Red Hat Directory Server 12

Directory Server 계획 및 설계

효과적인 디렉터리 서비스 계획을 위한 개념 및 구성 옵션

Last Updated: 2024-08-25

Red Hat Directory Server 12 Directory Server 계획 및 설계

효과적인 디렉터리 서비스 계획을 위한 개념 및 구성 옵션

법적 공지

초록

디렉터리 트리, 스키마, 토폴로지, 복제 및 보안을 포함하여 디렉터리 설계의 측면에 대해 알아봅니다. 디렉터리 서비스의 이점 및 옵션, Directory Server 구현 전략 및 높은 수준의 구성 예제에 대해 자세히 알아보십시오.

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. 디렉터리 서비스 소개

Red Hat Directory Server는 중앙 집중식 디렉터리 서비스를 제공합니다. Directory Server는 기존 시스템과 통합되고 직원, 고객, 공급업체 및 파트너 정보의 통합을 위한 중앙 집중식 리포지토리 역할을 합니다. Directory Server를 사용하면 사용자 프로필 및 인증을 관리할 수 있습니다.

다음 장에서 디렉토리를 설계하기 전에 이해해야 할 사항에 대해 알아보십시오.

1.1. 디렉터리 서비스 정보

*디렉터리 서비스*는 엔터프라이즈에 대한 정보를 저장하고 이 정보에 대한 액세스를 사용자에게 제공하는 소프트웨어, 하드웨어 및 프로세스의 컬렉션입니다. 디렉터리 서비스는 하나 이상의 Directory Server 인스턴스와 디렉터리 클라이언트 애플리케이션으로 구성됩니다. 클라이언트 애플리케이션은 디렉터리에 저장된 이름, 전화 번호, 주소 및 기타 데이터에 액세스할 수 있습니다.

디렉터리 서비스의 예는 DNS(Domain Name System) 서버입니다. DNS는 컴퓨터 호스트 이름을 IP 주소에 매핑합니다. DNS 클라이언트는 DNS 서버로 요청을 전송하고 서버는 server.example.com의 IP 주소를 응답합니다. 따라서 모든 호스트가 DNS 서버의 클라이언트가 됩니다. 또한 사용자는 IP 주소가 아닌 호스트 이름을 기억하여 네트워크에서 컴퓨터를 쉽게 찾을 수 있습니다. DNS 서버의 제한은 호스트 이름과 IP 주소의 두 가지 정보만 저장한다는 것입니다. 실제 디렉토리 서비스는 사실상 무제한의 정보를 저장합니다.

Red Hat Directory Server에서는 다음 데이터를 네트워크 액세스 가능한 하나의 리포지토리에 저장할 수 있습니다.

- 조직의 프린터 데이터(예: 위치, 제조업체, 구매 날짜, 일련 번호)와 같은 물리적 장치 정보입니다.
- 공용 직원 정보: 이름, 이메일 주소, 부서.
- 직원 정보: 급여, 정부 식별 번호, 집 주소, 전화번호, 급여.
- 계약 또는 계정 정보: 고객 이름, 최종 전달 날짜, 제안 정보, 계약 번호 및 프로젝트 날짜.

Directory Server는 포함된 정보에 액세스하고 많은 애플리케이션 요구 사항을 제공하는 표준 프로토콜 및 API(애플리케이션 프로그래밍 인터페이스)를 제공합니다.

1.1.1. 글로벌 디렉터리 서비스 정보

Red Hat Directory Server는 다양한 애플리케이션에 정보를 제공하여 글로벌 디렉터리 서비스를 제공합니다. 최근까지 많은 애플리케이션이 자체 전용 사용자 데이터베이스와 함께 번들화되었으며 해당 애플리케이션과 관련된 사용자에 대한 정보가 제공되었습니다. 하나의 애플리케이션만 사용하는 경우 독점 데이터베이스가 편리하지만 데이터베이스가 동일한 정보를 관리하는 경우 여러 데이터베이스가 관리 부담이 됩니다.

예를 들어 한 회사는 세 개의 다른 독점 이메일 시스템을 실행하고 각 이메일 시스템에는 고유한 독점 디렉터리 서비스가 있습니다. 사용자가 한 디렉터리에서 암호를 변경하면 변경 사항이 다른 디렉터리에 자동으로 복제되지 않습니다. 다른 위치에서 동일한 정보를 관리하면 하드웨어 및 인력 비용이 증가합니다. 증가된 유지 관리 오버헤드를 *n+1 디렉터리 문제라고* 합니다.

글로벌 디렉터리 서비스는 모든 애플리케이션에서 액세스할 수 있는 중앙 집중식 리포지토리를 제공하여 n+1 디렉터리 문제를 해결합니다. 그러나 디렉터리 서비스에 다양한 애플리케이션을 사용하려면 애플리케이션과 디렉터리 서비스 간에 통신할 네트워크 기반 수단이 필요합니다.

Red Hat Directory Server는 애플리케이션에 LDAP를 사용하여 글로벌 디렉터리 서비스에 액세스합니다.

1.1.2. LDAP 정보

LDAP에서는 클라이언트 애플리케이션 및 서버가 서로 통신하는 데 사용하는 공통 언어를 제공합니다. LDAP는 ISO X.500 표준에서 설명하는 DAP(Directory Access Protocol)의 "Lightweight" 버전입니다.

DAP는 확장 가능하고 강력한 정보 프레임워크를 통해 모든 애플리케이션에 액세스할 수 있지만 관리 비용이 높습니다. DAP는 인터넷 표준 프로토콜이 아니며 복잡한 디렉터리-나밍 규칙이 있는 통신 계층을 사용합니다.

LDAP는 관리 비용을 절감하면서 DAP의 최상의 기능을 유지합니다. LDAP는 TCP/IP 및 간소화된 인코딩 방법을 통해 실행되는 오픈 디렉터리 액세스 프로토콜을 사용합니다. 데이터 모델을 유지하고 하드웨어 및 네트워크 인프라에 대한 모드 투자를 위해 수백만 개의 항목을 지원할 수 있습니다.

1.2. 디렉터리 서버 소개

Red Hat Directory Server에는 여러 구성 요소가 있습니다. 디렉터리 코어는 LDAP 프로토콜을 구현하는 서버입니다. Red Hat Directory Server와 함께 LDAP SDK를 사용하여 작성한 다양한 LDAP 클라이언트, 타사 및 사용자 지정 애플리케이션을 사용할 수 있습니다.

Red Hat Directory Server 설치에는 다음과 같은 요소가 포함되어 있습니다.

- 코어 Directory Server LDAP 서버, LDAP v3 호환 네트워크 데몬(**ns-slapd**) 및 모든 관련 플러그인, 서버와 데이터베이스 관리를 위한 명령줄 도구, 구성 및 스키마 파일. 디렉터리 데이터베이스 구성 및 구성 및 스키마 참조 에서 자세히 알아보십시오.
- 디렉터리 서비스 설정 및 유지 관리를 단순화하는 그래픽 관리 콘솔인 웹 콘솔입니다. 웹 콘솔을 사용하여 Directory Server에 로그인 하는 방법에 대해 자세히 알아보십시오.
- SNMP(Simple Network Management Protocol)를 사용하여 Directory Server를 모니터링하는 SNMP 에이전트. SNMP를 사용한 모니터링 디렉터리 서버 모니터링 에서 자세히 알아보십시오.

Directory Server는 다른 LDAP 클라이언트 애플리케이션 없이 인트라넷 또는 엑스트라넷을 위한 기반을 제공합니다. 호환 가능한 서버 애플리케이션은 이 디렉토리를 직원, 고객, 공급업체 및 파트너 데이터와 같은 공유 서버 정보를 위한 중앙 리포지토리로 사용합니다. Directory Server는 사용자 인증, 액세스 제어, 사용자 기본 설정을 관리합니다. 호스팅된 환경에서 파트너, 고객 및 공급업체는 디렉터리 부분을 관리하여 관리 비용을 줄일 수 있습니다.

디렉터리 서버는 데이터베이스 계층, 복제 및 연결 데이터베이스와 같은 추가 기능을 위해 *플러그인* 을 사용합니다. 코어 디렉터리 서비스 작업과 관련이 없는 플러그인을 비활성화할 수 있습니다.

1.2.1. 디렉터리 서버 프런트 엔드 개요

Directory Server는 다중 스레드 애플리케이션입니다. 즉, 동일한 네트워크를 통해 여러 클라이언트가 동시에 서버에 바인딩할 수 있습니다. 디렉터리 서비스가 더 많은 수의 항목 또는 지리적으로 분산된 클라이언트를 포함하도록 확장되면 이 서비스에는 네트워크 주변의 전략적 위치에 배치된 여러 디렉터리 서버도 포함됩니다.

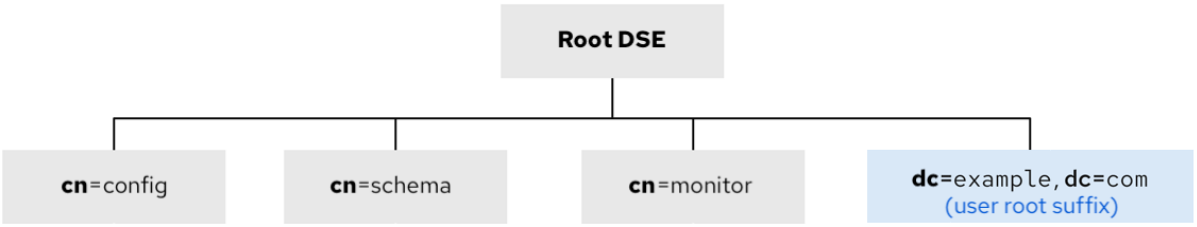
Directory Server의 서버 프런트 엔드는 LDAP over TCP/IP 및 LDAP over Unix 소켓 (LDAPI)을 사용하여 디렉터리 클라이언트 애플리케이션과의 통신을 관리합니다.

Directory Server는 클라이언트가 연결에 TLS를 사용하도록 협상하는지에 따라 TLS(Transport Layer Security)를 사용하여 보안(암호화) 연결을 설정할 수 있습니다. 클라이언트가 발급된 인증서인 경우 Directory Server는 TLS를 사용하여 클라이언트가 서버에 액세스할 수 있는 권한이 있는지 확인할 수 있습니다. 또한 TLS는 메시지 무결성 검사, 디지털 서명 및 서버 간 상호 인증과 같은 기타 보안 활동을 수행하는 데 사용됩니다.

1.2.2. 기본 디렉터리 서버 트리 개요

디렉터리 트리(DIT)는 대부분의 파일 시스템에서 사용하는 트리 모델을 미러링합니다. 설치하는 동안 Directory Server는 기본 디렉터리 트리를 만듭니다.

기본 디렉터리 트리



설치 후 디렉터리에는 다음 루트 접미사 및 하위 트리가 포함됩니다.

- **루트 DSE** (Root DSA별 항목)는 LDAP 서버의 특수 항목입니다. **루트 DSE** 고유 이름(DN)은 길이가 0인 문자열입니다.
- **CN=config** 에는 서버 내부 구성에 대한 정보가 포함되어 있습니다.
- **CN=monitor** 에는 서버 및 데이터베이스 모니터링 통계가 포함되어 있습니다.
- **CN=schema** 에는 현재 서버에 로드된 스키마 요소가 포함되어 있습니다.
- **사용자 루트 접미사**: 설치 중에 Directory Server가 생성하는 기본 사용자 데이터베이스의 접미사입니다. Directory Server 인스턴스를 생성할 때 사용자 루트 접미사 이름을 정의합니다. 사용자 root 접미사에는 **dc =example,dc=com** 과 같은 dc 이름 지정 규칙이 있거나 **o =example.com** 과 같은 조직에 o 속성을 사용하는 경우가 많습니다. 사용자 접미사 이름을 지정하는 방법에 대한 자세한 내용은 접미사 선택을 참조하십시오. 루트 사용자 접미사는 **userRoot** 데이터베이스와 연결됩니다. LDIF 파일을 가져오거나 항목을 생성하여 나중에 데이터베이스를 채웁니다.

디렉터리 설치와 관련된 데이터를 추가하여 기본 디렉터리 트리를 확장할 수 있습니다. 디렉터리 트리에 대한 자세한 내용은 디렉터리 트리 설계를 참조하십시오.

1.3. 디렉터리 서버 데이터 스토리지

데이터베이스는 스토리지, 성능, 복제 및 인덱싱의 기본 단위입니다. 데이터베이스에서 가져오기, 내보내기, 백업, 복원, 인덱싱 등의 작업을 수행할 수 있습니다. Directory Server는 **LDAP Database Manager** (LDBM) 데이터베이스에 데이터를 저장합니다. LDBM 데이터베이스는 디렉터리와 함께 자동으로 설치되고 기본적으로 활성화되어 있는 플러그인으로 구현됩니다.

기본적으로 Directory Server는 루트 접미사에 하나의 백엔드 데이터베이스 인스턴스를 사용하고 단일 데이터베이스는 디렉터리 트리를 포함하는 데 충분합니다. 이 데이터베이스는 수백만 개의 항목을 관리할 수 있습니다. 이 데이터베이스는 데이터 손실을 최소화하기 위해 데이터를 백업 및 복원하는 고급 방법을 지원합니다.

그러나 여러 데이터베이스를 사용하여 단일 데이터베이스에 저장할 수 있는 것보다 더 많은 데이터를 관리하기 위해 전체 Directory Server 배포를 지원할 수 있습니다.

