 사항

- 복제 토폴로지의 모든 서버에서 MemberOf 플러그인을 활성화했습니다. 자세한 내용은 MemberOf 플러그인 활성화를 참조하십시오.

절차

1. MemberOf 플러그인의 scope 값을 **dc=example,dc=com** 으로 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof set --scope "dc=example,dc=com"
```

2. **ou=private,dc=example,dc=com** 의 항목을 제외 :

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof set --exclude "ou=private,dc=example,com"
```

--scope DN 매개변수를 사용하여 사용자 항목을 범위 외부에서 이동한 경우:

- MemberOf 플러그인은 **member** 와 같은 멤버십 속성을 그룹 항목에서 업데이트하여 사용자 DN 값을 제거합니다.
- MemberOf 플러그인은 사용자 항목의 **memberOf** 속성을 업데이트하여 그룹 DN 값을 제거합니다.

참고

--exclude 매개변수에 설정된 값은 **--scope** 에 설정된 값보다 우선 순위가 높습니다. 두 매개변수 모두에 설정된 범위가 겹치면 MemberOf 플러그인은 겹치지 않는 디렉터리 항목에서만 작동합니다.

MemberOf 플러그인의 범위를 설정하는 방법에 대한 자세한 내용은 웹 콘솔을 사용하여 각 서버에서 MemberOf 플러그인 구성을 참조하십시오.

1.5.8. **수정** 작업을 사용하여 사용자 항목에서 **memberOf** 속성 값 업데이트

MemberOf 플러그인은 그룹 항목의 구성에 따라 그룹 멤버 항목의 **memberOf** 속성을 자동으로 관리합니다. 그러나 다음 상황에서 **수정** 작업을 실행하여 서버 플러그인이 관리하는 **memberOf** 구성과 사용자 항목에 정의된 실제 멤버십 간의 불일치를 방지해야 합니다.

- MemberOf 플러그인을 활성화하기 전에 그룹에 그룹 멤버를 추가했습니다.
- 사용자 항목에서 **memberOf** 속성을 수동으로 편집했습니다.
- **memberOf** 특성이 이미 있는 서버에 새 사용자 항목을 가져오거나 복제했습니다.

수정 작업은 로컬에서만 실행할 수 있습니다. 복제 환경에서 Directory Server는 Directory Server가 업데이트된 항목을 복제한 후 다른 서버의 항목에 대해 **memberOf** 특성을 업데이트합니다.

사전 요구 사항

- 복제 토폴로지의 모든 서버에서 MemberOf 플러그인을 활성화했습니다. 자세한 내용은 MemberOf 플러그인 활성화를 참조하십시오.

절차

- 예를 들어 **dc=example,dc=com** 항목 및 하위 항목에서 **memberOf** 값을 업데이트하려면 다음을 실행합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof fixup "dc=example,dc=com"
Attempting to add task entry...
Successfully added task entry
```

기본적으로 **수정** 작업은 **inetUser**,**inetAdmin** 또는 **nsMemberOf** 개체 클래스를 포함하는 모든 항목에서 **memberOf** 값을 업데이트합니다.

수정 작업을 다른 오브젝트 클래스가 포함된 항목에서도 사용하려면 **-f** 필터 옵션을 사용합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin memberof fixup
-f "(|(objectclass=inetuser)(objectclass=inetadmin)(objectclass=nsmemberof)
(objectclass=nsmemberof)(objectclass=inetOrgPerson))" "dc=example,dc=com"
```

이 **수정** 작업은 **inetUser**,**inetAdmin**,**nsMemberOf** 또는 **inetOrgPerson** 개체 클래스를 포함하는 모든 항목에서 **memberOf** 값을 업데이트합니다.

추가 리소스

- LDAP 검색 필터

2장. DIRECTORY SERVER에서 역할 사용

역할을 사용하여 Directory Server 항목을 그룹화할 수 있습니다. 역할은 정적 그룹 및 동적 그룹으로 작동합니다. 역할은 구현에 더 유연하므로 그룹보다 사용하기가 더 쉽습니다. 예를 들어 애플리케이션은 그룹을 선택하고 여러 그룹의 멤버 목록을 탐색하는 대신 항목 자체를 쿼리하여 항목이 속하는 역할 목록을 가져올 수 있습니다.

명령줄 또는 웹 콘솔 을 사용하여 역할을 관리할 수 있습니다.

2.1. DIRECTORY SERVER의 역할

역할은 하이브리드 그룹과 유사하게 정적 및 동적 그룹으로 작동합니다.

- 그룹을 사용하면 Directory Server에서 그룹 항목에 항목을 멤버로 추가합니다.
- 역할로 Directory Server는 역할 속성을 항목에 추가한 다음 이 속성을 사용하여 역할 항목의 멤버를 자동으로 식별합니다.

역할 멤버는 역할을 보유한 항목입니다. 역할 유형에 따라 역할의 멤버를 명시적으로 또는 동적으로 지정할 수 있습니다. Directory Server는 다음 유형의 역할을 지원합니다.

- *관리 역할*
  관리 역할에는 명시적 멤버 목록이 있습니다. 관리 역할을 사용하여 정적 그룹으로 수행하는 것과 동일한 작업을 수행할 수 있습니다.
- *필터링된 역할*
  동적 그룹으로 필터링하는 것과 유사하게 필터링된 역할을 사용하여 역할 멤버를 필터링할 수 있습니다. Directory Server는 항목에 역할에 정의된 특정 특성이 있는지 여부에 따라 필터링된 역할에 항목을 할당합니다.
- *중첩된 역할*
  중첩된 역할에는 관리 및 필터링된 역할이 포함될 수 있습니다.

역할을 생성할 때 사용자가 역할에서 자신을 추가하거나 제거할 수 있는지 확인합니다. 자세한 내용은 2.2 절. “Directory Server에서 안전하게 역할 사용” 에서 참조하십시오.

참고

역할을 평가하는 것은 서버가 클라이언트 애플리케이션에 대해 작업을 수행하기 때문에 그룹을 평가하는 것보다 리소스 집약적입니다. 역할을 사용하면 클라이언트 애플리케이션에서 **nsRole** 특성을 검색하여 역할 멤버십을 확인할 수 있습니다. **nsRole** 속성은 항목이 속하는 역할을 식별하는 계산된 속성입니다. Directory Server는 **nsRole** 속성을 저장하지 않습니다. 클라이언트 애플리케이션 관점에서 멤버십을 확인하는 방법은 균일하며 서버 측에서 수행됩니다.

계획 및 디렉터리 서비스 설계 설명서에서 [그룹과 역할 간 분리]에서 역할을 사용하기 위한 고려 사항을 찾습니다.

2.2. DIRECTORY SERVER에서 안전하게 역할 사용

새 역할을 생성할 때 사용자가 역할에서 자신을 쉽게 추가하거나 제거할 수 있는지 고려하십시오. 예를 들어 Cryostat **Biking** 관심 그룹 역할의 사용자가 쉽게 추가하거나 제거할 수 있습니다. 그러나 **마케팅** 역할이 할당된 사용자가 역할에서 자신을 추가하거나 제거하도록 허용해서는 안 됩니다.

한 가지 잠재적인 보안 위험은 역할을 활성화하여 사용자 계정을 활성화하는 것입니다. 비활성 역할에는

접미사에 대해 ACI(Special Access Control instructions)가 정의되어 있습니다. 관리자가 역할을 자유롭게 추가하고 제거할 수 있는 경우 이러한 사용자는 비활성 역할에서 자신을 제거하여 계정 잠금을 해제할 수 있습니다.

예를 들어 사용자에게 관리 역할이 할당됩니다. 계정 비활성화를 사용하여 Directory Server가 이 관리 역할을 잠그면 Directory Server에서 해당 사용자의 **nsAccountLock** 속성을 **true** 로 계산하므로 사용자가 서버에 바인딩할 수 없습니다. 그러나 사용자가 이미 Directory Server에 바인딩되어 관리 역할을 통해 잠겼으면 사용자는 해당 항목에서 **nsRoleDN** 속성을 제거하고 ACI를 제한하지 않는 경우 조직의 잠금을 해제할 수 있습니다.

사용자가 **nsRoleDN** 속성을 제거하지 못하도록 하려면 역할 유형에 따라 다음 ACI를 사용합니다.