1.3.1. 디렉터리 항목 정보

LDAP Data Interchange Format(LDIF) 은 디렉터리 항목을 설명하는 표준 텍스트 기반 형식입니다. 항목은 LDIF 파일의 여러 줄로 구성되며 조직의 사람 또는 네트워크의 프린터와 같은 오브젝트에 대한 정보를 포함합니다.

항목에 대한 정보는 일련의 속성 및 해당 값으로 표시됩니다. 각 항목에는 항목이 설명하는 오브젝트 유형을 지정하고 포함된 추가 특성 집합을 정의하는 오브젝트 클래스 속성이 있습니다. 각 속성은 항목의 특정 특성을 설명합니다.

예를 들어 항목에 **조직의 사람을 나타내는 organizationPerson** 오브젝트 클래스가 있을 수 있습니다. 이 오브젝트 클래스는 **givenName** 및 **telephoneNumber** 특성을 지원합니다. 이러한 속성에 할당된 값은 항목이 제공하는 사람의 이름과 전화 번호를 정의합니다.

Directory Server는 또한 서버에서 계산한 읽기 전용 *작동 속성을* 사용합니다. 관리자는 액세스 제어 및 기타 서버 기능에 대해 이러한 운영 속성을 수동으로 설정할 수 있습니다.

디렉터리 항목 검색 수행

디렉터리 트리는 계층 구조의 항목을 저장합니다. LDAP는 항목을 쿼리하고 디렉터리 트리의 분기 아래에 있는 모든 항목을 요청하는 툴을 지원합니다. 분기 하위 트리의 루트는 기본 고유 이름 또는 *기본 DN* 이라고 합니다. 예를 들어 **ou=people,dc=example,dc=com** 의 기본 DN을 지정하는 LDAP 검색 요청을 수행하는 경우 검색 작업에서 **dc=example,dc=com** 디렉터리 트리의 **ou=people** 하위 트리만 확인합니다.

기본적으로 LDAP 검색에서 모든 항목을 반환하지 않고 **ldapsubentry** 오브젝트 클래스가 있는 관리 항목을 제외합니다. 관리 항목을 사용하여 역할 또는 서비스 클래스를 정의할 수 있습니다. 검색 응답에 이러한 항목을 포함하려면 클라이언트 애플리케이션에서 **ldapsubentry** 오브젝트 클래스를 사용하여 항목을 추가로 검색해야 합니다.

추가 리소스

- Directory Server의 역할 정보
- 명령줄을 사용하여 항목 검색(ldapsearch)

1.3.2. 디렉터리 데이터 배포

디렉터리 트리의 일부를 별도의 데이터베이스에 저장하면 디렉터리에서 클라이언트 요청을 병렬로 처리할 수 있습니다. 성능 향상을 위해 데이터베이스를 다른 머신에 저장할 수도 있습니다. 디렉터리의 일부를 연결하기 위해 Directory Server는 데이터베이스 링크 및 체인을 사용합니다. 데이터베이스 링크 및 체인에 대한 자세한 내용은 Directory Server에서 참조 사용을 참조하십시오.

추가 리소스

- 디렉터리 데이터 배포

1.4. 설계 프로세스 개요

1. 디렉터리 데이터 계획
   디렉터리에는 사용자 이름, 전화 번호 및 그룹 세부 정보와 같은 데이터가 포함되어 있습니다. 계획 장은 조직의 다양한 데이터 소스를 분석하고 해당 관계를 이해하는 데 도움이 됩니다. 디렉터리가 저장할 수 있는 데이터 유형과 Directory Server의 콘텐츠를 설계하기 위해 수행할 작업을 파악합니다.

2. 디렉터리 스키마 설계
   디렉터리는 하나 이상의 디렉터리 지원 애플리케이션을 지원하도록 설계되었습니다. 이러한 애플리케이션에는 파일 형식과 같이 디렉터리가 저장하는 데이터의 요구 사항이 있습니다. 디렉터리 스키마는 이 데이터의 특성을 결정합니다. Directory Server와 함께 제공되는 표준 스키마, 스키마

를 사용자 지정하는 방법에 대한 설명 및 일관된 스키마를 유지 관리하기 위한 팁에 대해 알아봅니다.

3. 디렉터리 트리 설계
   데이터 계층 구조 설계 및 예제 개요를 읽은 후 저장된 데이터를 구성하고 참조하는 방법을 결정합니다.

4. 디렉터리 토폴로지 설계
   디렉토리를 여러 물리적 디렉터리 서버로 분할하고 이러한 서버가 서로 통신하는 방법을 알아보려면 토폴로지 설계에 대해 알아봅니다.

5. 복제 프로세스 설계
   복제 개념, 복제 가능 데이터 유형, 다양한 복제 시나리오 및 고가용성 디렉터리 서비스 팁에 대해 알아봅니다.

6. 보안 디렉터리 설계
   디렉토리를 보호하는 방법을 찾으십시오. 보안 위협, 보안 방법 개요, 보안 분석 단계, 디렉터리 데이터 무결성을 보호하기 위해 액세스 제어를 설계하기 위한 팁에 대해 알아보십시오.

7. 동기화 설계
   혼합 플랫폼 인프라에서 Microsoft Active Directory 데이터베이스에 저장된 정보와 동기화를 고려하십시오.

1.5. 디렉터리 배포

먼저 테스트 서버 인스턴스를 설치하여 서비스가 사용자 부하를 처리할 수 있는지 확인합니다. 서비스에 최적의 초기 구성이 없는 경우 설계를 조정하고 다시 테스트합니다. 엔터프라이즈 요구 사항을 충족할 때까지 설계를 조정합니다.

성공적인 테스트 Directory Server 인스턴스를 생성하고 튜닝한 후 다음과 같은 고려 사항을 고려하여 디렉터리 서비스를 프로덕션으로 이동할 계획을 개발합니다.

- 필요한 리소스의 추정치
- 해야 할 일 및 시기의 일정
- 배포 성공을 측정하기 위한 일련의 기준

1.6. 추가 리소스

디렉터리, LDAP 및 LDIF의 주요 리소스:

- RFC 2849: LDAP Data Interchange Format (LDIF) 기술 사양
- RFC 2251: Lightweight Directory Access Protocol (v3)

2장. 디렉터리 데이터 계획

디렉터리 데이터는 사용자 이름, 이메일 주소, 전화 번호, 사용자 그룹 및 기타 정보를 포함할 수 있습니다. 디렉터리에 저장하려는 데이터 유형에 따라 디렉터리 구조, 데이터에 지정된 액세스 권한, 이 액세스가 요청 및 부여되는 방법이 결정됩니다.

2.1. 디렉터리 데이터 소개

디렉터리에 적합한 데이터는 다음과 같은 특징이 있습니다.

- 데이터는 작성된 것보다 더 자주 읽습니다.
- 데이터는 attribute-value 형식으로 표현 가능합니다(예: **surname=jensen**).
- 데이터는 한 개인 또는 그룹뿐만 아니라 유용합니다. 예를 들어 여러 사용자와 애플리케이션에서 직원 이름 또는 프린터 위치를 사용할 수 있습니다.
- 데이터는 둘 이상의 물리적 위치에서 액세스할 수 있습니다.

예를 들어, 애플리케이션의 단일 인스턴스만 정보에 액세스해야 하므로 소프트웨어 애플리케이션에 대한 직원의 기본 설정은 디렉터리에 적합하지 않습니다. 그러나 애플리케이션에서 디렉터리의 기본 설정 내용을 읽고 사용자가 다른 사이트의 기본 설정에 따라 애플리케이션을 사용하려는 경우 디렉터리에 이러한 설정을 포함하는 것이 유용합니다.

2.1.1. 디렉터리에 포함할 정보

특성으로 개인 또는 자산에 대한 유용한 정보를 항목에 추가할 수 있습니다. 예를 들면 다음과 같습니다.

- 전화 번호, 실제 주소 및 이메일 주소와 같은 연락처 정보
- 직원 번호, 직책, 관리자 또는 관리자 식별, 직무 관련 관심 분야와 같은 설명적 정보입니다.
- 전화 번호, 실제 주소, 관리자 식별 및 비즈니스 설명과 같은 조직 연락처 정보.
- 프린터 물리적 위치, 프린터 유형, 프린터가 생성할 수 있는 분당 페이지 수와 같은 장치 정보입니다.
- 기업 거래 파트너, 고객 및 고객의 연락처 및 청구 정보.
- 고객 이름, 마감일, 작업 설명, 가격 정보와 같은 계약 정보입니다.
- 개별 소프트웨어 기본 설정 또는 소프트웨어 구성 정보.
- 웹 서버 포인터 또는 특정 파일 또는 애플리케이션의 파일 시스템과 같은 리소스 사이트.

서버 관리 이외의 용도로 Directory Server를 사용하려면 디렉터리에 저장할 다른 유형의 정보를 계획해야 합니다. 예를 들어 다음 정보 유형을 포함할 수 있습니다.

- 계약 또는 클라이언트 계정 세부 정보
- 급여 데이터
- 물리적 장치 정보
- 홈 연락처 정보

- 회사 내에서 다른 사이트의 사무실 연락처 정보

2.1.2. 디렉터리에서 제외할 정보

Red Hat Directory Server는 클라이언트 애플리케이션이 읽기 및 경우에 따라 업데이트되는 대규모 데이터 볼륨을 관리하지만 Directory Server는 이미지 또는 기타 미디어와 같은 구조화되지 않은 큰 개체를 처리하도록 설계되지 않았습니다. 이러한 오브젝트는 파일 시스템에서 유지 관리해야 합니다. 그러나 디렉터리는 FTP, HTTP 및 기타 URL 유형을 사용하여 이러한 유형의 애플리케이션에 대한 포인터를 저장할 수 있습니다.

2.2. 디렉터리 요구 정의

디렉터리 데이터를 설계할 때 현재 필요한 데이터뿐만 아니라 디렉터리(및 조직)가 시간이 지남에 따라 변경될 수 있는 방법에 대해서도 생각할 수 있습니다. 설계 프로세스 중에 디렉터리의 향후 요구 사항을 고려하여 디렉터리의 데이터가 구조화 및 분산되는 방식에 영향을 미칩니다.

다음 사항을 고려하십시오.

- 현재 디렉터리에는 무엇이 포함되어 있습니까?
- 디렉터리를 배포하여 해결하려는 즉각적인 문제는 무엇입니까?
- 사용하는 디렉터리 지원 애플리케이션의 즉시 필요한 것은 무엇입니까?
- 가까운 시일 내에 디렉터리에 추가하고 싶은 것은 무엇입니까? 예를 들어, 기업은 현재 LDAP를 지원하지 않는 계정 패키지를 사용하지만 이 계정 패키지는 몇 개월 후에 LDAP를 사용하도록 설정합니다. LDAP 호환 애플리케이션에서 사용하는 데이터를 식별하고 기술을 사용할 수 있게 되면 데이터를 디렉터리로 마이그레이션할 계획입니다.
- 향후 디렉터리에 저장하려는 정보는 무엇입니까? 예를 들어 호스팅 회사는 이미지 또는 미디어 파일을 저장하는 등 현재 고객과 다른 데이터 요구 사항을 가진 향후 고객을 보유할 수 있습니다. 이러한 방식으로 계획하면 고려하지 않은 데이터 소스를 식별하는 데 도움이 됩니다.

2.3. 사이트 설문 조사 수행

*사이트 설문 조사*는 디렉터리 콘텐츠를 검색하고 특성화하는 공식적인 방법입니다. 디렉터리 아키텍처에 대한 준비가 중요하므로 설문 조사를 수행하는 데 더 많은 시간을 계획합니다. 사이트 설문 조사는 다음 작업으로 구성됩니다.

- 디렉터리를 사용하는 애플리케이션을 식별합니다.
  엔터프라이즈에 배포하는 디렉터리 지원 애플리케이션과 해당 데이터 요구 사항을 결정합니다.
- 데이터 소스 식별.
  엔터프라이즈를 조사하고 Active Directory, 기타 LDAP 서버, PBX 시스템, 인적 리소스 데이터베이스 및 이메일 시스템을 포함한 데이터 소스를 식별합니다.
- 디렉터리에 포함해야 하는 데이터를 특성화합니다.
  디렉터리에 있어야 하는 오브젝트(예: 사용자 이름 및 그룹)와 디렉터리에 유지할 이러한 오브젝트의 속성(예: 사용자 이름 및 암호)을 결정합니다.
- 제공할 서비스 수준을 결정합니다.
  클라이언트 애플리케이션에 대한 디렉터리 데이터의 가용성을 결정하고 그에 따라 아키텍처를 설계합니다. 디렉터리 가용성은 원격 서버에 저장된 데이터를 연결하기 위해 데이터 복제 및 연결 정책을 구성하는 방법에 영향을 미칩니다.

- 데이터 공급자를 식별합니다.
  데이터 공급자에는 디렉터리 데이터에 대한 기본 소스가 포함되어 있습니다. 로드 밸런싱 및 복구를 위해 이 데이터를 다른 서버에 미러링할 수 있습니다. 각 데이터 세그먼트에 대한 데이터 공급자를 결정합니다.

- 데이터 소유권을 결정합니다.
  모든 데이터에 대해 데이터 업데이트에 대한 책임이 있는 사람을 결정합니다.

- 데이터 액세스를 결정합니다.
  다른 소스에서 데이터를 가져올 때 대량 가져오기 및 증분 업데이트에 대한 전략을 개발합니다. 이 전략의 일부로 데이터를 한 곳에서 관리하고 데이터를 변경할 수 있는 애플리케이션 수를 제한합니다. 또한 지정된 데이터에 쓰는 사용자의 수를 제한합니다. 소규모 그룹은 관리 오버헤드를 줄이는 동시에 데이터 무결성을 보장합니다.

- 사이트 설문 조사를 문서화합니다.

디렉터리의 여러 조직에 영향을 미치는 경우 영향을 받는 각 조직의 대표자가 포함된 디렉터리 배포 팀을 생성하여 사이트 설문 조사를 수행하는 것이 좋습니다.

기업은 일반적으로 인사관리 부서, 회계 또는 계정, 제조 기관, 영업 조직 및 개발 조직을 보유하고 있습니다. 각 조직의 대표를 포함하면 설문 조사 프로세스를 수행하고 로컬 데이터 저장소에서 중앙 집중식 디렉터리로 마이그레이션하는 데 도움이 될 수 있습니다.

2.3.1. 디렉터리를 사용하는 애플리케이션 식별

디렉터리에 액세스하는 애플리케이션 및 이러한 애플리케이션의 데이터 요구 사항은 디렉터리 콘텐츠 계획을 안내합니다. 디렉터리를 사용하는 다양한 일반적인 애플리케이션은 다음과 같습니다.

- *온라인 전화 서적과 같은 디렉터리 브라우저 애플리케이션* . 사용자가 필요로 하는 정보를 결정하고 디렉터리에 포함합니다.

- *이메일 애플리케이션, 특히 이메일 서버*. 모든 이메일 서버에는 디렉터리에서 일부 라우팅 정보를 사용할 수 있어야 합니다. 그러나 일부에는 사용자 username이 저장된 디스크의 위치, 휴가 알림 세부 정보 및 프로토콜 정보(예: Cryostat 대 POP)와 같은 고급 정보가 필요할 수 있습니다.

- *디렉터리 지원 인적 리소스 애플리케이션*. 여기에는 정부 식별 번호, 집 주소, 집 전화번호, 생년월일, 급여 및 직책과 같은 추가 개인정보가 필요합니다.

- *Microsoft Active Directory*. Windows 사용자 동기화를 통해 Windows 디렉터리 서비스를 통합하여 Directory Server와 함께 작동할 수 있습니다. 두 디렉토리 모두 사용자 정보와 그룹 정보를 저장할 수 있습니다. 사용자, 그룹 및 기타 디렉터리 데이터를 동기화할 수 있도록 기존 Windows 서버 배포 후 Directory Server 배포를 구성합니다.

디렉터리를 사용할 애플리케이션을 평가할 때 각 애플리케이션에서 사용하는 정보 유형을 고려하십시오. 다음 표에서는 애플리케이션의 예와 애플리케이션에서 사용하는 정보를 제공합니다.

표 2.1. 애플리케이션 데이터 요구 예

애플리케이션	데이터 클래스	data
전화북	사람	이름, 이메일 주소, 전화번호, 사용자 ID, 비밀번호, 부서 번호, 매니저, 메일 중지

애플리케이션	데이터 클래스	data
웹 서버	사람, 그룹	사용자 ID, 암호, 그룹 이름, 그룹 멤버, 그룹 소유자
일정 서버	자주하는 질문	이름, 사용자 ID, 항목 번호, 컨퍼런스실 이름

각 애플리케이션에서 사용하는 애플리케이션 및 정보를 식별하면 둘 이상의 애플리케이션에서 사용하는 데이터 유형을 파악할 수 있습니다. 계획의 이 단계는 디렉터리의 데이터 중복성을 방지하고 데이터 디렉터리 종속 애플리케이션에 필요한 데이터를 명확하게 표시할 수 있습니다.

다음 요인은 디렉터리에서 유지 관리되는 데이터 유형과 정보를 디렉터리로 마이그레이션할 때 최종 결정에 영향을 미칩니다.

- 다양한 레거시 애플리케이션 및 사용자에게 필요한 데이터
- 레거시 애플리케이션이 LDAP 디렉터리와 통신할 수 있는 기능

2.3.2. 데이터 소스 식별

디렉터리에 포함할 모든 데이터를 확인하려면 기존 데이터 저장소에 대한 설문 조사를 수행합니다. 설문 조사에는 다음이 포함되어야 합니다.

- 정보를 제공하는 조직을 식별합니다.
  정보 서비스, 인적 자원, 급여 및 회계 부서와 같은 중요한 정보를 관리하는 모든 조직을 찾으십시오.
- 정보 소스인 툴 및 프로세스를 식별합니다.
  일반적인 정보에는 네트워킹 운영 체제(예: Windows, Novell Netware, UNIX NIS), 이메일 시스템, 보안 시스템, PBX(tele phone switching) 시스템 및 인적 리소스 애플리케이션이 포함됩니다.
- 각 데이터를 중앙 집중화하는 것이 데이터 관리에 어떤 영향을 미치는지 결정합니다.
  중앙 집중식 데이터 관리에는 새로운 도구와 새로운 프로세스가 필요할 수 있습니다. 경우에 따라 중앙 집중화에 조직의 직원 및 실패가 필요할 수 있습니다.

설문 조사 중에 아래 표와 같이 엔터프라이즈의 모든 정보 소스를 식별하는 매트릭스를 개발하십시오.

표 2.2. 정보 소스 예

데이터 소스	데이터 클래스	data
인사관리 데이터베이스	사람	이름, 주소, 전화번호, 부서 번호, 매니저
이메일 시스템	사용자, 그룹	이름, 이메일 주소, 사용자 ID, 비밀번호, 이메일 기본 설정
시설 시스템	기능	이름, 플로어 이름, 항목 번호, 액세스 코드 구축

2.3.3. 디렉터리 데이터 특성화

디렉터리에 포함할 데이터를 다음과 같은 방식으로 특성화합니다.

- 형식
- 크기
- 다양한 애플리케이션의 발생 수
- 데이터 소유자
- 다른 디렉터리 데이터와의 관계

디렉터리에 포함할 데이터에서 일반적인 특성을 찾습니다. 이렇게 하면 디렉터리 스키마 설계에 설명된 스키마 설계 단계에서 시간을 절약할 수 있습니다.

디렉터리 데이터를 특징으로 하는 아래 표를 고려하십시오.

표 2.3. 디렉터리 데이터 특성

data	형식	크기	소유자	관련 정보
직원 이름	텍스트 문자열	128자	인적 리소스	사용자 항목
Fax 번호	전화 번호	14자리	기능	사용자 항목
이메일 주소	텍스트	여러 문자	IS 부서	사용자 항목

2.3.4. 서비스 수준 확인

제공하는 서비스 수준은 디렉터리 지원 애플리케이션에 의존하는 사용자의 기대치에 따라 다릅니다. 각 애플리케이션에 필요한 서비스 수준을 확인하려면 애플리케이션이 사용되는 방법과 시기를 결정합니다.

디렉터리가 진화하면 이 디렉터리는 프로덕션에서 미션 크리티컬 수준에 이르기까지 다양한 서비스 수준을 지원해야 할 수 있습니다. 디렉토리 배포 후 서비스 수준을 올리기 어렵기 때문에 초기 설계가 향후 요구 사항을 충족하는지 확인합니다.

예를 들어 총 실패 위험을 제거하려면 여러 공급업체가 동일한 데이터를 처리하는 다중 공급 업체 구성을 사용합니다.

2.3.5. 데이터 공급자 고려

*데이터 공급자*는 *데이터*를 제공하는 서버입니다. 여러 위치에 동일한 정보를 저장하면 데이터 무결성이 저하됩니다. 데이터 공급자는 여러 위치에 저장된 모든 정보가 일관되고 정확한지 확인합니다. 다음 시나리오에는 데이터 공급자가 필요합니다.

- 디렉터리 서버 간 복제
- Directory Server와 Active Directory 간의 동기화
- Directory Server 데이터에 액세스하는 독립 클라이언트 애플리케이션

다중 제공 복제를 사용하면 Directory Server에 여러 서버의 기본 정보 사본을 포함할 수 있습니다. 여러 공

급업체가 변경 로그를 유지하고 충돌을 안전하게 해결합니다. 변경을 수락하고 데이터를 복제본 또는 소비자 서버에 복제할 수 있는 제한된 수의 공급업체 서버를 구성할 수 있습니다. [1]. 여러 데이터 공급자 서버는 서버가 오프라인 상태가 되면 안전한 페일오버를 제공합니다. 다중 제공 복제에 대한 자세한 내용은 TBA[복제 프로세스 설계]를 참조하십시오.

동기화를 사용하면 Directory Server 사용자, 그룹, 속성 및 암호를 Microsoft Active Directory 사용자, 그룹, 속성 및 암호와 통합할 수 있습니다. 두 개의 디렉터리 서비스가 있는 경우 동일한 정보를 관리할지 여부, 공유할 정보의 양 및 데이터를 제공할 서비스를 결정합니다. 데이터를 관리하고 동기화 프로세스에서 다른 서비스에서 항목을 추가, 업데이트 또는 삭제할 수 있도록 하나의 애플리케이션을 선택하는 것이 좋습니다.

디렉터리와 간접적으로 통신하는 애플리케이션을 사용하는 경우 데이터 공급자 소스를 고려하십시오. 데이터 변경 프로세스를 가능한 한 간단하게 유지합니다. 데이터 조각을 관리하기 위한 위치를 결정한 후 동일한 위치를 사용하여 여기에 포함된 다른 모든 데이터를 관리합니다. 단일 위치에서 데이터베이스의 동기화가 손실될 때 문제 해결을 단순화합니다.A single place simplifies troubleshooting when databases lose synchronization across the enterprise.

다음과 같은 방법으로 데이터 공급을 제공할 수 있습니다.

- 디렉터리를 사용하지 않는 디렉터리 및 모든 애플리케이션에서 데이터를 관리합니다.
  여러 데이터 공급업체를 유지 관리하는 데는 데이터 전송을 위한 사용자 지정 스크립트가 필요하지 않습니다. 이 경우 기업 전체에서 데이터 동기화를 방지하기 위해 다른 모든 사이트의 데이터를 변경해야 하지만 이는 디렉터리 목적에 반합니다.

- 비 디렉터리 애플리케이션에서 데이터를 관리하고 해당 데이터를 디렉터리로 가져오기 위해 스크립트, 프로그램 또는 게이트웨이를 작성합니다.
  비 디렉터리 애플리케이션에서 데이터를 관리하는 것은 이미 애플리케이션을 사용하여 데이터를 관리하는 경우 가장 적합합니다. 또한 이 디렉터리는 온라인 회사 전화 서적과 같은 조회에만 사용할 수 있습니다.

데이터의 주요 복사본을 유지 관리하는 방법은 특정 디렉터리의 필요에 따라 다릅니다. 그러나 항상 유지 관리를 간단하고 일관되게 유지합니다. 예를 들어 여러 위치에서 데이터를 관리한 다음 경쟁하는 애플리케이션 간에 데이터를 자동으로 교환하지 마십시오. 이렇게 하면 업데이트가 손실되고 관리 오버헤드가 증가합니다.

예를 들어 디렉터리는 LDAP 디렉터리와 인사 리소스 데이터베이스 모두에 저장된 직원 홈 전화 번호를 관리합니다. 인적 리소스 애플리케이션은 LDAP를 사용할 수 있으며 LDAP 디렉토리에서 인사 데이터베이스로 데이터를 자동으로 전송할 수 있으며 그 반대의 경우도 마찬가지입니다.

LDAP 디렉토리 및 인적 리소스 데이터베이스 모두에서 해당 직원 전화 번호 변경 사항을 관리하려고 하면 전화 번호가 변경된 마지막 위치가 다른 데이터베이스의 정보를 덮어씁니다. 이는 데이터를 작성한 마지막 애플리케이션에 올바른 정보가 있는 경우에만 허용됩니다.

해당 정보가 오래된 경우(예: 인적 리소스 데이터가 백업에서 복원되었기 때문에) LDAP 디렉터리의 올바른 전화 번호가 삭제됩니다.

2.3.6. 데이터 소유권 확인

데이터 소유권 은 데이터가 최신 상태인지 확인하는 사람 또는 조직을 나타냅니다. 데이터 설계 단계에서 디렉터리에 데이터를 쓸 수 있는 사람을 결정합니다. 다음은 데이터 소유권을 결정하기 위한 몇 가지 일반적인 전략입니다.

- 작은 디렉토리 콘텐츠 관리자 그룹을 제외한 모든 사용자에 대해 디렉토리에 대한 읽기 전용 액세스를 허용합니다.

- 개별 사용자가 자신의 암호, 조직 내의 역할, 자신의 라이센스 번호, 전화 번호 또는 사무실 번호와 같은 연락처 정보와 같은 정보의 전략적 하위 집합을 관리할 수 있습니다.
- 개인 관리자가 연락처 정보 또는 직무 제목과 같은 해당 개인 정보의 전략적 하위 집합을 작성할 수 있습니다.
- 조직 관리자가 해당 조직의 항목을 생성하고 관리하도록 허용하여 디렉터리 콘텐츠 관리자 역할을 할 수 있습니다.
- 사용자 그룹이 액세스 권한을 읽거나 쓸 수 있도록 하는 역할을 생성합니다. 인적 리소스, 재무 또는 회계에 대한 역할을 생성할 수 있습니다. 이러한 각 역할에 대해 그룹에 필요한 데이터에 대한 읽기 액세스, 쓰기 액세스 또는 둘 다 가질 수 있도록 허용합니다. 여기에는 급여 정보, 정부 식별 번호 및 집 전화번호와 주소가 포함될 수 있습니다.