- 관리 역할. 관리 역할의 멤버인 항목의 경우 다음 ACI를 사용합니다.

```
aci: (targetattr="nsRoleDN")
(targattrfilters= add=nsRoleDN:(!(nsRoleDN=cn=AdministratorRole,dc=example,dc=com)),
del=nsRoleDN:(!(nsRoleDN=cn=nsManagedDisabledRole,dc=example,dc=com)))
(version3.0;acl "allow mod of nsRoleDN by self but not to critical values"; allow(write)
userdn=ldap:///self;)
```

- 필터링된 역할입니다. 필터에 속하는 속성 보호(**nsRoleFilter**). 사용자가 필터링된 역할에서 사용하는 특성을 추가, 삭제 또는 수정할 수 없도록 합니다. Directory Server가 filter 속성의 값을 계산하는 경우 이 필터 특성 값을 수정할 수 있는 모든 특성을 보호해야 합니다.
- 중첩된 역할. 중첩된 역할에는 필터링된 역할과 관리 역할이 포함될 수 있습니다. 따라서 중첩된 역할에 포함된 역할의 각 속성에 대해 ACI의 수정 작업을 제한해야 합니다.

추가 리소스

- 사용자 및 역할 수동 활성화

2.3. 명령줄을 사용하여 DIRECTORY SERVER에서 역할 관리

명령줄을 사용하여 역할을 보고, 만들고, 삭제할 수 있습니다.

2.3.1. Directory Server에서 관리되는 역할 생성

관리되는 역할은 명시적 열거된 멤버 목록이 있는 역할입니다. **ldapmodify** 유틸리티를 사용하여 관리 역할을 생성할 수 있습니다. 다음 예제에서는 마케팅 팀에 대한 관리 역할을 생성합니다.

사전 요구 사항

- **ou=people,dc=example,dc=com** 상위 항목이 Directory Server에 있습니다.
- **cn=Bob Cryostat,ou=people,dc=example,dc=com** 사용자 항목이 Directory Server에 있습니다.

절차

1. **ldapmodify** 명령을 **-a** 옵션과 함께 사용하여 **cn=Marketing** managed 역할 항목을 생성합니다.

```
# ldapmodify -a -D "cn=Directory Manager" -W -H ldap://server.example.com -x << EOF
dn: cn=Marketing,ou=people,dc=example,dc=com
objectclass: top
```

```
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsSimpleRoleDefinition
objectclass: nsManagedRoleDefinition
cn: Marketing
description: managed role for the marketing team
EOF
```

관리 역할 항목에는 다음 오브젝트 클래스가 포함되어야 합니다.

- **LDAPsubentry**
- **nsRoleDefinition**
- **nsSimpleRoleDefinition**
- **nsManagedRoleDefinition**

2. 이 사용자 항목에 **nsRoleDN** 특성을 추가하여 **cn=Marketing,ou=bob Cryostat,dc=example,dc=com 사용자 항목을 cn=Bob Cryostat,ou=people,dc=** example,dc=com 사용자 항목에 할당합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x << EOF
dn: cn=Bob Jones,ou=people,dc=example,dc=com
changetype: modify
add: nsRoleDN
nsRoleDN: cn=Marketing,ou=people,dc=example,dc=com
EOF

modifying entry "cn=Bob Jones,ou=people,dc=example,dc=com"
```

3. 선택 사항: **userRoot** 데이터베이스에서 **nsRoleDN** 속성에 대한 *같음* 인덱스를 구성하여 인덱싱되지 않은 검색을 방지합니다.

```
#  dsconf -D "cn=Directory Manager" ldap://server.example.com backend index add --index-type eq --attr nsroleDN --reindex userRoot
```

검증

- **cn=Marketing,ou=people,dc=example,dc=com** 관리 역할에 속하는 사용자 항목을 나열합니다.

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -x -b "dc=example,dc=com" "(nsRole=cn=Marketing,ou=people,dc=example,dc=com)" dn

dn: cn=Bob Jones,ou=people,dc=example,dc=com

dn: cn=Tom Devis,ou=people,dc=example,dc=com
```

추가 리소스

- LDAP 브라우저에서 역할 생성
- ldapadd, ldapmodify 및 ldapdelete 유틸리티에 입력 제공

- 명령줄을 사용하여 특정 데이터베이스의 인덱스 유지
- **ldapmodify(1)** 매뉴얼 페이지

2.3.2. Directory Server에서 필터링된 역할 생성

항목에 특정 특성이 정의된 경우 Directory Server는 필터링된 역할에 항목을 할당합니다. 역할 정의는 **nsRoleFilter** LDAP 필터를 지정합니다. 필터와 일치하는 항목은 역할의 멤버입니다.

ldapmodify 유틸리티를 사용하여 필터링된 역할을 생성할 수 있습니다. 다음 예제에서는 영업 부서 관리자에 대해 필터링된 역할을 생성합니다.

사전 요구 사항

- **ou=people,dc=example,dc=com** 상위 항목이 Directory Server에 있습니다.

절차

1. **ldapmodify** 명령을 **-a** 옵션과 함께 사용하여 **cn=SalesManagerFilter** 필터링된 역할 항목을 생성합니다.

```
# ldapmodify -a -D "cn=Directory Manager" -W -H ldap://server.example.com -x << EOF
dn: cn=SalesManagerFilter,ou=people,dc=example,dc=com
changetype: add
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: SalesManagerFilter
nsRoleFilter: o=sales managers
Description: filtered role for sales managers
EOF
```

cn=SalesManagerFilter 필터링된 역할 항목에는 **o=sales 관리자가** 역할에 대해 필터링합니다. **영업 관리자** 값이 있는 **o** 속성이 있는 모든 사용자 항목은 필터링된 역할의 멤버입니다.

필터링된 역할의 멤버인 사용자 항목의 예:

```
dn: cn=Pat Smith,ou=people,dc=example,dc=com
objectclass: person
cn: Pat
sn: Smith
userPassword: password
o: sales managers
```

필터링된 역할 항목에는 다음과 같은 오브젝트 클래스가 있어야 합니다.

- **LDAPsubentry**
- **nsRoleDefinition**
- **nsComplexRoleDefinition**
- **nsFilteredRoleDefinition**

2. 선택 사항: 인덱싱되지 않은 검색을 방지하기 위해 **nsRoleFilter** 역할 필터에서 사용하는 속성에 대한 *같음* 인덱스를 구성합니다. 지정된 예에서 역할은 **o=sales 관리자를** 필터로 사용합니다. 따라서 검색 성능을 개선하기 위해 **o** 특성을 인덱싱합니다.

```
#  dsconf -D "cn=Directory Manager" ldap://server.example.com backend index add --
index-type eq --attr o --reindex userRoot
```

검증

- **cn=SalesManagerFilter,ou=people,dc=example,dc=com** 필터링된 역할에 속하는 사용자 항목을 나열합니다.

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -x -b
"dc=example,dc=com" "
(nsRole=cn=SalesManagerFilter,ou=people,dc=example,dc=com)" dn

dn: cn=Jess Mor,ou=people,dc=example,dc=com

dn: cn=Pat Smith,ou=people,dc=example,dc=com
```

추가 리소스

- LDAP 브라우저에서 역할 생성
- ldapadd, ldapmodify 및 ldapdelete 유틸리티에 입력 제공
- **ldapmodify(1)** 매뉴얼 페이지

2.3.3. Directory Server에서 중첩된 역할 생성

중첩된 역할에는 관리 및 필터링된 역할이 포함될 수 있습니다. 중첩된 역할 항목에는 중첩할 역할을 식별하는 **nsRoleDN** 속성이 필요합니다.

ldapmodify 유틸리티를 사용하여 중첩된 역할을 생성할 수 있습니다. 다음 예제에서는 Directory Server에서 관리되는 역할 만들기 및 Directory Server에서 필터링된 역할 만들기에서 만든 관리 및 필터링된 역할이 포함된 중첩 역할을 만듭니다.

사전 요구 사항

- **ou=people,dc=example,dc=com** 상위 항목이 Directory Server에 있습니다.

절차

1. **ldapmodify** 명령을 **-a** 옵션과 함께 사용하여 **cn=SalesManagerFilter** 필터링된 역할과 **cn=Marketing** managed 역할을 포함하는 **cn=MarketingSales** 중첩 역할 항목을 생성합니다.

```
# ldapmodify -a -D "cn=Directory Manager" -W -H ldap://server.example.com -x << EOF
dn: cn=MarketingSales,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsNestedRoleDefinition
cn: MarketingSales
```

```
nsRoleDN: cn=SalesManagerFilter,ou=people,dc=example,dc=com
nsRoleDN: cn=Marketing,ou=people,dc=example,dc=com
EOF
```

선택적으로 역할에 **description** 속성이 있을 수 있습니다.

중첩된 역할 항목에는 다음과 같은 오브젝트 클래스가 있어야 합니다.

- **LDAPsubentry**
- **nsRoleDefinition**
- **nsComplexRoleDefinition**
- **nsNestedRoleDefinition**

검증

- **cn=MarketingSales** 중첩 역할에 속하는 사용자 항목을 나열합니다.

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -x -b
"dc=example,dc=com" "(nsRole=cn=MarketingSales,ou=people,dc=example,dc=com)"
dn

dn: cn=Bob Jones,ou=people,dc=example,dc=com
dn: cn=Pat Smith,ou=people,dc=example,dc=com
dn: cn=Jess Mor,ou=people,dc=example,dc=com
dn: cn=Tom Devis,ou=people,dc=example,dc=com
```

추가 리소스

- LDAP 브라우저에서 역할 생성
- ldapadd, ldapmodify 및 ldapdelete 유틸리티에 입력 제공
- **ldapmodify(1)** 매뉴얼 페이지

2.3.4. 항목의 역할 보기

항목의 역할을 보려면 명시적으로 지정된 **nsRole** 가상 속성과 함께 **ldapsearch** 명령을 사용하십시오.