여러 개인에서 동일한 정보에 대한 쓰기 권한이 필요할 수 있습니다. 예를 들어 정보 시스템 또는 디렉터리 관리 그룹에는 직원 암호에 대한 쓰기 권한이 필요할 수 있습니다. 또한 직원에게는 자신의 암호에 대한 쓰기 액세스 권한이 필요합니다. 여러 사용자가 동일한 정보에 액세스할 수 있지만 이 그룹을 작게 유지하여 데이터 무결성을 보장하십시오.

2.3.7. 데이터 액세스 확인

데이터 소유권을 확인한 후 각 데이터를 읽을 수 있는 사용자를 결정합니다. 예를 들어, 직원 홈 전화번호를 디렉터리에 저장할 수 있습니다. 이 데이터는 직원 관리자 및 인적 자원 부서를 포함한 여러 사용자에게 유용할 수 있습니다. 직원은 확인 목적으로 이 정보를 읽을 수 있어야 합니다. 그러나 집의 연락처 정보는 민감한 것으로 간주 될 수 있습니다.

디렉터리에 저장된 모든 정보에 대해 다음을 고려하십시오.

- 누군가가 익명의 데이터를 읽을 수 있습니까?
  LDAP 프로토콜은 익명 액세스를 지원하며 쉽게 정보를 조회할 수 있습니다. 그러나 누구나 디렉토리에 액세스할 수 있는 이러한 익명성으로 인해 이 기능을 신중하게 사용하십시오.
- 누군가가 기업 전체에서 데이터를 광범위하게 읽을 수 있습니까?
  특정 정보를 읽기 위해 클라이언트가 디렉토리에 로그인(또는 바인딩)해야 하는 방식으로 액세스 제어를 설정할 수 있습니다. 익명 액세스와 달리 이러한 유형의 액세스 제어를 통해 조직의 멤버만 디렉터리 정보에 액세스할 수 있습니다. 또한 Directory Server 액세스 로그에는 정보에 액세스한 사용자에 대한 레코드가 포함되어 있습니다.

  액세스 제어에 대한 자세한 내용은 액세스 제어 설계를 참조하십시오.
- 데이터에 액세스해야 하는 식별 가능한 사용자 또는 애플리케이션 그룹이 있습니까?
  데이터에 대한 쓰기 권한이 있는 모든 사용자에게 읽기 액세스 권한도 필요합니다(암호에 대한 쓰기 액세스 제외). 디렉터리에는 특정 조직 또는 프로젝트 그룹과 관련된 데이터를 포함할 수도 있습니다. 이러한 액세스 요구 사항을 식별하면 필요한 그룹, 역할 및 액세스 제어 권한을 결정하는 데 도움이 됩니다.

  그룹 및 역할에 대한 자세한 내용은 디렉터리 트리 설계를 참조하십시오. 액세스 제어에 대한 자세한 내용은 액세스 제어 설계를 참조하십시오.

디렉터리 데이터의 각 부분에 대해 이러한 결정을 내릴 때 디렉터리에 대한 보안 정책을 정의합니다. 이러한 결정은 사이트의 특성과 사이트에서 이미 사용 가능한 보안에 따라 달라집니다. 예를 들어 방화벽이 있거나 인터넷에 직접 액세스할 수 없으므로 디렉터리가 인터넷에 직접 배치된 것보다 익명 액세스를 지원하는 것이 더 안전합니다. 또한 일부 정보는 액세스를 적절하게 제한하기 위해 액세스 제어 및 인증 조치만 필요할 수 있습니다. 다른 민감한 정보는 저장 시 데이터베이스 내에서 암호화해야 할 수 있습니다.

대부분의 국가의 데이터 보호 법률은 기업이 개인 정보를 유지하고 액세스하는 방법을 관리합니다. 예를 들어, 법률에 따라 정보에 대한 익명 액세스를 금지하거나 사용자가 해당 정보를 나타내는 항목에서 정보

를 보고 편집할 수 있도록 요청할 수 있습니다. 조직의 법률 부서에 문의하여 디렉터리 배포가 기업이 운영하는 국가의 데이터 보호법을 준수하는지 확인하십시오.

보안 정책 생성 및 구현 방법은 보안 디렉토리 설계에 자세히 설명되어 있습니다.

복제에서 *소비자 서버* 또는 *복제본 서버*는 공급자 서버 또는 허브 서버에서 업데이트를 수신합니다.

2.4. 사이트 설문 조사 기록

데이터 설계의 복잡성으로 인해 사이트 조사 결과를 문서화합니다. 사이트 조사의 모든 단계에서 간단한 테이블을 사용하여 데이터를 추적할 수 있습니다. 의사 결정 및 뛰어난 우려 사항을 설명하는 공급자 테이블을 만들 수 있습니다. 콘텐츠를 쉽게 정렬하고 검색할 수 있는 다양한 기능을 사용하는 것이 좋습니다.

아래 표는 사이트 설문 조사에서 식별한 각 데이터에 대한 데이터 소유권 및 데이터 액세스를 식별합니다.

표 2.1. 예: 데이터 소유권 및 액세스 탭

데이터 이름	소유자	portal server/Application	자체 읽기/쓰기	글로벌 읽기	HR 요구 사항	It is Writable
직원 이름	HR	PeopleSoft	읽기 전용	제공됨 (익명)	제공됨	제공됨
사용자 암호	IS	디렉터리 US-1	읽기/쓰기	없음	없음	제공됨
홈 전화번호	HR	PeopleSoft	읽기/쓰기	없음	예	없음
직원 위치	IS	디렉터리 US-1	읽기 전용	제공됨(로그인해야 함)	없음	제공됨
사무실 전화 번호	기능	전화 스위치	읽기 전용	제공됨 (익명)	없음	없음

표의 각 행은 평가되는 정보의 유형, 관심 있는 부서, 정보를 사용하고 액세스하는 방법을 나타냅니다. 예를 들어 첫 번째 행에서 *직원 이름* 데이터에는 다음과 같은 관리 고려 사항이 있습니다.

- *소유자*. 인적 리소스는 이러한 정보를 소유하므로 업데이트 및 변경에 대한 책임이 있습니다.
- *공급업체 서버/애플리케이션*. PeopleSoft 애플리케이션은 직원 이름 정보를 관리합니다.
- *자체 읽기/쓰기*. 자신의 이름을 읽을 수는 있지만 쓰기 또는 변경할 수 없습니다.
- *글로벌 읽기*. 직원 이름은 디렉터리에 액세스할 수 있는 모든 사람이 무관하게 읽을 수 있습니다.
- *인사할 수 있는*. 인적 리소스 그룹 멤버는 디렉터리에서 직원 이름을 변경, 추가 및 삭제할 수 있습니다.
- *쓸 수 있습니다*. 정보 서비스(IS) 그룹 멤버는 디렉터리에서 직원 이름을 변경, 추가 및 삭제할 수 있습니다.

2.5. 사이트 설문 조사 반복

특히 기업이 여러 도시 또는 국가에 사무실이 있는 경우 두 개 이상의 사이트 설문 조사가 필요할 수 있습니다. 정보 요구 사항은 매우 복잡하여 여러 조직이 하나의 중앙 집중식 사이트 대신 현지 사무실에 정보를 보관해야 할 수 있습니다.

이 경우 주요 정보 사본을 보관하는 각 사무실은 자체 사이트 설문 조사를 수행해야 합니다. 사이트 설문 조사가 완료되면 엔터프라이즈 전체 데이터 스키마 모델 및 디렉터리 트리의 설계에 사용할 수 있도록 각 설문 조사의 결과를 중앙 팀 (각 사무실의 대표로 구성)으로 반환해야합니다.

---

[1] 복제에서 *소비자 서버* 또는 *복제본 서버*는 공급자 서버 또는 허브 서버에서 업데이트를 수신합니다.

3장. 디렉터리 스키마 설계

디렉터리 스키마는 디렉터리의 데이터 유형을 설명합니다. 디렉터리에 저장된 데이터의 표현을 알고 있을 때 사용할 스키마를 결정할 수 있습니다. 각 데이터 요소는 LDAP 특성에 매핑되며, 관련 요소는 스키마 설계 프로세스 중에 LDAP 개체 클래스로 수집됩니다. 잘 설계된 스키마는 디렉터리 데이터의 무결성을 유지하는 데 도움이 됩니다.

3.1. 스키마 설계 프로세스 개요

개체 클래스 및 특성을 선택하고 정의하여 스키마 설계 프로세스 중에 Directory Server에 의해 저장된 항목을 나타낼 수 있습니다. 스키마 설계 프로세스 중에 다음 단계가 수행됩니다.

- 데이터 요구 사항을 충족하기 위해 사전 정의된 스키마 요소를 선택합니다.
- 표준 Directory Server 스키마를 확장하여 요구 사항을 충족하는 새 요소를 정의합니다.
- 스키마 유지 관리 계획.

Directory Server에서 제공하는 표준 스키마에 정의된 기존 스키마 요소를 사용할 수 있습니다. 표준 스키마 요소는 디렉터리 지원 애플리케이션과의 호환성을 확인하는 데 도움이 됩니다. 스키마는 LDAP 표준을 기반으로 하므로 스키마가 다양한 디렉터리 사용자가 검토하고 동의합니다.

3.2. 표준 스키마

디렉터리 스키마는 데이터 값의 크기, 범위 및 형식에 제약 조건을 설정하여 디렉터리에 저장된 데이터의 무결성을 유지 관리합니다. 스키마는 디렉터리에 포함된 다양한 유형의 항목(예: 사람, 장치, 조직)과 각 항목에 사용할 수 있는 속성을 식별합니다.

Directory Server의 사전 정의된 스키마에는 서버의 기능을 지원하는 표준 LDAP 스키마 및 애플리케이션별 스키마가 모두 포함되어 있습니다. 디렉터리의 고유한 요구 사항을 수용하도록 새 오브젝트 클래스 및 속성을 추가하여 스키마를 확장할 수 있습니다.

3.2.1. 스키마 형식

Directory Server의 스키마 형식은 LDAP 프로토콜의 버전 3을 기반으로 합니다. 이 프로토콜을 사용하려면 디렉터리 클라이언트 애플리케이션이 프로그래밍 방식으로 스키마를 검색하고 동작을 조정할 수 있도록 LDAP를 통해 스키마를 게시해야 합니다. **cn=schema** 항목에서 Directory Server의 글로벌 스키마 세트를 찾을 수 있습니다.

디렉터리 서버 스키마는 전용 개체 클래스 및 특성을 사용하므로 LDAPv3 스키마와 다릅니다. 또한 스키마 항목이 원래 정의된 위치를 설명하는 **X-ORIGIN 389 Directory Server** 라는 스키마 항목의 개인 필드를 사용합니다.

표준 LDAPv3 스키마에 스키마 항목을 정의할 때 **X-ORIGIN 389 Directory Server** 필드는 RFC 2252를 나타냅니다. Directory Server 사용을 위해 Red Hat에서 항목을 정의한 경우 **X-ORIGIN 389 Directory Server** 필드에 값 **389 Directory Server** 가 포함됩니다. 예를 들어 표준 person 오브젝트 클래스는 스키마에 표시됩니다.

```
# objectclasses: ( 2.5.6.6 NAME 'person' DESC 'Standard Person Object Class' SUP top
MUST (objectclass $ sn $ cn) MAY (description $ seeAlso $ telephoneNumber $ userPassword)
X-ORIGIN 'RFC 2252' )
```

이 스키마 항목 상태는 다음과 같습니다.

- 클래스의 개체 식별자(OID)입니다(**2.5.6.6**)
- 오브젝트 클래스의 이름(**person**)
- 클래스에 대한 설명 (**표준 사람**)
- 필수 속성(**objectclass, sn 및 cn**)
- 선택적 속성(**설명,전화 번호** 및 **userPassword**)을**참조하십시오**.

3.2.2. 표준 속성

특성에는 이름 또는 고정 번호와 같은 특정 데이터 요소가 포함됩니다. Directory Server는 데이터를 특정 정보와 관련된 설명적인 스키마 속성인 attribute-data 쌍으로 나타냅니다. 이를 **attribute-value assertions** 또는 **AVA 라고도 합니다**.

예를 들어 디렉터리는 사용자 이름과 같은 데이터를 표준 속성이 있는 쌍에 저장할 수 있습니다. **Babs Jensen** 이라는 이름의 항목에는 attribute-data 쌍 **cn: Babs Jensen** 이 있습니다.

전체 항목은 일련의 특성 데이터 쌍으로 표시됩니다. Babs Jensen의 전체 항목은 다음과 같습니다.

```
dn: uid=bjensen,ou=people,dc=example,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Babs Jensen
sn: Jensen
givenName: Babs
givenName: Barbara
mail: bjensen@example.com
```

스키마의 각 속성 정의에는 다음 정보가 포함되어 있습니다.

- 고유한 이름
- 속성의 OID(오브젝트 식별자)
- 특성에 대한 설명A text description of the attribute
- 특성 구문의 OID
- 다음에 대한 표시:
  a. 속성은 단일 값 또는 다중 값입니다.
  b. 이 속성은 디렉터리의 자체 용도입니다.
  c. 속성의 원본입니다.
  d. 속성과 연결된 추가 일치 규칙입니다.

cn 속성 정의는 다음과 같이 스키마에 표시됩니다.

```
attributetypes: ( 2.5.4.3 NAME 'cn' DESC 'commonName Standard Attribute'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 )
```

특성의 구문을 사용하여 속성에 저장할 수 있는 값의 형식을 정의할 수 있습니다. Directory Server는 모든 표준 특성 구문을 지원합니다.

추가 리소스

- 지원되는 LDAP 속성 구문

3.2.3. 표준 오브젝트 클래스

개체 클래스는 사람 또는 성정기 시스템과 같은 실제 개체를 나타냅니다. 관련 정보를 그룹화하는 데 사용됩니다. 오브젝트 클래스를 사용하기 전에 스키마에서 오브젝트 클래스 및 해당 속성을 식별해야 합니다. 디렉터리는 기본적으로 표준 오브젝트 클래스 목록을 인식합니다.

각 디렉터리 항목은 하나 이상의 오브젝트 클래스에 속합니다. 항목의 스키마에 식별된 오브젝트 클래스를 배치하면 해당 항목에 특정 특성 값 세트가 있을 수 있으며 다른 더 작은 필수 특성 값 세트가 있어야 함을 디렉터리 서버에 알립니다.

오브젝트 클래스 정의에서 다음 정보를 사용할 수 있습니다.

- 고유한 이름
- OID(오브젝트 식별자)
- 필수 속성 세트
- 허용 또는 선택적 속성 세트

스키마의 표준 person 개체 클래스입니다.

```
objectclasses: ( 2.5.6.6 NAME 'person' DESC 'Standard Person Object Class' SUP top
    MUST (objectclass $ sn $ cn) MAY (description $ seeAlso $ telephoneNumber $ userPassword)
    X-ORIGIN 'RFC 2252' )
```

참고

오브젝트 클래스가 정의되고 Directory Server에 직접 저장되므로 표준 LDAP 작업으로 디렉터리 스키마를 쿼리하고 변경할 수 있습니다.

추가 리소스

- 오브젝트 클래스의 표준 목록

3.3. 데이터를 기본 스키마에 매핑

사이트 설문 조사 중에 식별된 데이터를 기존 기본 디렉터리 스키마에 매핑해야 합니다. 스키마의 요소가 기존 기본 스키마와 일치하지 않으면 사용자 지정 개체 클래스 및 특성을 만들 수 있습니다.

기본 디렉터리 스키마는 Directory Server의 공통 스키마를 모두 포함하는 **/usr/share/dirsrv/schema/** 디렉터리에 저장됩니다. LDAPv3 표준 사용자 및 조직 스키마는 **00core.ldif** 파일에서 찾을 수 있습니다. 이전 버전의 디렉터리에서 사용하는 구성 스키마는 **50ns-directory.ldif** 파일에서 찾을 수도 있습니다.

주의

기본 디렉터리 스키마를 수정하지 마십시오.

3.3.1. 스키마 요소와 일치하는 데이터

사이트 설문 조사에서 식별된 데이터를 기존 디렉터리 스키마에 매핑할 수 있습니다. 이 프로세스에는 다음 단계가 포함됩니다.

- 데이터가 설명하는 오브젝트 유형을 식별해야 합니다.

경우에 따라 데이터 조각이 여러 오브젝트를 설명할 수 있습니다. 디렉터리 스키마에 차이점을 기록해야 하는지 확인합니다. 예를 들어, 전화 번호는 직원의 전화 번호와 회의실의 전화 번호를 설명할 수 있습니다. 이러한 종류의 데이터가 디렉터리 스키마의 다른 개체로 간주되어야 하는지 여부를 결정합니다.

- 기본 스키마에서 유사한 오브젝트 클래스를 선택해야 합니다. 그룹, 사람 및 조직과 같은 공통 오브젝트 클래스를 사용하는 것이 가장 좋습니다.
- 일치하는 오브젝트 클래스에서 유사한 속성을 선택해야 합니다.
- 사이트 설문 조사에서 일치하지 않는 데이터를 식별해야 합니다. 기본 디렉터리 스키마에서 정의한 오브젝트 클래스 및 특성과 일치하지 않는 데이터 일부가 스키마를 사용자 지정합니다.

참고

Directory Server 구성, 명령 및 파일 참조는 데이터에 사용할 수 있는 속성을 결정하는 데 유용합니다. 각 속성은 이를 허용하는 오브젝트 클래스와 함께 나열되며 각 오브젝트 클래스는 필수 및 허용되는 특성에 따라 교차 목록에 추가됩니다.

추가 리소스

- Red Hat Directory Server 구성, 명령 및 파일 참조

3.4. 스키마 사용자 정의

특성 및 개체 클래스를 추가하여 Directory Server의 웹 콘솔을 사용하여 표준 스키마를 확장할 수 있습니다. LDIF 파일을 생성하고 수동으로 스키마 요소를 추가할 수도 있습니다.

스키마를 사용자 지정하는 동안 다음 규칙을 적용할 수 있습니다.

- 스키마를 간단하게 유지해야 합니다.
- 스키마 요소를 다시 사용해야 합니다.
- 각 오브젝트 클래스에 대해 정의된 필수 특성 수를 최소화해야 합니다.
- 동일한 목적(데이터)에 대해 둘 이상의 오브젝트 클래스 또는 속성을 정의하지 마십시오.
- 특성 또는 개체 클래스의 기존 정의를 수정하지 마십시오.

참고

스키마를 사용자 정의할 때 표준 스키마를 삭제하거나 교체할 수 없습니다. 이렇게 하면 다른 디렉터리 또는 LDAP 클라이언트 애플리케이션과의 호환성 문제가 발생할 수 있습니다.

사용자 지정 오브젝트 클래스 및 속성은 **99user.ldif** 파일에 정의됩니다. 각 인스턴스는 **/etc/dirsrv/slapd-*instance_name*/schema/** 디렉터리에 자체 **99user.ldif** 파일을 유지 관리합니다. 사용자 지정 스키마 파일을 생성하고 서버에 스키마를 동적으로 다시 로드할 수도 있습니다.

지정된 개체 클래스가 조직에 대한 특수 정보를 저장할 수 없는 경우 스키마를 확장할 수 있지만 Directory Server와 함께 제공되는 오브젝트 클래스와 특성은 가장 일반적인 기업 요구 사항을 충족해야 합니다. 스키마를 확장하여 LDAP 지원 애플리케이션의 고유한 데이터 요구에 필요한 오브젝트 클래스 및 속성을 지원할 수도 있습니다.

3.4.1. 오브젝트 식별자 할당

각 LDAP **오브젝트** 클래스 또는 속성에 대해 고유 이름 및 OID(오브젝트 ID)를 할당해야 합니다. 스키마를 정의할 때 요소에는 조직에 고유한 기본 OID가 필요합니다. 다른 계층 구조를 추가하여 특성 및 개체 클래스에 대한 새 분기를 만듭니다. 스키마에서 OID를 가져오고 할당하려면 다음 단계를 수행해야 합니다.

1. INA(**Internet Assigned Numbers Authority**) 또는 국가 조직에서 OID를 받습니다. 일부 국가에는 이미 OID가 할당되어 있습니다.

2. OID 레지스트리를 생성하여 OID 할당을 추적합니다. OID 레지스트리는 OID 목록 및 디렉터리 스키마에 사용되는 OID에 대한 설명입니다. 이렇게 하면 OID가 두 개 이상의 용도로 사용되지 않습니다. 그런 다음 OID 레지스트리를 스키마와 함께 게시합니다.

3. OID 트리에 분기를 생성하여 스키마 요소를 수용합니다. OID 분기 또는 디렉터리 스키마 아래에 두 개 이상의 분기를 생성합니다. 속성의 경우 OID.**1** 및 개체 클래스의 경우 OID. **2** 를 사용합니다. 필요에 따라 새 분기를 추가하여 사용자 정의 일치 규칙 또는 제어(예: OID.'3)를 정의합니다.

추가 리소스

- IANA

3.4.2. 새 오브젝트 클래스를 정의하는 전략

다음과 같은 두 가지 방법으로 새 오브젝트 클래스를 생성할 수 있습니다.

- 특성을 추가할 수 있는 각 개체 클래스 구조에 대해 하나씩 새 개체 클래스를 만듭니다.

- 디렉터리에 대해 생성된 모든 사용자 지정 특성을 지원하는 단일 오브젝트 클래스를 만듭니다. 이 오브젝트 클래스를 보조 오브젝트 클래스로 정의하여 생성할 수 있습니다.

두 가지 방법을 혼합할 수 있습니다. 예를 들어 예제 **DateOfBirth ,example PreferredOS**,**exampleBuildingFloor** 및 **exampleVicePresident** 속성을 생성하려고 합니다. 간단한 솔루션은 이러한 속성의 일부 하위 집합을 허용하는 여러 개체 클래스를 만드는 것입니다.

- **examplePerson** 오브젝트 클래스는 **exampleDateOfBirth** 및 **examplePreferredOS** 를 허용합니다. **examplePerson** 의 상위는 **inetOrgPerson** 입니다.

- **exampleOrganization** 오브젝트 클래스를 사용하면 **exampleBuildingFloor** 및 **exampleVicePresident**. exampleOrganization의 상위는 **조직** 오브젝트 클래스입니다.

새 오브젝트 클래스는 다음과 같이 LDAPv3 스키마 형식으로 표시됩니다.

■

```
objectclasses: ( 2.16.840.1.117370.999.1.2.3 NAME 'examplePerson' DESC 'Example Person
Object Class'
    SUP inetorgPerson MAY (exampleDateOfBirth $ examplePreferredOS) )

objectclasses: ( 2.16.840.1.117370.999.1.2.4 NAME 'exampleOrganization' DESC 'Organization
Object Class'
    SUP organization MAY (exampleBuildingFloor $ exampleVicePresident) )
```

또는 이러한 모든 특성을 허용하는 단일 오브젝트 클래스를 만들고 이를 필요한 모든 항목과 함께 사용할 수 있습니다. 단일 오브젝트 클래스는 다음과 같습니다.

```
objectclasses: (2.16.840.1.117370.999.1.2.5 NAME 'exampleEntry' DESC 'Standard Entry Object
Class' SUP top
    AUXILIARY MAY (exampleDateOfBirth $ examplePreferredOS $ exampleBuildingFloor $
exampleVicePresident) )
```

새로운 **exampleEntry** 오브젝트 클래스는 A Cryostat **ILIARY** 로 표시됩니다. 즉, 구조적인 오브젝트 클래스에 관계없이 모든 항목에 사용할 수 있습니다.

조직 환경에 따라 새 오브젝트 클래스를 구성할 수 있습니다. 새 개체 클래스의 구현을 결정할 때는 다음을 고려하십시오.

- 스키마에 두 개 이상의 개체 클래스가 추가되는 경우 단일 개체 클래스를 사용해야 합니다.
- 여러 오브젝트 클래스에는 데이터 설계가 필요합니다. 데이터 설계는 모든 데이터가 유용하거나 번거로울 수 있는 오브젝트 클래스 구조에 중점을 둡니다.
- 데이터를 사람 및 자산 항목과 같은 둘 이상의 오브젝트 클래스에 적용할 수 있는 경우 단일 개체 클래스를 사용하여 데이터를 사용할 수 있습니다. 예를 들어 사용자와 그룹 항목 모두에 사용자 정의 **preferredOS** 특성을 설정할 수 있습니다. 단일 오브젝트 클래스는 두 유형의 항목에 대해 이 속성을 허용할 수 있습니다.
- 새 개체 클래스에 대한 필수 특성을 피해야 합니다. 새 개체 클래스의 속성을 **허용하는** 대신 **require** 를 지정하면 스키마를 유연하게 만들 수 있습니다. 새 개체 클래스를 정의한 후 허용 및 필요한 특성과 특성을 상속하는 오브젝트 클래스를 결정합니다.

3.4.3. 새 특성을 정의하는 전략

애플리케이션 호환성과 장기 유지 관리 모두에 표준 특성을 사용해야 합니다. 기본 디렉터리 스키마에 이미 존재하는 특성을 검색하고 새 개체 클래스와 함께 사용하거나 Directory Server 스키마 가이드를 확인해야 합니다. 그러나 표준 스키마에 필요한 모든 정보가 포함되어 있지 않은 경우 새 특성과 새 개체 클래스를 추가합니다.

예를 들어 사람 항목에는 기본적으로 사람, **organizationalPerson** 또는 **inetOrgPerson** 개체 클래스보다 많은 속성이 필요할 수 있습니다. 생년월일을 저장하기 위한 표준 Directory Server 스키마 내에는 특성이 없습니다. 새 특성을 생성하고 설정할 수 있습니다. **dateOfBirth** 는 새 보조 개체 클래스 내에서 허용되는 속성인 **examplePerson**:

```
attributetypes: ( dateofbirth-oid NAME 'dateofbirth' DESC 'For employee birthdays'
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 X-ORIGIN 'Example defined')

objectclasses: ( 2.16.840.1.117370.999.1.2.3 NAME 'examplePerson' DESC 'Example Person
Object Class'
    SUP inetorgPerson MAY (exampleDateOfBirth $ cn) X-ORIGIN 'Example defined')
```

참고

표준 스키마 요소에 사용자 지정 속성을 추가하거나 삭제할 수 없습니다. 디렉터리에 사용자 지정 특성이 필요한 경우 사용자 지정 개체 클래스를 추가하여 포함합니다.

3.4.4. 스키마 요소 삭제

Directory Server에 기본적으로 포함된 스키마 요소는 삭제할 수 없습니다. 사용되지 않는 스키마 요소는 운영 또는 관리 오버헤드를 나타내지 않습니다. 표준 LDAP 스키마의 일부를 삭제하면 나중에 Directory Server 및 기타 디렉터리 지원 애플리케이션의 호환성 문제가 발생할 수 있습니다.