사전 요구 사항

- 역할 항목이 있습니다.
- **uid=user_name** 사용자 항목에 역할이 할당되었습니다.

절차

- **nsRole** 가상 속성이 지정된 **uid=*user_name*** 항목을 검색합니다.

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"dc=example,dc=com" -s sub -x "(uid=user_name)" nsRole

dn: uid=user_name,ou=people,dc=example,dc=com
```

```
...
nsRole: cn=Role for Managers,dc=example,dc=com
nsRole: cn=Role for Accounting,dc=example,dc=com
```

명령은 **uid=*user_name*** 사용자가 멤버인 모든 역할을 검색합니다.

2.3.5. Directory Server에서 역할 삭제

Directory Server에서 역할을 삭제하려면 **ldapmodify** 명령을 사용할 수 있습니다.

다음은 Directory Server에서 **cn=Marketing** 관리 역할을 삭제하는 예입니다.

절차

- **cn=Marketing** 관리 역할 항목을 삭제하려면 다음을 입력합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -x << EOF
dn: cn=Marketing,ou=People,dc=example,dc=com
changetype: delete
EOF

deleting entry "cn=Marketing,ou=People,dc=example,dc=com"
```

참고

역할을 삭제하면 Directory Server는 역할 항목만 삭제하고 각 역할 멤버에 대한 **nsRoleDN** 속성을 삭제하지 않습니다. 각 역할 멤버에 대한 **nsRoleDN** 속성을 삭제하려면 Referential Integrity 플러그인을 활성화하고 **nsRoleDN** 특성을 관리하도록 이 플러그인을 구성합니다.

참조 무결성 플러그인에 대한 자세한 내용은 참조 무결성 사용을 참조하십시오. 항목 간의 관계를 유지 관리합니다.

추가 리소스

- LDAP 브라우저에서 역할 삭제
- **ldapmodify(1)** 매뉴얼 페이지

2.4. 웹 콘솔을 사용하여 DIRECTORY SERVER에서 역할 관리

웹 콘솔에서 **LDAP 브라우저를** 사용하여 역할을 보고, 생성하고, 삭제할 수 있습니다.

2.4.1. LDAP 브라우저에서 역할 생성

웹 콘솔에서 **LDAP 브라우저** 마법사를 사용하여 Red Hat Directory Server 항목에 대한 역할을 생성할 수 있습니다.

사전 요구 사항

- 웹 콘솔에 액세스합니다.
- 상위 항목은 Directory Server에 있습니다.

절차

1. 웹 콘솔에 로그인하고 **Red Hat Directory Server** 를 클릭합니다.
2. 웹 콘솔에서 **Red Hat Directory Server** 인터페이스를 로드한 후 **LDAP 브라우저를 엽니다**.
3. LDAP 항목을 선택하고 **옵션** 메뉴를 엽니다.
4. 드롭다운 메뉴에서 **새로** 만들기를 선택하고 **새 역할 만들기** 를 클릭합니다.
5. 마법사의 단계에 따라 각 단계를 완료한 후 **다음** 버튼을 클릭합니다.
6. 역할을 생성하려면 **Create Role** (역할 만들기) 단계에서 역할 설정을 검토하고 Create(**생성**) 버튼을 클릭합니다. **뒤로** 버튼을 클릭하여 역할 설정을 수정하거나 **취소** 버튼을 클릭하여 역할 생성을 취소할 수 있습니다.
7. 마법사 창을 닫으려면 **완료** 버튼을 클릭합니다.

검증

- LDAP 항목을 확장하고 항목 매개 변수 사이에 새 역할이 표시되는지 확인합니다.

2.4.2. LDAP 브라우저에서 역할 삭제

웹 콘솔에서 **LDAP** 브라우저를 사용하여 Red Hat Directory Server 항목에서 역할을 삭제할 수 있습니다.

사전 요구 사항

- 웹 콘솔에 액세스합니다.
- 상위 항목은 Directory Server에 있습니다.

절차

1. 웹 콘솔에 로그인하고 **Red Hat Directory Server** 를 클릭합니다.
2. 웹 콘솔에서 **Red Hat Directory Server** 인터페이스를 로드한 후 **LDAP 브라우저를** 클릭합니다.
3. LDAP 항목을 확장하면 삭제할 역할을 선택합니다.
4. **옵션** 메뉴를 열고 **삭제** 를 선택합니다.
5. 삭제할 역할에 대한 데이터를 확인하고 **삭제** 단계에 도달할 때까지 **다음** 버튼을 클릭합니다.
6. 스위치를 **Yes로 전환합니다. 위치를 확인하고 삭제** 버튼을 클릭합니다.
7. 마법사 창을 닫으려면 **완료** 버튼을 클릭합니다.

검증

- LDAP 항목을 확장하고 역할이 더 이상 항목 매개변수의 일부가 아닌지 확인합니다.

2.4.3. LDAP 브라우저에서 역할 수정

웹 콘솔에서 **LDAP** 브라우저를 사용하여 Red Hat Directory Server 항목의 역할 매개변수를 수정할 수 있습니다.

사전 요구 사항

- 웹 콘솔에 액세스합니다.
- Red Hat Directory Server에 상위 항목이 있습니다.

절차

1. 웹 콘솔에 로그인하고 **Red Hat Directory Server** 를 클릭합니다.
2. 웹 콘솔에서 **Red Hat Directory Server** 인터페이스를 로드한 후 **LDAP 브라우저를 클릭합니다**.
3. LDAP 항목을 확장하고 수정할 역할을 선택합니다.
4. **옵션** 메뉴를 클릭하고 **편집을** 선택하여 역할의 매개 변수를 수정하거나 **Rename** 을 선택하여 역할의 이름을 바꿉니다.
5. 마법사 창에서 필요한 매개변수를 수정하고 **LDIF 문** 단계를 확인할 때까지 각 단계 후에 **다음을** 클릭합니다.
6. 업데이트된 매개 변수를 확인하고 **항목 이름 수정 또는 입력 이름 변경을** 클릭합니다.
7. 마법사 창을 닫으려면 **완료** 버튼을 클릭합니다.

검증

- LDAP 항목을 확장하고 업데이트된 매개변수가 역할에 나열되어 있는지 확인합니다.

3장. DIRECTORY MANAGER 암호 변경

Directory Manager는 Linux 운영 체제의 **root** 사용자와 비교할 수 있는 권한 있는 데이터베이스 관리자입니다. Directory Manager 항목 및 해당 암호는 인스턴스 설치 중에 설정됩니다. 관리자는 Directory Manager 암호를 변경하여 다른 암호를 사용할 수 있습니다.

3.1. 명령줄을 사용하여 DIRECTORY MANAGER 암호 변경

dsconf 명령줄 유틸리티를 사용하여 Directory Manager의 새 암호를 설정하거나 **nsslapd-rootpw** 매개변수를 설정하여 수동으로 설정할 수 있습니다.

중요

암호화된 연결만 사용하여 암호를 설정합니다. 암호화되지 않은 연결을 사용하면 암호가 네트워크에 노출될 수 있습니다. 서버가 암호화된 연결을 지원하지 않는 경우 웹 콘솔을 사용하여 Directory Manager 암호를 업데이트합니다.

절차

- 다음 옵션 중 하나를 사용하여 Directory Manager 암호를 설정합니다.
  - 암호를 자동으로 암호화하려면 다음을 수행합니다.

```
# dsconf -D "cn=Directory Manager" ldaps://server.example.com config replace
nsslapd-rootpw=password
```

Directory Server는 **nsslapd-rootpw** 매개변수에 설정한 일반 텍스트 값을 자동으로 암호화합니다.

주의

암호에는 중괄호 **{}** 을 사용하지 마십시오. Directory Server는 **{password-storage-scheme}hashed_password** 형식으로 암호를 저장합니다. 서버는 중괄호의 문자를 암호 스토리지 체계로 해석합니다. 문자열이 잘못된 스토리지 체계이거나 암호가 올바르게 해시되지 않으면 Directory Manager가 서버에 연결할 수 없습니다.