그러나 사용되지 않는 사용자 지정 스키마 요소를 삭제할 수 있습니다. 스키마에서 오브젝트 클래스 정의를 제거하기 전에 오브젝트 클래스를 사용하여 각 항목을 수정합니다. 먼저 정의를 제거하면 오브젝트 클래스를 사용하는 항목이 나중에 수정되지 않을 수 있습니다. 알 수 없는 개체 클래스 값이 항목에서 제거되지 않으면 스키마가 수정된 항목도 실패합니다.

3.4.5. 사용자 정의 스키마 파일 생성

Directory Server에 대한 사용자 지정 스키마 파일을 생성하여 Directory Server와 함께 제공된 **99user.ldif** 파일 외에도 사용할 수 있습니다. 이러한 스키마 파일에는 조직과 관련된 새로운 사용자 지정 속성 및 개체 클래스가 있습니다. 새 스키마 파일은 스키마 디렉터리 **/etc/dirsrv/slapd-instance_name/schema/** 에 있습니다. 모든 표준 특성 및 개체 클래스는 사용자 지정 스키마 요소가 로드된 후에만 로드됩니다.

참고

사용자 지정 스키마 파일은 **99user.ldif** 보다 숫자 또는 알파벳으로 높을 수 없습니다.

사용자 지정 스키마 파일을 만든 후 다음과 같은 방식으로 모든 서버에 스키마 변경 사항을 배포할 수 있습니다.

- 이러한 사용자 지정 스키마 파일을 인스턴스의 스키마 디렉터리 **/etc/dirsrv/slapd-instance/schema** 에 수동으로 복사하고 스키마를 로드하거나, 서버를 다시 시작하거나, **schema-reload.pl** 스크립트를 실행하여 동적으로 스키마를 다시 로드할 수 있습니다.
- 웹 콘솔과 같은 LDAP 클라이언트를 사용하거나 **ldapmodify** 명령을 사용하여 서버의 스키마를 수정할 수 있습니다.
- 복제를 사용하면 복제된 모든 스키마 요소가 소비자 서버 **99user.ldif** 파일에 복사됩니다. **90example_schema.ldif** 와 같은 사용자 지정 스키마 파일에 스키마를 유지하려면 파일을 소비자 서버로 수동으로 복사해야 합니다. 복제는 스키마 파일을 복사하지 않습니다.

이러한 사용자 지정 스키마 파일을 모든 서버에 복사하지 않으면 공급자 서버의 스키마를 변경하는 경우에만 스키마 정보가 소비자 서버에 복제됩니다. 스키마 정의가 아직 없는 소비자 서버에 복제되면 **99user.ldif** 파일에 저장됩니다.

참고

디렉터리는 스키마 정의가 저장되는 위치를 추적하지 않습니다. 스키마가 공급자 서버에서만 유지되는 경우 소비자의 **99user.ldif** 파일에 스키마 요소를 저장할 수 있습니다.

3.4.6. 사용자 정의 스키마 모범 사례

다음 제안 사항은 호환 가능하고 관리하기 쉬운 사용자 정의 스키마를 정의하는 데 도움이 됩니다.

스키마 파일 이름 지정

사용자 지정 스키마 파일의 이름을 **99user.ldif** 보다 숫자 및 알파벳순으로 낮게 지정합니다. **99user.ldif 파일에** 는 'user defined'의 **X-ORIGIN** 값이 있는 속성이 포함되어 있습니다. Directory Server는 모든 '사용자 정의' 스키마 요소를 가장 높은 이름이 지정된 파일에 쓰고 숫자 및 알파벳순으로 작성합니다. 스키마 파일 이름이 **99zzz.ldif** 이고 스키마가 업데이트되면 'user defined'의 **X-ORIGIN** 값이 있는 모든 속성이 **99zzz.ldif** 파일에 기록됩니다. 결과적으로 중복 정보가 포함된 LDIF 파일 두 개 모두 삭제되고 **99zzz.ldif 파일** 의 일부 정보는 삭제될 수 있습니다.

사용자 지정 스키마 파일 이름을 지정할 때 다음 이름 지정 형식을 사용합니다 **.yourName.ldif**.

'user defined'를 원본으로 사용

LDAP를 통해 스키마를 추가할 때 '사용자 정의' **가** 디렉터리 서버에서 내부적으로 사용되므로 사용자 지정 스키마 파일의 **X-ORIGIN** 필드에 'user defined'를 사용하지 마십시오.

사용자 지정 스키마 요소가 수동으로 **99user.ldif** 에 직접 추가되는 경우 **X-ORIGIN** 값으로 'user defined'를 사용합니다. 다른 **X-ORIGIN** 값이 설정된 경우 서버는 간단히 이를 덮어쓸 수 있습니다.

'사용자 정의' 값의 **X-ORIGIN** 을 사용하면 Directory Server에서 **99user.ldif** 파일에서 스키마 정의를 제거할 수 없습니다.

예를 들면 다음과 같습니다.

```
attributetypes: ( exampleContact-oid NAME 'exampleContact'
DESC 'Example Corporate contact'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
X-ORIGIN 'Example defined')
```

Directory Server가 스키마 항목을 로드하면 다음과 같이 표시됩니다.

```
attributetypes: ( exampleContact-oid NAME 'exampleContact'
DESC 'Example Corporate contact'
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
X-ORIGIN ('Example defined' 'user defined') )
```

오브젝트 클래스 이전의 속성 정의

새 스키마 요소를 추가하면 개체 클래스에서 사용하기 전에 모든 특성을 정의합니다. 특성 및 개체 클래스는 동일한 스키마 파일에 정의할 수 있습니다.

단일 파일에서 스키마 정의

가장 최근에 생성된 스키마를 로드할 때 서버가 이전 정의를 덮어쓰지 않도록 각 사용자 지정 특성 또는 개체 클래스를 스키마 파일에 정의합니다. 서버는 먼저 스키마를 숫자순으로 로드한 다음 알파벳순으로 로드합니다. 중복 파일에 스키마를 사용하지 않도록 하는 방법을 결정합니다.

- 각 스키마 파일에 포함된 스키마 요소에 주의하십시오.
- 스키마 파일의 이름 지정 및 업데이트할 때는 주의하십시오. LDAP 도구를 통해 스키마 요소를 편집하면 변경 사항이 자동으로 마지막 파일에 알파벳순으로 작성됩니다. 대부분의 스키마 변경 사항은 기본 **99user.ldif** 파일에 기록되며 **60example.ldif** 와 같은 사용자 지정 스키마 파일에는 기록되지 않습니다. **99user.ldif** 파일의 스키마 요소는 다른 스키마 파일의 중복 요소를 재정의합니다.
- 웹 콘솔을 사용하여 스키마를 관리하는 경우 **99user.ldif** 파일에 모든 스키마 정의를 추가합니다.

3.5. 일관된 스키마 개요

LDAP 클라이언트 애플리케이션은 Directory Server에서 일관된 스키마를 사용하여 디렉터리 항목을 찾습니다. 동일한 정보를 저장하는 데 다른 속성 또는 형식을 사용하므로 일관되지 않은 스키마를 사용하여 디렉터리 트리에서 정보를 찾을 수 없습니다.

다음과 같은 방법으로 스키마 일관성을 유지할 수 있습니다.

- 스키마 검사를 사용하여 속성 및 개체 클래스가 스키마 규칙을 확인하는지 확인할 수 있습니다.
- 구문 검증을 사용하여 속성 값이 필수 특성 구문과 일치하는지 확인할 수 있습니다.
- 일관된 데이터 형식을 선택하고 적용할 수 있습니다.

3.5.1. 스키마 검사

스키마 검사는 모든 신규 또는 수정된 디렉터리 항목이 스키마 규칙을 준수하는지 확인합니다. 기본적으로 디렉터리는 스키마 검사를 활성화합니다. 규칙이 위반되면 디렉터리는 요청된 변경을 거부합니다.

참고

스키마 검사는 적절한 속성이 있는지 확인합니다. 구문 검증을 사용하여 속성 값이 올바른 구문에 있는지 확인할 수 있습니다. 이 기능을 비활성화하지 마십시오.

스키마 확인이 활성화된 경우 오브젝트 클래스에 정의된 필수 및 허용되는 속성에 주의해야 합니다. 항목의 개체 클래스 정의에 따라 필요하지 않거나 허용되지 않은 항목에 특성을 추가할 때 Directory Server에서 개체 클래스 위반 메시지를 반환할 수 있습니다.

예를 들어 **organizationalPerson** 오브젝트 클래스를 사용하도록 항목이 정의된 경우 항목에 공통 이름(**cn**) 및 surname(**sn**) 속성이 필요합니다. 이러한 속성의 값은 항목을 만들 때 설정해야 합니다. 또한 **전화번호**,**uid**,**StreetAddress** 및 **userPassword** 와 같은 설명적 특성을 포함하여 항목에서 선택적으로 사용할 수 있는 속성의 긴 목록이 있습니다.

3.5.2. 구문 검증 개요

구문 검증은 Directory Server에서 특성 값이 해당 속성에 필요한 구문과 일치하는지 확인합니다. 예를 들어 구문 검증에서는 새 **telephoneNumber** 속성에 해당 값에 유효한 전화 번호가 있는지 확인할 수 있습니다. 기본적으로 활성화되어 있습니다.

선택적으로 구문 위반에 대한 경고 메시지를 기록하도록 구문 검증에 대한 추가 설정을 구성한 다음 수정을 거부하거나 수정 프로세스가 성공하도록 허용할 수 있습니다.

새 속성 값이 추가되면 구문 검증에서 LDAP 작업을 확인합니다. 복제와 같은 데이터베이스 작업을 통해 추가된 기존 특성 또는 특성을 처리하지 않습니다. 기존 속성은 **dsconf** schema validate-syntax 명령을 사용하여 검증할 수 있습니다.

이 기능은 필수 형식이 없는 바이너리 구문 및 비표준 구문을 제외한 모든 특성 구문의 유효성을 검사합니다. 구문은 덜 엄격한 **RFC 1779** 또는 **RFC 2253** 에 대해 검증되는 DN을 제외하고 **RFC 4514** 에 대해 검증됩니다.

참고

엄격한 DN 검증을 구성할 수 있습니다.

3.5.2.1. 디렉터리 서버 작업에 대한 구문 검증

구문 검증은 항목(add) 또는 편집 속성(modify)과 같은 표준 LDAP 작업에 적용할 수 있습니다. 특성 구문을 검증하면 다른 Directory Server 작업에 영향을 미칠 수 있습니다.

데이터베이스 암호화

LDAP 작업을 위해 값을 데이터베이스에 쓰기 전에 속성을 암호화할 수 있습니다. 즉, 속성 구문의 유효성을 검사한 후 암호화가 수행됩니다. 암호화된 데이터베이스를 가져오고 내보낼 수 있습니다.

참고

가져오기 작업에 구문 유효성 검사를 수행할 수 있는 **--encrypted(dsctl)** 플래그를 사용하여 내보내기 및 가져오기 작업을 수행해야 합니다.

--encrypted 플래그(지원되지 않음)를 사용하지 않고 암호화된 데이터베이스를 내보내는 경우 암호화된 값이 있는 LDIF가 생성됩니다. 암호화된 특성, 경고가 기록되며 이 LDIF를 가져올 때 가져온 항목에서 속성 유효성 검사를 건너뛸 수 없습니다.

동기화

Windows Active Directory 항목 및 Directory Server 항목에서 특성에 대해 허용된 구문 또는 강제 구문에 차이가 있을 수 있습니다. 구문 검증이 Directory Server 항목에서 RFC 표준을 적용하므로 Active Directory 값을 동기화할 수 없습니다.

복제

Directory Server 11.0 인스턴스가 소비자에 변경 사항을 복제하는 공급자인 경우 구문 검증을 사용할 수 있습니다. 그러나 복제의 공급자가 이전 버전의 Directory Server이거나 구문 검증이 비활성화되었다고 가정합니다. 이 경우 Directory Server 11.0 소비자는 공급자가 허용하는 특성 값을 거부할 수 있기 때문에 11.0 소비자에서 구문 유효성 검사를 사용할 수 없습니다.

3.5.3. 일관된 데이터 형식

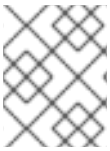
LDAP 스키마를 사용하여 속성 값으로 데이터를 배치할 수 있습니다. 그러나 LDAP 클라이언트 애플리케이션 및 디렉터리 사용자에게 적합한 형식을 선택하여 디렉터리 트리에 일관되게 데이터를 저장하는 것이 중요합니다.

LDAP 프로토콜 및 디렉터리 서버를 사용하여 **RFC 2252** 에 지정된 데이터 형식으로 데이터를 나타낼 수 있습니다. 예를 들어 전화 번호에 대한 올바른 LDAP 형식은 다음 두 **ITU-T** 권장 사항 문서에 정의되어 있습니다.

- **ITU-T 권장 사항 E.123**. 국가 및 국제 전화 번호에 대한 표기법입니다.
- **ITU-T 권장 사항 E.163**. 국제 전화 서비스에 대한 번호 지정 계획 예를 들어, 미국 전화번호는 **+1 555 222 1717** 로 포맷되어 있습니다.

또 다른 예로, **postalAddress** 속성에는 달러 기호($)를 줄 구분 기호로 사용하는 다중 줄 문자열 형식의 속성 값이 있습니다. 올바르게 포맷된 디렉터리 항목이 다음과 같이 표시됩니다.

```
postalAddress: 1206 Directory Drive$Pleasant View, MN$34200
```



참고

속성에는 문자열, 바이너리 입력, 정수 및 기타 형식이 필요합니다. 특성의 스키마 정의에서 형식을 설정할 수 있습니다.