  - 암호를 수동으로 암호화하려면 다음을 수행합니다.
    1. 새 암호 해시를 생성합니다. 예를 들면 다음과 같습니다.

```
# pwdhash -D /etc/dirsrv/slapd-instance_name password
{PBKDF2_SHA256}AAAgAMwPYIhEkQozTagoX6RGG5E7d6/6oOJ8TVty...
```

암호는 Directory Server 인스턴스 구성의 **nsslapd-rootpwstoragescheme** 속성에 설정된 암호 스토리지 스키마를 사용하여 암호화됩니다.

    2. STARTTLS 연결을 사용하여 **nsslapd-rootpw** 특성을 이전 단계에 표시된 값으로 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldaps://server.example.com config replace
nsslapd-
rootpw="{PBKDF2_SHA256}AAAgAMwPYlhEkQozTagoX6RGG5E7d6/6oOJ8TV
ty..."
```

추가 리소스

- 웹 콘솔을 사용하여 Directory Manager 암호 변경

3.2. 웹 콘솔을 사용하여 DIRECTORY MANAGER 암호 변경

웹 콘솔을 사용하여 Directory Manager의 새 암호를 설정할 수 있습니다.

사전 요구 사항

- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. **Server → Server Settings → Directory Manager** 메뉴를 엽니다.
2. **Directory Manager Password** 및 **Confirm Password** 필드에 새 암호를 입력합니다.
3. 선택 사항: 다른 암호 스토리지 스키마를 설정합니다.
4. **저장**을 클릭합니다.

4장. DIRECTORY MANAGER 암호 재설정

Directory Manager는 Linux 운영 체제의 **root** 사용자와 비교할 수 있는 권한 있는 데이터베이스 관리자입니다. 인스턴스 설치 중에 Directory Manager 암호가 설정됩니다. 암호가 손실되면 디렉터리에 대한 권한 있는 액세스 권한으로 재설정할 수 있습니다.

4.1. 명령줄을 사용하여 DIRECTORY MANAGER 암호 재설정

Directory Server 인스턴스에 대한 root 액세스 권한이 있는 경우 Directory Manager의 암호를 재설정할 수 있습니다.

절차

1. 새 암호 해시를 생성합니다. 예를 들면 다음과 같습니다.

```
# pwdhash -D /etc/dirsrv/slapd-instance_name new_password
{PBKDF2_SHA256}AAAgABU0bKhyjY53NcxY33ueoPjOUWtl4iyYN5uW...
```

   Directory Server 인스턴스 구성 경로를 지정했기 때문에 **pwdhash** 생성기는 **nsslapd-rootpwstoragescheme** 속성에 설정된 암호 스토리지 스키마를 자동으로 사용하여 새 암호를 암호화합니다.

2. Directory Server 인스턴스를 중지합니다.

```
# dsctl instance_name stop
```

3. **/etc/dirsrv/slapd-*instance_name*/dse.ldif** 파일을 편집하고 **nsslapd-rootpw** 속성을 첫 번째 단계에 표시된 값으로 설정합니다.

```
nsslapd-rootpw: {PBKDF2_SHA256}AAAgABU0bKhyjY53NcxY33ueoPjOUWtl4iyYN5uW...
```

4. Directory Server 인스턴스를 시작합니다.

```
# dsctl instance_name start
```

5장. 암호 정책 구성

암호 정책은 특정 수준의 보안을 적용하여 암호 사용과 관련된 위험을 최소화합니다. 예를 들어 다음과 같이 암호 정책을 정의할 수 있습니다.

- 사용자는 일정에 따라 암호를 변경해야 합니다.
- 사용자는 중요하지 않은 암호를 제공해야 합니다.
- 암호 구문은 특정 복잡성 요구 사항을 충족해야 합니다.

5.1. 암호 정책의 작동 방식

Directory Server는 일반적인 피라미드에서 작동하는 세분화된 암호 정책을 지원합니다. 글로벌 암호 정책은 사용자 수준 암호 정책으로 대체되는 하위 트리 수준 암호 정책으로 대체됩니다.

다음을 정의할 수 있습니다.

- 전체 디렉터리에 적용되는 글로벌 암호 정책
- 로컬 암호 정책
  - 특정 하위 트리에 적용되는 하위 트리 수준 정책
  - 특정 사용자에게 적용되는 사용자 수준 정책

암호 정책은 추가되지 않습니다. 하나의 암호 정책만 항목에 적용됩니다. 예를 들어 사용자 수준 암호 정책에는 특정 특성을 글로벌 또는 하위 트리 수준 암호 정책에 구성할 때 이 속성은 사용자에게 적용되지 않습니다. 이 경우 사용자가 로그인을 시도하면 사용자 수준 정책만 활성화됩니다.

주의

암호 관리자 계정 또는 디렉터리 관리자(root DN)를 사용하여 암호를 설정하는 경우 암호 정책을 바이패스합니다. 일반 사용자 암호 관리에 이러한 계정을 사용하지 마십시오. 암호 정책을 우회하거나 재설정 후 임시 암호를 설정하기 위한 현재 암호 제약 조건을 의도적으로 재정의하는 등 암호 정책을 우회해야 하는 암호 관리 작업을 수행하는 경우에만 사용합니다.

사용자 계정에 적용되는 전체 암호 정책은 다음 요소로 구성됩니다.

- **암호 정책의 유형 또는 수준을 확인합니다.** 이 정보는 서버가 글로벌 암호 정책 또는 로컬 암호 정책을 확인하고 적용해야 하는지 여부를 나타냅니다.
- **암호 추가 및 수정** 암호 정보에는 암호 구문 및 암호 기록 정보가 포함됩니다.
- **정보를 바인딩합니다.** 바인딩 정보에는 허용된 유예 로그인 수, 암호 사용 기간 속성 및 바인딩 실패 추적이 포함됩니다.

참고

암호 정책을 설정한 후에는 계정 잠금 정책을 구성하여 잠재적인 위협으로부터 사용자 암호를 보호할 수 있습니다. 계정 잠금은 사용자 암호를 반복적으로 추측하여 디렉토리에 침입하려는 시도로부터 보호됩니다.

추가 리소스

- 암호 기반 계정 잠금 정책 구성
- 시간 기반 계정 잠금 정책 구성

5.2. 명령줄을 사용하여 글로벌 암호 정책 구성

기본적으로 글로벌 암호 정책 설정은 비활성화되어 있습니다. **dsconf** 명령줄 유틸리티를 사용하여 글로벌 암호 정책을 구성할 수 있습니다.

절차

1. 현재 설정을 표시합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy get
Global Password Policy: cn=config
---------------------------------
passwordstoragescheme: PBKDF2_SHA256
passwordChange: on
passwordMustChange: off
passwordHistory: off
passwordInHistory: 6
...
```

2. 암호 정책 설정을 조정합니다. 사용 가능한 설정의 전체 목록은 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --help
```

예를 들어 암호 구문 검사를 활성화하고 최소 암호 길이를 **12** 자로 설정하려면 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwdchecksyntax on --pwdmintokenlen 12
```

3. 암호 정책의 계정 잠금 기능을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwdlockout on
```

5.3. 웹 콘솔을 사용하여 글로벌 암호 정책 구성

기본적으로 글로벌 암호 정책 설정은 비활성화되어 있습니다. 웹 콘솔을 사용하여 글로벌 암호 정책을 구성할 수 있습니다.

사전 요구 사항

- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. **Database → Password Policies → Global Policy** 메뉴를 엽니다.
2. 글로벌 암호 정책 설정을 설정합니다. 다음 카테고리에서 매개변수를 설정할 수 있습니다.
   - 암호 스토리지 스키마와 같은 일반 설정
   - 암호가 만료되는 시간과 같은 암호 만료 설정
   - 계정 잠금을 해제해야 하는 로그인 실패 발생 후와 같이 계정 잠금 설정
   - 최소 암호 길이와 같은 암호 구문 설정
     매개변수에 대한 **cn=config** 항목에 툴 팁과 해당 속성 이름을 표시하려면 설정 위에 마우스 커서를 놓습니다.
3. **저장**을 클릭합니다.

5.4. 로컬 암호 정책 항목

dsconf localpwp addsubtree 또는 **dsconf localpwp adduser** 명령을 사용하면 Directory Server에서 로컬 암호 정책 속성을 저장할 항목을 자동으로 생성합니다.

하위 트리의 경우 다음 항목이 추가됩니다.

표 5.1. 하위 트리의 로컬 암호 정책 항목

항목 이름	설명	내용
nsPwPolicyContainer	하위 트리 수준의 컨테이너 항목	하위 트리 및 모든 하위 항목에 대한 다양한 암호 정책 관련 항목
nsPwPolicyEntry	실제 암호 정책 사양 항목	하위 트리와 관련된 모든 암호 정책 속성
nsPwTemplateEntry	CoS 템플릿 항목	**nsPwPolicyEntry** 항목을 가리키는 **pwdpolicysubentry** 값
<cos 정의 항목 DN>	하위 트리 수준의 CoS 정의 항목	cos 정의 항목

예 5.1. 하위 트리 ou=people,dc=example,dc=com의 nsPwPolicyContainer 항목

```
dn: cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectClass: top
objectClass: nsContainer
cn: nsPwPolicyContainer
```

예 5.2. 하위 트리 ou=people,dc=example,dc=com의 nsPwPolicyEntry 항목

```
dn: cn="cn=nsPwPolicyEntry,ou=people,dc=example,dc=com",
 cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: ldapsubentry
objectclass: passwordpolicy
```

예 5.3. 하위 트리 ou=people,dc=example,dc=com의 nsPwTemplateEntry 항목

```
dn: cn="cn=nsPwTemplateEntry,ou=people,dc=example,dc=com",
 cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: costemplate
objectclass: ldapsubentry
cosPriority: 1
pwdpolicysubentry: cn="cn=nsPwPolicyEntry,ou=people,dc=example,dc=com",
    cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
```

예 5.4. 하위 트리 ou=people,dc=example,dc=com의 CoS 사양 항목

```
dn: cn=newpwdpolicy_cos,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: cosSuperDefinition
objectclass: cosPointerDefinition
cosTemplateDn: cn=cn=nsPwTemplateEntry\,ou=people\,dc=example,dc=com,
 cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
cosAttribute: pwdpolicysubentry default operational
```

사용자의 경우 다음 항목이 추가됩니다.

표 5.2. 사용자의 로컬 암호 정책 항목

항목 이름	설명	내용
nsPwPolicyContainer	상위 수준의 컨테이너 항목	사용자와 모든 하위 항목에 대한 다양한 암호 정책 관련 항목
nsPwPolicyEntry	실제 암호 정책 사양 항목	사용자와 관련된 모든 암호 정책 속성

예 5.5. 사용자 uid=user_name,ou=people,dc=example,dc=com의 nsPwPolicyContainer 항목

```
dn: cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectClass: top
objectClass: nsContainer
cn: nsPwPolicyContainer
```

예 5.6. 사용자 uid=user_name,ou=people,dc=example,dc=com의 nsPwPolicyEntry 항목

```
dn: cn="cn=nsPwPolicyEntry,uid=user_name,ou=people,dc=example,dc=com",
 cn=nsPwPolicyContainer,ou=people,dc=example,dc=com
objectclass: top
objectclass: extensibleObject
objectclass: ldapsubentry
objectclass: passwordpolicy
```

5.5. 명령줄을 사용하여 로컬 암호 정책 구성

전체 디렉터리에 대한 설정을 정의하는 글로벌 암호 정책과 달리 로컬 암호 정책은 특정 사용자 또는 하위 트리에 대한 정책입니다. 현재는 명령줄을 사용하여 로컬 암호 정책만 설정할 수 있습니다.

사전 요구 사항

- 정책을 생성할 사용자 또는 하위 트리 항목이 이미 디렉터리에 있습니다.

절차

1. 하위 트리 또는 사용자 항목에 대한 로컬 암호 정책이 이미 있는지 확인합니다. 예를 들면 다음과 같습니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp get
"ou=People,dc=example,dc=com"
Enter password for cn=Directory Manager on ldap://server.example.com:
Error: No password policy was found for this entry
```

로컬 정책이 없는 경우 다음을 생성합니다.

- **하위 트리** 암호 정책을 생성하려면 다음을 수행합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp
addsubtree "ou=People,dc=example,dc=com"
```

- **사용자** 암호 정책을 생성하려면 다음을 수행합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp adduser
"uid=user_name,ou=People,dc=example,dc=com"
```

2. 로컬 정책 특성을 설정합니다. 사용 가능한 설정의 전체 목록은 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp set --help
```

예를 들어 암호 만료를 활성화하고 최대 암호 기간을 14일(**1209600** 초)으로 설정하려면 다음을 수행합니다.

- 하위 **트리** 암호 정책에서 다음을 수행합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp set --
pwdexpire on --pwdmaxage 1209600 "ou=People,dc=example,dc=com"
```

- **사용자** 암호 정책에서 다음을 수행합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp set --
pwdexpire on --pwdmaxage 1209600
"uid=user_name,ou=People,dc=example,dc=com"
```

5.6. 명령줄을 사용하여 로컬 암호 정책 비활성화

새 로컬 정책을 생성할 때 **cn=config** 항목의 **nsslapd-pwpolicy-local** 매개변수가 자동으로 **on** 로 설정됩니다. 로컬 암호 정책을 활성화하지 않아야 하는 경우 명령줄을 사용하여 수동으로 비활성화할 수 있습니다.

절차

- 모든 로컬 정책을 비활성화하거나 특정 로컬 정책을 제거합니다.
  - 모든 로컬 암호 정책을 비활성화하려면 다음을 수행합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwdlocal off
```

  - 기존의 단일 **하위 트리** 암호 정책을 제거하려면 다음을 수행합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp remove
"ou=People,dc=example,dc=com"
```

  - 기존 단일 **사용자** 암호 정책을 제거하려면 다음을 수행합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp remove
"uid=user_name,ou=People,dc=example,dc=com"
```

6장. 임시 암호 규칙 구성

Directory Server 암호 정책은 사용자 계정에서 임시 암호 설정을 지원합니다. 사용자에게 임시 암호를 할당하는 경우 Directory Server는 사용자가 암호를 변경할 때까지 암호 변경 이외의 다른 작업을 거부합니다.

임시 암호의 기능은 다음과 같습니다.

- **cn=Directory Manager** 계정만 임시 암호를 할당할 수 있습니다.
- Directory Server는 공격자가 암호를 조사하지 않도록 고정 횟수만 인증 시도를 허용합니다.
- Directory Server를 사용하면 지정된 지연 후 인증 시도에서 임시 암호를 설정한 후 직접 사용할 수 없도록 구성할 수 있습니다.
- Directory Server에서는 사용자가 사용하거나 재설정하지 않는 경우 임시 암호가 만료되도록 지정된 시간 동안만 인증 시도를 허용합니다.
- 인증이 성공한 경우 Directory Server에서는 사용자가 다른 작업을 수행하기 전에 암호를 재설정해야 합니다.

기본적으로 임시 암호 규칙은 비활성화되어 있습니다. 글로벌 또는 로컬 암호 정책으로 구성할 수 있습니다.

6.1. 글로벌 암호 정책에서 임시 암호 규칙 활성화

전체 Directory Server 인스턴스에 대한 임시 암호 기능을 활성화하려면 다음을 수행합니다.

1. 관리자가 재설정한 경우 해당 사용자를 활성화하여 암호를 변경해야 합니다.
2. 글로벌 암호 정책에서 기능을 구성합니다.

관리자가 사용자의 **userPassword** 속성을 업데이트하고 **passwordMustChange** 속성을 의 로 설정하면 Directory Server **에서** 임시 암호 규칙을 적용합니다.

절차

1. 관리자가 암호를 재설정한 후 사용자가 암호를 변경하도록 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com  pwpolicy set --
pwdmustchange on
```

2. 글로벌 암호 정책에서 임시 암호 규칙 설정을 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwptprmaxuse 5 --pwptprdelayexpireat 3600 --pwptprdelayvalidfrom 60
```

이 예제에서는 다음을 수행합니다.

- **--pwptprmaxuse** 옵션은 사용자가 임시 암호를 **5** 로 사용할 수 있는 최대 시도 수를 설정합니다.
- **--pwptprdelayexpireat** 옵션은 임시 암호가 **3600** 초(1시간)로 만료되기 전의 시간을 설정합니다.

- **--pwptprdelayvalidfrom** 옵션은 관리자가 사용자 암호를 재설정한 후 **--pwptprdelayexpireat** 에 설정된 시간이 **60** 초 후에 시작되도록 구성합니다.