3.5.4. 복제된 스키마에서 일관성 유지 관리 정보

변경 사항은 디렉터리 스키마를 편집할 때 변경 로그에 기록됩니다. 복제 중에 변경 사항이 복제되고 변경 사항이 복제되는 경우 변경 로그를 스캔합니다. 복제 스키마에서 일관성을 유지 관리하면 오류 없이 복제를 계속할 수 있습니다.

복제된 환경에서 일관된 스키마를 유지 관리하려면 다음 사항을 고려하십시오.

- 읽기 전용 복제본에서 스키마를 수정하지 마십시오.
  읽기 전용 복제본에서 스키마를 수정하면 스키마의 불일치가 발생하고 복제가 실패합니다.

- 다른 구문을 사용하는 것과 동일한 이름의 두 속성을 생성하지 마십시오.
  공급자 복제본의 속성과 동일한 이름으로 읽기-쓰기 복제본에 속성을 생성할 때 공급자의 속성과 다른 구문이 있는 경우 복제가 실패합니다.

3.6. 추가 리소스

- RFC 2251: Lightweight Directory Access Protocol (v3)
- RFC 2252: LDAPv3 속성 구문 정의
- RFC 2256: LDAPv3에 사용할 X.500 사용자 스키마 요약

4장. 디렉터리 트리 설계

디렉터리 트리를 사용하여 Directory Server로 저장된 데이터를 볼 수 있습니다. 디렉터리 트리의 설계는 디렉터리에 저장된 정보 유형, 엔터프라이즈의 물리적 특성, 디렉터리와 함께 사용되는 애플리케이션, 구현된 복제 유형을 기반으로 합니다.

4.1. 디렉터리 트리 소개

디렉터리 데이터의 이름을 지정하고 디렉터리 트리를 사용하여 클라이언트 애플리케이션을 참조할 수 있습니다. 디렉터리 트리는 디렉터리 데이터에 대한 액세스를 배포, 복제 또는 제어하는 데 사용할 수 있는 옵션을 포함하여 다른 설계 결정과 상호 작용할 수 있습니다. 디렉터리 서비스가 작동하는 경우 배포 중 및 나중에 배포 단계에서 시간과 노력을 절약할 수 있는 배포 전에 디렉터리 트리를 설계할 수 있습니다.

잘 설계된 디렉토리 트리를 사용하면 다음을 수행할 수 있습니다.

- 디렉터리 데이터를 유지하기만 하면 됩니다.
- 복제 정책 및 액세스 제어를 유연하게 생성합니다.
- 디렉터리 서비스를 사용하여 애플리케이션을 지원합니다.
- 사용자를 위한 디렉터리 탐색 간소화.

디렉터리 트리의 구조는 계층적 LDAP 모델을 따릅니다. 디렉터리 트리는 그룹, 인력 또는 위치별 등 다양한 논리 방식으로 데이터를 구성하는 방법을 제공합니다. 디렉터리 트리를 사용하여 여러 서버에서 데이터를 분할하는 방법도 확인할 수 있습니다. 예를 들어 각 데이터베이스는 접미사 수준에서 데이터를 분할해야 합니다. 적절한 디렉터리 트리 구조 없이는 데이터를 여러 서버에 효율적으로 분배할 수 없습니다.

또한 복제는 사용되는 디렉터리 트리 구조 유형으로 제한됩니다. 디렉터리 트리의 일부만 복제하려면 설계 프로세스 중에 이를 고려합니다.

4.2. 디렉터리 트리 설계

디렉터리 트리를 계획할 때 다음과 같은 주요 결정을 내립니다.

- 데이터를 포함할 접미사를 선택합니다.
- 디렉터리 트리 구조를 생성하여 데이터 항목 간 계층 관계를 결정합니다.
- 디렉터리 트리 계층 구조의 항목의 이름을 지정합니다.

4.2.1. 접미사 선택

접미사는 디렉터리 트리의 루트에 있는 항목의 이름이며 디렉터리 데이터는 그 아래에 저장됩니다. 디렉터리에는 접미사가 두 개 이상 포함될 수 있습니다. 공통 루트가 없는 두 개 이상의 정보 트리가 있는 경우 여러 접미사를 사용할 수 있습니다. 기본적으로 표준 Directory Server 배포에는 여러 접미사가 포함되어 있으며, 하나는 구성 정보 및 디렉터리 스키마와 같은 내부 디렉터리 작업에 필요한 데이터에 대해 데이터를 저장하는 데 사용됩니다.

접미사 이름 지정 규칙

디렉터리의 모든 항목을 일반 기본 항목인 *루트 접미사로* 찾아야 합니다. 루트 디렉터리 접미사 이름을 선택하는 경우 이름을 유효하게 만들려면 다음을 수행해야 합니다.

- 전역적으로 고유

- 고정
- 이렇게 하면 아래 항목을 쉽게 읽을 수 있습니다.
- 쉽게, 사람이 입력하고 기억할 수 있도록

단일 엔터프라이즈 환경에서는 엔터프라이즈의 DNS 이름 또는 인터넷 도메인 이름과 일치하는 디렉토리 접미사를 선택할 수 있습니다. 예를 들어 엔터프라이즈에 **example.com** 의 도메인 이름이 있는 경우 디렉터리 접미사는 **dc=example,dc=com** 입니다. **dc** 속성은 도메인 이름을 구성 요소 부분으로 분할하여 접미사를 나타냅니다. 일반적으로 임의의 속성을 사용하여 루트 접미사의 이름을 지정할 수 있습니다. 그러나 호스팅 조직의 경우 루트 접미사를 다음 속성으로 제한해야 합니다.

dc

도메인 이름의 구성 요소를 정의합니다.

c

ISO에서 정의한 대로 국가 이름을 나타내는 두 자리 코드를 포함합니다.

l

항목이 위치하거나 항목과 연결된 시지, 도시 또는 기타 지리적 영역을 식별합니다.

st

항목이 있는 상태 또는 주를 식별합니다.

o

항목이 속한 조직의 이름을 식별합니다.

이러한 속성은 구독자 애플리케이션과의 상호 운용성을 제공합니다. 예를 들어 호스팅 조직은 이러한 속성을 사용하여 클라이언트 **example_a** 중 하나에 root 접미사 **o=example_a, st=Washington,c=US** 를 생성할 수 있습니다.

조직 이름 뒤에 국가 지정을 사용하는 것은 접미사에 대한 **X.500** 이름 지정 규칙에 따라 일반적입니다.

여러 접미사 이름

디렉터리의 각 접미사는 고유한 디렉터리 트리입니다. 별도의 데이터베이스 Directory Server에 저장된 여러 디렉터리 트리를 만들 수 있습니다.

예를 들어 **example_a** 및 **example_b** 에 대한 별도의 접미사를 생성하여 별도의 데이터베이스에 저장할 수 있습니다.

리소스 제한에 따라 단일 서버 또는 여러 서버에 데이터베이스를 저장할 수 있습니다.

4.2.2. 디렉터리 트리 구조 생성

플랫 또는 계층 트리 구조를 사용할지 여부를 결정합니다. 디렉터리 트리를 최대한 플랫으로 만듭니다. 그러나 복제를 준비할 때 또는 액세스 제어를 설정할 때 정보가 여러 데이터베이스에 분할되는 경우 일정 양의 계층 구조가 중요할 수 있습니다.

트리 구조에는 다음 단계와 고려 사항이 포함됩니다.

- 디렉터리 분기
- 분기 지점 식별
- 복제 고려 사항
- 액세스 제어 고려 사항

4.2.2.1. 디렉터리 분기

문제가 있는 이름 변경을 방지하려면 네임스페이스가 최대한 플랫이어야 합니다. 디렉터리 트리를 계층화할수록 이름에 더 많은 구성 요소가 있을 수 있으며 이름이 변경될 가능성이 높아집니다.

디렉터리 트리 계층 구조를 설계하려면 다음 지침을 사용합니다.

- 기업에서 가장 큰 조직 하위 분류만 표시하도록 트리를 분기합니다. 분기 지점을 기업 정보 서비스, 고객 지원, 영업 및 엔지니어링과 같은 부서로 제한해야 합니다. 디렉터리 트리를 분기하는 데 사용되는 분할이 안정적인지 확인합니다. 기업이 자주 재구성하는 경우 이러한 종류의 분기를 수행하지 마십시오.
- 분기 지점의 실제 조직 이름이 아닌 기능 또는 일반 이름을 사용합니다. 하위 트리의 이름을 바꿀 때 접미사가 많은 경우 이름 변경 프로세스가 리소스 집약적이고 오래 걸립니다. 예를 들어 research **and Development 대신 Engineering 을** 사용하십시오.
- 유사한 기능을 수행하는 조직이 여러 개 있는 경우 해당 기능에 대한 단일 분기 지점을 생성합니다. 예를 들어 여러 마케팅 조직이 있지만 각 조직이 특정 제품 라인을 담당하는 경우 각각 단일 **ou=Marketing** 하위 트리를 생성합니다. 그런 다음 모든 마케팅 항목은 해당 트리에 속합니다.

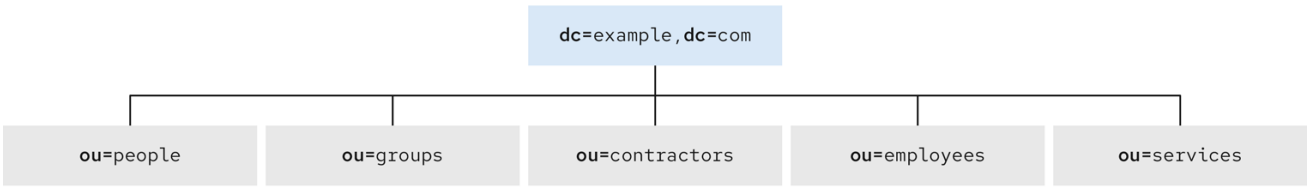
엔터프라이즈 환경의 분기

변경할 가능성이 없는 정보를 기반으로 디렉터리 트리 구조를 계획하는 경우 이름 변경을 방지할 수 있습니다. 예를 들어, 조직이 아닌 트리의 오브젝트 유형을 기반으로 하는 구조를 기반으로 하는 경우입니다.

다음 공통 오브젝트를 사용하여 구조를 정의합니다.

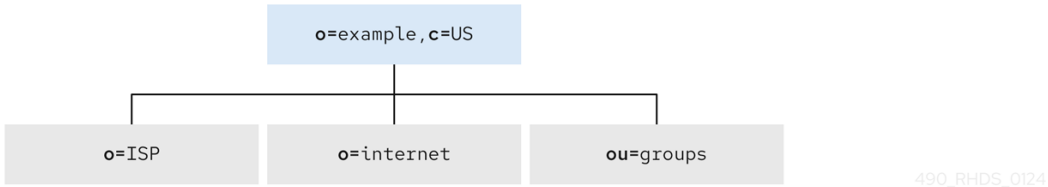
- **ou=people**
- **ou=groups**
- **ou=contracts**
- **ou=services**

다음 다이어그램은 이러한 오브젝트를 사용하여 구성된 디렉터리 트리를 보여줍니다.



호스팅 환경에서 분기

호스팅 환경의 경우 오브젝트 클래스 **조직** (O)의 두 항목이 포함된 트리와 루트 접미사 아래에 있는 개체 클래스 **organizationalUnit** (**ou**)의 하나의 항목을 만듭니다. **예를 들어 Example HCI라는 인터넷 서비스 공급자는 다음과 같은 방식으로 디렉토리를 분기합니다.**

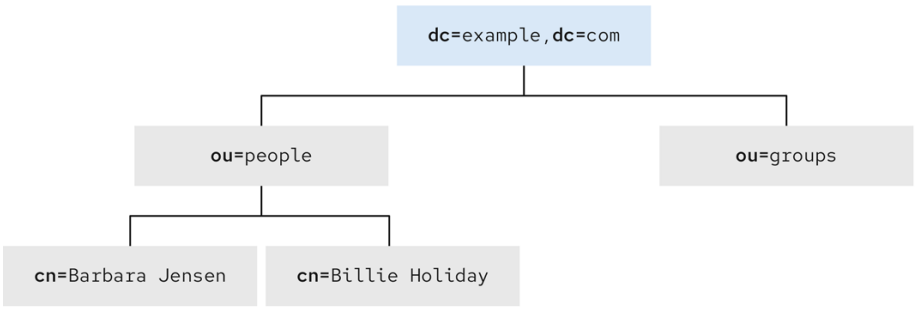


4.2.2.2. 분기 지점 식별

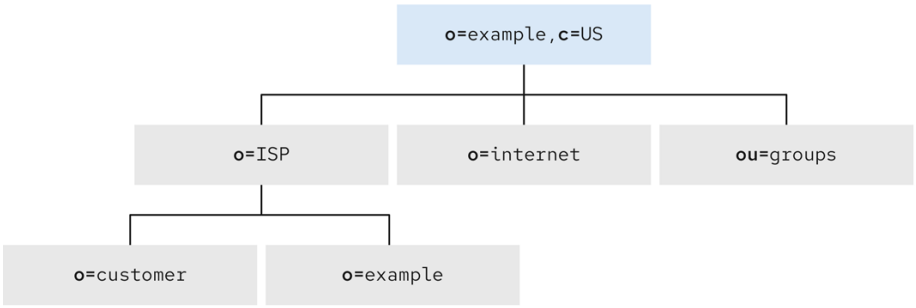
디렉터리 트리에서 분기를 계획할 때 분기 지점을 식별하는 데 사용할 속성을 결정합니다. 분기 지점은 ou=people,l=Japan,cn=Barbara Jansen 등과 같은 attribute-data 쌍입니다. DN은 이러한 특성 데이터 쌍으로 구성된 고유한 문자열입니다. 예를 들어, Example Company 의 직원인 Barbara Jensen 에 대한 항목의 DN은 다음과 같습니다.

uid=bjensen,ou=people,dc=example,dc=com.