검증

- 임시 암호 규칙을 저장하는 속성을 표시합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy get | grep -i TPR
passwordTPRMaxUse: 5
passwordTPRDelayExpireAt: 3600
passwordTPRDelayValidFrom: 60
```

6.2. 로컬 암호 정책에서 임시 암호 규칙 활성화

특정 사용자 또는 하위 트리에 대한 임시 암호 기능을 활성화하려면 관리자가 암호를 재설정한 경우 해당 암호를 변경하고 로컬 암호 정책에 기능을 구성해야 합니다.

관리자가 사용자의 **userPassword** 속성을 업데이트하고 **passwordMustChange** 속성을 **on** 에서 로 설정하면 사용자가 다음과 같은 경우 디렉터리 서버가 임시 암호 규칙을 적용합니다.

- 로컬 암호 정책이 활성화되어 있음
- 로컬 암호 정책이 활성화된 하위 트리에 저장됩니다.

절차

1. 관리자가 암호를 재설정한 후 사용자가 암호를 변경하도록 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com  pwpolicy set --pwdmustchange on
```

2. 임시 암호 규칙 설정을 구성합니다.
   - 기존 하위 트리의 경우:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp addsubtree --pwptprmaxuse 5 --pwptprdelayexpireat 3600 --pwptprdelayvalidfrom 60 ou=People,dc=example,dc=com
```

   - 기존 사용자의 경우:

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp adduser --pwptprmaxuse 5 --pwptprdelayexpireat 3600 --pwptprdelayvalidfrom 60 uid=example,ou=People,dc=example,dc=com
```

   다음 예제에서는 다음을 수행합니다.

   - **--pwptprmaxuse** 옵션은 사용자가 임시 암호를 **5** 로 사용할 수 있는 최대 시도 수를 설정합니다.
   - **--pwptprdelayexpireat** 옵션은 임시 암호가 **3600** 초(1시간)로 만료되기 전의 시간을 설정합니다.

- **--pwptprdelayvalidfrom** 옵션은 관리자가 사용자 암호를 재설정한 후 **--pwptprdelayexpireat** 에 설정된 시간이 **60** 초 후에 시작되도록 구성합니다.

검증

- 고유 이름(DN)의 로컬 암호 정책을 표시합니다.

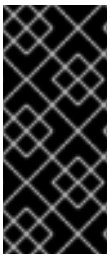
```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp get <DN> |
grep -i TPR
passwordTPRMaxUse: 5
passwordTPRDelayExpireAt: 3600
passwordTPRDelayValidFrom: 60
```

7장. 암호 관리자 권한 할당

디렉터리 관리자는 사용자 또는 사용자 그룹에 *암호 관리자* 역할을 할당할 수 있습니다. 암호 관리자는 적절한 권한으로ACI(액세스 제어 지침)가 필요하므로 단일 ACI가 모든 암호 관리자를 관리할 수 있도록 그룹을 구성하는 것이 좋습니다.

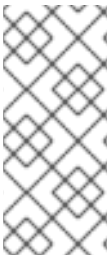
암호 관리자 역할을 사용하는 것은 다음 시나리오에서 유용합니다.

- 다음 로그인 시 사용자가 비밀번호를 변경하도록 강제하는 구독 설정
- 사용자 암호를 암호 정책에 정의된 다른 스토리지 체계로 변경



중요

암호 관리자는 모든 사용자 암호 작업을 수행할 수 있습니다. 암호 관리자 계정 또는 디렉터리 관리자(root DN)를 사용하여 암호를 설정하는 경우 암호 정책을 바이패스하고 확인하지 않습니다. 일반 사용자 암호 관리에 이러한 계정을 사용하지 마십시오. Red Hat은 **userPassword** 속성만 업데이트할 수 있는 권한이 있는 데이터베이스의 기존 역할에서 일반 암호 업데이트를 수행할 것을 권장합니다.



참고

cn=config 항목 아래에 새 **passwordAdminSkipInfoUpdate: on/off** 설정을 추가하여 암호 관리자가 수행하는 암호 업데이트를 세밀하게 제어할 수 있습니다. 이 설정을 활성화하면 암호 업데이트는 특정 속성(예: **passwordHistory**,**passwordExpirationTime**,**passwordRetryCount**,**pwdReset**, **passwordExpWarned**)을 업데이트하지 않습니다.

7.1. 글로벌 정책에서 암호 관리자 권한 할당

글로벌 정책에서는 사용자 또는 사용자 그룹에 암호 관리자 역할을 할당할 수 있습니다. Red Hat은 모든 암호 관리자를 관리하도록 하나의ACI(액세스 제어 명령)를 설정할 수 있도록 그룹을 구성하는 것이 좋습니다.

사전 요구 사항

- 암호 관리자 역할을 할당하려는 모든 사용자를 포함하는 **password_admins** 그룹을 생성했습니다.

절차

1. 암호 관리자 역할에 대한 권한을 정의하는 ACI를 생성합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x << EOF
dn: ou=people,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="userPassword || nsAccountLock || userCertificate || nsSshPublicKey")
(targetfilter="(objectClass=nsAccount)")(version 3.0; acl "Enable user password reset"; allow
(write, read)(groupdn="ldap:///cn=password_admins,ou=groups,dc=example,dc=com");)
EOF
```

2. 암호 관리자 역할을 그룹에 할당합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwdadmin "cn=password_admins,ou=groups,dc=example,dc=com"
```

7.2. 로컬 정책에서 암호 관리자 권한 할당

로컬 정책에서는 사용자 또는 사용자 그룹에 암호 관리자 역할을 할당할 수 있습니다. Red Hat은 모든 암호 관리자를 관리하도록 하나의ACI(액세스 제어 명령)를 설정할 수 있도록 그룹을 구성하는 것이 좋습니다.

사전 요구 사항

- 암호 관리자 역할을 할당하려는 모든 사용자를 포함하는 **password_admins** 그룹을 생성했습니다.

절차

1. 암호 관리자 역할에 대한 권한을 정의하는 ACI를 생성합니다.

```
# ldapmodify -D "cn=Directory Manager" -W -p 389 -h server.example.com -x << EOF
dn: ou=people,dc=example,dc=com
changetype: modify
add: aci
aci: (targetattr="userPassword || nsAccountLock || userCertificate || nsSshPublicKey")
(targetfilter="(objectClass=nsAccount)")(version 3.0; acl "Enable user password reset"; allow
(write, read)(groupdn="ldap:///cn=password_admins,ou=groups,dc=example,dc=com");)
EOF
```

2. 암호 관리자 역할을 그룹에 할당합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com localpwp set
ou=people,dc=example,dc=com --pwdadmin
"cn=password_admins,ou=groups,dc=example,dc=com"
```

7.3. 추가 리소스

- Directory Server 백업 (change it)

8장. 익명 바인딩 비활성화

사용자가 자격 증명을 제공하지 않고 Directory Server에 연결을 시도하는 경우 이 작업을 **anonymous bind** 라고 합니다. 익명 바인딩은 사용자가 먼저 인증하지 않아도 디렉터리에서 전화 번호를 찾는 것과 같은 검색 및 읽기 작업을 단순화합니다. 그러나 계정이 없는 사용자가 데이터에 액세스할 수 있기 때문에 익명 바인딩은 보안 위험이 될 수 있습니다.

주의

기본적으로 익명 바인딩은 검색 및 읽기 작업을 위해 Directory Server에서 활성화됩니다. 이를 통해 루트 디렉터리 서버 항목(DSE)과 같은 구성 항목과 구성 항목에 대한 무단 액세스를 허용합니다.

8.1. 명령줄을 사용하여 익명 바인딩 비활성화

보안을 강화하기 위해 익명 바인딩을 비활성화할 수 있습니다.

절차

- **nsslapd-allow-anonymous-access** 구성 매개변수를 **off** 로 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com config replace
nsslapd-allow-anonymous-access=off
```

검증

- 사용자 계정을 지정하지 않고 검색을 실행합니다.

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -x
ldap_bind: Inappropriate authentication (48)
 additional info: Anonymous access is not allowed
```

8.2. 웹 콘솔을 사용하여 익명 바인딩 비활성화

보안을 강화하기 위해 익명 바인딩을 비활성화할 수 있습니다.

사전 요구 사항

- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. **Server → Server Settings → Advanced Settings** 로 이동합니다.
2. **Allow Anonymous Access** 매개변수를 **off** 로 설정합니다.
3. **저장**을 클릭합니다.

검증

- 사용자 계정을 지정하지 않고 검색을 실행합니다.

```
# ldapsearch -H ldap://server.example.com -b "dc=example,dc=com" -x
ldap_bind: Inappropriate authentication (48)
        additional info: Anonymous access is not allowed
```

9장. 사용자 및 역할 수동 활성화

manually-inactivating-users-and-roles

Directory Server에서는 단일 사용자 계정 또는 계정 세트를 일시적으로 활성화할 수 있습니다. 계정이 비활성화되면 사용자는 디렉터리에 바인딩할 수 없습니다. 인증 작업이 실패합니다.

9.1. 명령줄을 사용하여 사용자 및 역할의 활성화 및 활성화

명령줄 또는 운영 특성을 사용하여 사용자 및 역할을 수동으로 활성화할 수 있습니다.

역할은 정적 그룹 및 동적 그룹으로 작동합니다. 그룹을 사용하면 항목이 그룹 항목에 멤버로 추가됩니다. 역할을 사용하면 역할 속성이 항목에 추가되고 해당 특성이 역할 항목의 멤버를 자동으로 식별하는 데 사용됩니다.

사용자와 역할은 동일한 절차를 실행할 수 없습니다. 그러나 역할이 비활성화되면 역할 항목이 아니라 역할의 멤버가 활성화됩니다.

사용자 및 역할을 활성화하려면 명령줄에서 다음 명령을 실행합니다.

- 사용자 계정을 활성화하려면 다음을 수행합니다.

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account lock "uid=user_name,ou=People,dc=example,dc=com"
```

- 역할 활성화를 위해 다음을 수행합니다.

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" role lock "cn=Marketing,ou=People,dc=example,dc=com"
```

사용자 및 역할을 활성화하려면 명령줄에서 다음 명령을 실행합니다.

- 사용자 계정 활성화의 경우:

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account unlock
"uid=user_name,ou=People,dc=example,dc=com"
```

- 역할 활성화의 경우:

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" role unlock "cn=Marketing,ou=People,dc=example,dc=com"
```

선택적으로 명령을 사용하는 대신 **nsAccountLock** 을 항목에 추가할 수 있습니다. 항목에 **true** 값이 있는 **nsAccountLock** 속성이 포함된 경우 서버는 바인딩을 거부합니다.

9.2. 계정 또는 역할의 상태를 표시하는 명령

명령줄에서 해당 명령을 사용하여 계정 또는 디렉터리 서버의 역할을 표시할 수 있습니다.

상태를 표시하는 명령

- 계정 상태를 표시합니다.

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account entry-status
"uid=user_name,ou=People,dc=example,dc=com"
Entry DN: uid=user_name,ou=People,dc=example,dc=com
Entry Creation Date: 20210813085535Z (2021-08-13 08:55:35)
Entry Modification Date: 20210813085535Z (2021-08-13 08:55:35)
Entry State: activated
```

선택 사항: **-V** 옵션은 추가 세부 정보를 표시합니다.

예 9.1. 활성 계정에 대한 자세한 출력

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account entry-status
"uid=user_name,ou=People,dc=example,dc=com" -V
Entry DN: uid=user_name,ou=People,dc=example,dc=com
Entry Creation Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Modification Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Last Login Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Time Until Inactive: 2 seconds (2021-08-24 16:07:45)
Entry State: activated
```

예 9.2. 비활성 계정에 대한 세부 출력

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account entry-status
"uid=user_name,ou=People,dc=example,dc=com" -V
Entry DN: uid=user_name,ou=People,dc=example,dc=com
Entry Creation Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Modification Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Last Login Date: 20210824160645Z (2021-08-24 16:06:45)
Entry Time Since Inactive: 3 seconds (2021-08-24 16:07:45)
Entry State: inactivity limit exceeded
```

- 역할 상태를 표시합니다.

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" role entry-status
"cn=Marketing,ou=People,dc=example,dc=com"
Entry DN: cn=Marketing,ou=people,dc=example,dc=com
Entry State: activated
```

- 하위 트리의 상태를 표시합니다.

```
# dsidm -D "cn=Directory Manager" ldap://server.example.com -b
"dc=example,dc=com" account subtree-status "ou=People,dc=example,dc=com" -f
"(uid=*)" -V -o "2021-08-25T14:30:30"
```

하위 트리에서 검색 결과를 필터링하려면 다음을 사용합니다.

  - 검색 필터를 설정하는 **-f** 옵션

- 검색 범위를 설정하는 **-s** 옵션
- 비활성 계정만 반환하는 **-i** 옵션
- 지정된 날짜 **YYYY-MM-DDTHH:MM:SS**전에 비활성 상태인 계정만 반환하는 **-o** 옵션

10장. 복제 환경의 모든 서버에서 계정 잠금 속성 동기화

Directory Server는 계정 잠금 특성을 로컬로 저장합니다. 여러 서버가 있는 환경에서는 이러한 속성에 대한 복제를 구성하여 계정 잠금 수에 도달한 다음 다른 서버에서 계속될 때까지 공격자가 하나의 서버에 로그인하지 못하도록 합니다.

10.1. 복제 환경에서 DIRECTORY SERVER에서 암호 및 계정 잠금 정책을 처리하는 방법

Directory Server는 다음과 같이 암호 및 계정 잠금 정책을 적용합니다.

- 암호 정책은 데이터 공급자에 적용됩니다.
- 복제 토폴로지의 모든 서버에 계정 잠금 정책이 적용됨

Directory Server는 다음과 같은 암호 정책 속성을 복제합니다.

- **passwordMinAge**
- **passwordMaxAge**
- **passwordExp**
- **passwordWarning**

그러나 기본적으로 Directory Server는 일반 계정 잠금 특성을 복제하지 않습니다.

- **passwordRetryCount**
- **retryCountResetTime**
- **accountUnlockTime**

공격자가 계정 잠금 수에 도달하고 다른 서버에서 계속할 때까지 한 서버에 로그인하지 못하도록 이러한 계정 잠금 특성을 복제합니다.

추가 리소스

- 계정 잠금 속성을 복제하도록 Directory Server 구성

10.2. 계정 잠금 속성을 복제하도록 DIRECTORY SERVER 구성

passwordRetryCount,**retryCountResetTime** 또는 **accountUnlockTime** 속성을 업데이트하는 계정 잠금 정책 또는 암호 정책을 사용하는 경우 이러한 속성을 복제하여 해당 값이 모든 서버에서 동일하게 Directory Server를 구성합니다.

복제 토폴로지의 모든 공급자에 대해 이 절차를 수행합니다.

사전 요구 사항

- 언급된 속성을 하나 이상 업데이트하는 계정 잠금 정책 또는 암호 정책을 구성했습니다.
- 복제 환경에서 Directory Server를 사용합니다.

절차

1. 암호 정책 속성 복제를 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com pwpolicy set --
pwdisglobal="on"
```

2. 소수 복제를 사용하는 경우 복제에서 제외된 속성 목록을 표시합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com repl-agmt get --suffix
"dc=example,dc=com" example-agreement | grep "nsDS5ReplicatedAttributeList"
```

기본 설정을 사용하여 출력이 표시되지 않으며 Directory Server는 계정 잠금 특성을 복제합니다. 그러나 명령에서 다음 예와 같이 제외된 속성 목록을 반환하는 경우 속성 목록을 확인합니다.

```
nsDS5ReplicatedAttributeList: (objectclass=*) $ EXCLUDE accountUnlockTime
passwordRetryCount retryCountResetTime example1 example2
```

이 예에서는 **accountUnlockTime**,**passwordRetryCount**, **retryCountResetTime** 잠금 정책 속성은 복제에서 제외되며 다른 두 가지 특성과 함께 복제에서 제외됩니다.

3. 이전 명령의 출력에 계정 잠금 속성 중 하나가 나열되는 경우 잠금 정책 속성 이외의 속성만 포함하도록 소수 복제 설정을 업데이트합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com repl-agmt set --suffix
"dc=example,dc=com" --frac-list "example1 example2" example-agreement
```

검증

1. 잘못된 암호를 사용하여 사용자로 검색을 시도합니다.

```
# ldapsearch -H ldap://server.example.com -D
"uid=example,ou=People,dc=example,dc=com" -w "invalid-password" -b
"dc=example,dc=com" -x
ldap_bind: Invalid credentials (49)
```

2. 사용자의 **passwordRetryCount** 특성을 표시합니다.

```
# ldapsearch -H ldap://server.example.com -D "cn=Directory Manager" -W -b
"uid=example,ou=People,dc=example,dc=com" -x passwordRetryCount
...
dn: uid=example,ou=People,dc=example,dc=com
passwordRetryCount: 1
```

3. 복제 토폴로지의 다른 서버에서 이전 명령을 실행합니다. **passwordRetryCount** 특성의 값이 동일한 경우 Directory Server가 특성을 복제합니다.

추가 리소스

- 암호 기반 계정 잠금 정책 구성

11장. 참조 무결성을 사용하여 항목 간 관계를 유지 관리

참조 무결성은 Directory Server가 관련 항목 간의 관계를 유지 관리하도록 하는 데이터베이스 메커니즘입니다. 이 기능을 사용하여 디렉터리의 한 항목에 대한 업데이트가 업데이트된 항목을 참조하는 다른 항목에 올바르게 반영되도록 할 수 있습니다.

예를 들어 디렉토리에서 사용자를 제거하고 Referential Integrity 플러그인이 활성화된 경우 서버는 사용자가 멤버인 모든 그룹에서 사용자를 제거합니다. 플러그인이 활성화되어 있지 않으면 관리자가 수동으로 제거할 때까지 사용자는 그룹의 멤버로 유지됩니다.

사용자 및 그룹 관리를 위해 Directory Server를 Directory Server를 사용하는 다른 제품과 통합하면 참조 무결성은 중요한 기능입니다.

11.1. REFERENTIAL INTEGRITY 플러그인이 작동하는 방식

Referential Integrity 플러그인을 활성화하면 **멤버**,**uniqueMember**,**owner** 에서 무결성 업데이트를 수행하고 작업 직후 기본적으로 **도 특성을 볼** 수 있습니다.

예를 들어 관리자가 디렉터리 내에서 그룹 또는 사용자를 삭제, 업데이트, 이름 변경 또는 이동하는 경우 Directory Server는 참조 무결성 로그 파일에 작업을 기록합니다. 그러면 Directory Server는 이 로그 파일의 고유 이름(DN)을 사용하고 플러그인 구성에 지정된 속성과 일치하는 항목을 검색한 다음 일치하는 항목을 업데이트합니다. 예를 들어 **cn=demo,dc=example,dc=com** 항목을 삭제한 후 플러그인은 **member** 속성이 **cn=demo,dc=example,dc=com** 으로 설정된 항목을 검색하고 이러한 **멤버** 속성을 제거합니다. 이후 플러그인은 **uniqueMember**,**owner** 및 **see** also 특성에 대해 동일하게 수행합니다.

기본적으로 Directory Server는 원래 작업과 동일한 트랜잭션에서 검색하고 업데이트합니다. 검색 및 업데이트 작업은 시간이 오래 걸릴 수 있으므로 원래 작업이 완료된 후 이를 지연할 수 있습니다. **dsconf 플러그인 referential-integrity set 명령의 --update-delay 옵션을 사용하여 원래 작업을 무결성** 업데이트와 분리할 수 있습니다.

수정 및 삭제 작업의 성능이 저하되지 않도록 하려면 참조 무결성 플러그인 구성에 지정한 속성을 인덱싱합니다.

추가 리소스

- 인덱스 관리

11.2. 명령줄을 사용하여 REFERENTIAL INTEGRITY 플러그인 구성

명령줄을 사용하여 Referential Integrity 플러그인을 구성할 수 있습니다.

복제 토폴로지의 모든 공급자에 대해 이 절차를 수행합니다.

절차

1. Referential Integrity 플러그인을 활성화합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity enable
```

2. 플러그인이 사용자 항목의 삭제 또는 이름 변경 작업을 검색하는 하위 트리를 설정합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity set --entry-scope "ou=People,dc=example,dc=com"
```

3. 선택 사항: 항목 범위 아래에 하위 트리를 제외합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity set --exclude-entry-scope "ou=Special Users,ou=People,dc=example,dc=com"
```

이 명령은 **ou=Special Users,ou=People,dc=example,dc=com** 하위 트리에서 수행되는 삭제 또는 이름 변경 작업을 무시하도록 플러그인을 구성합니다.

4. 플러그인에서 그룹 항목을 업데이트하는 하위 트리를 구성합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity set --container-scope "ou=Groups,dc=example,dc=com"
```

5. 기본적으로 플러그인은 **멤버**,**uniqueMember**,**owner** 에서 무결성 업데이트를 수행하고 **도 특성을 참조하십시오**. 다른 속성을 지정하려면 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity set --membership-attr attribute_1 attribute_2
```

이 명령은 플러그인 구성의 속성 목록을 덮어씁니다. 속성을 추가하려면 현재 속성 목록과 추가 속성을 **--membership-attr** 옵션에 전달합니다.

6. 선택 사항: 기본적으로 Directory Server는 참조 무결성 검사를 즉시 수행합니다. 지연을 설정하려면 다음을 입력합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity set --update-delay=5
```

이 명령은 참조 무결성 검사를 **5** 초로 지연합니다. 여러 공급업체에 대한 참조 무결성을 활성화한 경우 지연을 설정하면 복제 루프 및 디렉터리 불일치가 발생할 수 있습니다. 이러한 문제를 방지하려면 토폴로지의 한 공급자에서만 플러그인을 활성화합니다.

7. 인스턴스를 다시 시작하십시오.

```
# dsctl instance_name restart
```

검증

1. 참조 무결성 플러그인 구성을 표시합니다.

```
# dsconf -D "cn=Directory Manager" ldap://server.example.com plugin referential-integrity show
...
nsslapd-plugincontainerscope: ou=Groups,dc=example,dc=com
nsslapd-pluginentryscope: ou=People,dc=example,dc=com
...
referint-membership-attr: member
referint-membership-attr: uniquemember
referint-membership-attr: owner
referint-membership-attr: seeAlso
referint-update-delay: 0
...
```

2. 그룹의 멤버 특성을 표시하여 그룹의 **멤버** 를 나열합니다.

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"cn=demoGroup,ou=Groups,dc=example,dc=com" member
...
member: uid=demoUser,ou=People,dc=example,dc=com
```

3. **uid=demoUser,ou=People,dc=example,dc=com** 사용자를 삭제합니다.

```
# dsidm -D "cn=Directory manager" ldap://server.example.com -b
"dc=example,dc=com" user delete "uid=demoUser,ou=People,dc=example,dc=com"
```

4. 그룹 멤버를 다시 표시합니다.

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"cn=demoGroup,ou=People,dc=example,dc=com" member
```

uid=demoUser,ou=People,dc=example,dc=com 이 더 이상 그룹의 멤버로 나열되지 않으면 Referential Integrity 플러그인이 작동합니다.

11.3. 웹 콘솔을 사용하여 참조 무결성 플러그인 구성

Directory Server 웹 콘솔을 사용하여 Referential Integrity 플러그인을 구성할 수 있습니다.

복제 토폴로지의 모든 공급자에 대해 이 절차를 수행합니다.

사전 요구 사항

- 웹 콘솔에서 인스턴스에 로그인되어 있습니다.

절차

1. **플러그인 → 참조 무결성**.
2. 플러그인을 활성화합니다.
3. **Actions → Restart Instance** (인스턴스 재시작)를 클릭합니다.
4. **플러그인 → 참조 무결성**.
5. 기본적으로 플러그인은 **멤버**,**uniqueMember**,**owner** 에서 무결성 업데이트를 수행하고 **도 특성을 참조하십시오**. 다른 특성을 지정하려면 **멤버십 속성** 필드에서 목록을 업데이트합니다.
6. **Entry Scope** 필드를 플러그인에서 사용자 항목의 삭제 또는 이름 변경 작업을 검색해야 하는 하위 트리의 DN으로 설정합니다.
7. 선택 사항: 항목 범위 아래에 하위 트리를 제외하려면 **제외 항목 범위** 필드에 하위 트리의 DN을 입력합니다.
8. 컨테이너 **범위** 필드를 플러그인에서 그룹 항목을 업데이트해야 하는 하위 트리의 DN으로 설정합니다.
9. 선택 사항: 참조 무결성 로그 파일의 경로를 업데이트합니다. Directory Server는 이 파일을 사용하여 디렉터리의 변경 사항을 추적합니다. **dirsrv** 사용자에게 이 위치에 대한 쓰기 권한이 있어야 합니다.

10. 선택 사항: 기본적으로 Directory Server는 참조 무결성 검사를 즉시 수행합니다. 지연을 설정하려면 **업데이트 지연** 필드에 설정합니다.
    여러 공급업체에 대한 참조 무결성을 활성화한 경우 지연을 설정하면 복제 루프 및 디렉터리 불일치가 발생할 수 있습니다. 이러한 문제를 방지하려면 토폴로지의 한 공급자에서만 플러그인을 활성화합니다.

11. **Save Config** 를 클릭합니다.

검증

1. 그룹의 멤버 특성을 표시하여 그룹의 **멤버** 를 나열합니다.

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"cn=demoGroup,ou=Groups,dc=example,dc=com" member
...
member: uid=demoUser,ou=People,dc=example,dc=com
```

2. **uid=demoUser,ou=People,dc=example,dc=com** 사용자를 삭제합니다.

```
# dsidm -D "cn=Directory manager" ldap://server.example.com -b
"dc=example,dc=com" user delete "uid=demoUser,ou=People,dc=example,dc=com"
```

3. 그룹 멤버를 다시 표시합니다.

```
# ldapsearch -D "cn=Directory Manager" -W -H ldap://server.example.com -b
"cn=demoGroup,ou=People,dc=example,dc=com" member
```

uid=demoUser,ou=People,dc=example,dc=com 이 더 이상 그룹의 멤버로 나열되지 않으면 Referential Integrity 플러그인이 작동합니다.