다음 다이어그램에서 ou=people,ou=groups,cn=Barbara Jensen,cn=Billie Holiday 분기 지점이 있는 Example Company 의 디렉터리 트리의 예를 참조하십시오.



다음 다이어그램에서 인터넷 공급자 예제 **Example HCI**의 디렉터리 트리 예제를 참조하십시오.



루트 접미사 항목 **o=example,c=US** 아래에 트리가 세 개의 분기로 나뉩니다. **o=ISP** 브랜치에는 고객 데이터 및 **Example Cryostat**에 대한 내부 정보가 포함되어 있습니다. **o=internet** 분기는 도메인 트리입니다. **ou=groups** 분기에는 관리 그룹에 대한 정보가 포함되어 있습니다.

분기 지점의 속성을 선택할 때 다음 권장 사항을 고려하십시오.

- 일관성이 있어야 합니다.

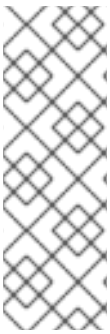
  DN 형식이 디렉터리 트리에서 일치하지 않는 경우 일부 **LDAP** 클라이언트 애플리케이션에서 **DN**(고유 이름)을 찾지 못할 수 있습니다. 디렉터리 트리의 한 부분에서 **ou** 가 **o** 아래에 있는 경우 디렉터리 서비스의 다른 모든 부분에서 **ou** 가 **o** 아래에 있는지 확인합니다.

- 기존 속성만 사용합니다.

  기존 속성을 사용하면 **Directory Server**가 타사 **LDAP** 클라이언트 애플리케이션과 호환될

가능성이 높아집니다. 기존 속성을 사용하면 기본 디렉터리 스키마가 이를 알고 있음을 의미합니다.

기존 속성	설명
dc	도메인 이름의 요소(예: **dc=example**)입니다. dc=example,dc=com 또는 **dc=mtv, dc=example,dc=com** 과 같은 도메인에 따라 쌍으로 또는 더 오래 지정됩니다. 도메인 이름 이름에 대한 자세한 내용은 접미사 이름 섹션 을 참조하십시오.
c	국가 이름입니다.
o	조직 이름입니다. 이 속성을 사용하여 기업 부서, 학문적 평가, 인도, 과학, 자회사 또는 기업 내의 기타 주요 분기와 같은 대규모 부서 분기를 나타냅니다. 이 속성을 사용하여 도메인 이름을 나타낼 수 있습니다.
ou	조직 단위입니다. 이 속성을 사용하여 조직보다 소규모 기업 분기를 나타냅니다. 조직 단위는 일반적으로 이전 조직에 종속됩니다.
st	주 또는 주 이름입니다.
L 또는 **locality**	도시, 국가, 사무실 또는 시설 이름과 같은 지역입니다.



참고

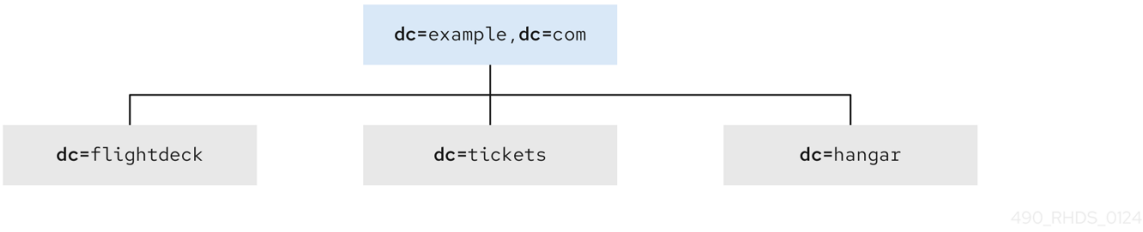
일반적인 오류는 고유 이름에 사용된 특성을 기반으로 디렉터리가 검색된다고 가정하는 것입니다. 고유 이름은 디렉터리 항목의 고유 식별자일 뿐이며 검색 키로 사용할 수 없습니다. 대신 항목 자체에 저장된 **attribute-data** 쌍을 기반으로 항목을 검색합니다. 따라서 항목 고유 이름이 **uid=bjensen,ou=People,dc=example,dc=com** 인 경우 **dc:example** 이 해당 항목의 속성으로 명시적으로 추가되지 않는 한 해당 항목과 일치하지 않습니다.

4.2.2.3. 복제 고려 사항

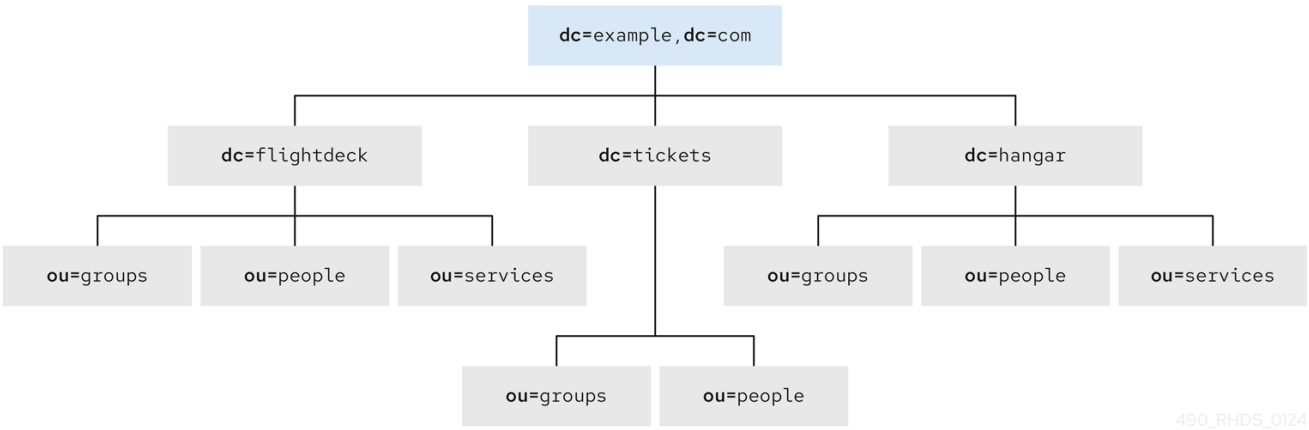
복제할 항목을 계획합니다. 하위 트리 상단에 DN을 지정하고 그 아래의 모든 항목을 복제할 수 있습니다. 이 하위 트리는 디렉터리 데이터의 일부를 포함하는 디렉터리 부분인 데이터베이스에도 해당합니다.

예를 들어 엔터프라이즈 환경에서는 엔터프라이즈의 네트워크 이름에 해당하도록 디렉터리 트리를 구성할 수 있습니다. 네트워크 이름은 변경 되지 않으므로 디렉터리 트리 구조가 안정적입니다.

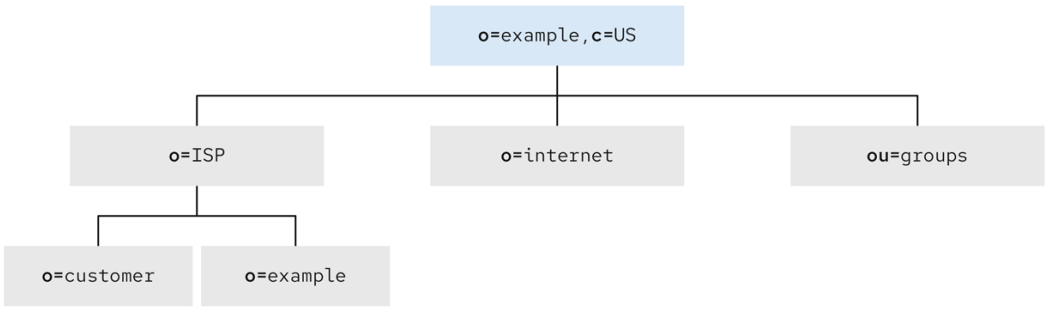
예를 들어, **Example Company**의 세 가지 주요 네트워크는**lightd Cryostat.example.com , ticket .example.com, ticket.example.com, hangar.example.com** 이라는 세 가지 주요 네트워크가 있습니다. 이 회사는 처음에 디렉터리 트리를 주요 조직 부서의 세 가지 주요 그룹으로 분기합니다. 다음 그림에서 디렉터리 트리의 초기 분기를 참조하십시오.



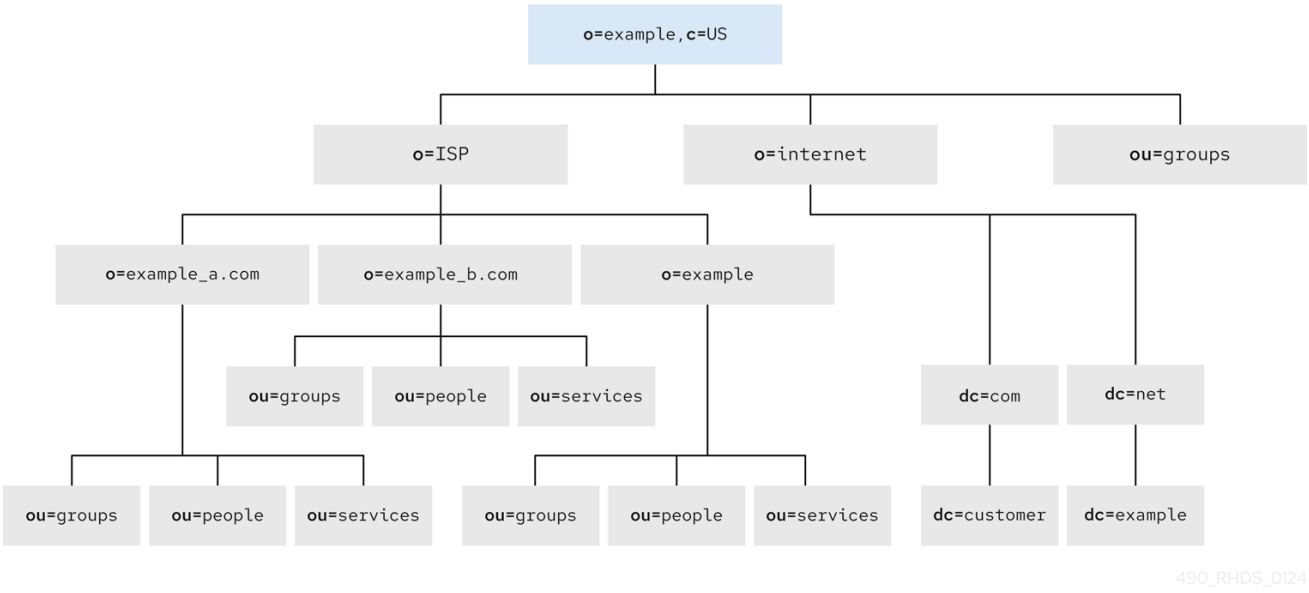
트리의 초기 구조를 만든 후 회사는 추가 분기를 생성합니다. 다음 그림에서 확장 분기를 참조하십시오.



또 다른 예에서 인터넷 공급자인 **ExampleHCI**에는 공급자 요구 사항을 충족하기 위해 다음과 같은 초기 분기가 있습니다.



나중에 **ExampleHCI**에서 논리 하위 그룹에 대한 추가 분기를 생성합니다. 다음 그림에서 확장 분기를 참조하십시오.



Enterprise Example Company와 호스팅 조직 둘 다 일반적으로 변경되지 않는 정보를 기반으로 데이터 계층을 설계합니다.

4.2.2.4. 액세스 제어 고려 사항

디렉터리 트리의 계층 구조를 사용하여 특정 유형의 액세스 제어를 활성화할 수 있습니다. 복제와 마찬가지로 유사한 항목을 그룹화한 다음 단일 분기에서 관리하는 것이 더 쉽습니다.

계층적 디렉터리 트리를 통해 관리할 수 있습니다. 예를 들어 마케팅 부서의 관리자에게 마케팅 항목에 대한 액세스 권한과 영업 부서의 관리자에게 영업 항목에 대한 액세스 권한을 부여하려면 해당 부서에

따라 디렉터리 트리를 설계합니다.

또한 디렉터리 트리가 아닌 디렉터리 콘텐츠를 기반으로 액세스 제어를 설정할 수 있습니다. **ACS**(액세스 제어 명령) 메커니즘을 사용하면 특정 항목이 특정 특성 값을 포함하는 모든 항목에 액세스할 수 있도록 허용할 수 있습니다. 예를 들어 영업 관리자에게 속성 값이 포함된 모든 항목에 액세스할 수 있는 **ACI**를 설정합니다.

그러나 **ACI**는 관리하기가 어려울 수 있습니다. 디렉터리 트리 계층 구조의 조직 분기 또는 두 가지 조합의 액세스 제어 방법을 결정합니다.

4.2.3. 항목 이름 지정

디렉터리 트리의 계층을 설계한 후 구조 내의 항목의 이름을 지정할 때 사용할 속성을 결정해야 합니다. 하나 또는 여러 특성 값을 선택할 때 *상대 고유 이름* (**RDN**)을 형성합니다. **RDN**은 **DN**의 왼쪽 부분이며, 해당 부분에 대해 선택한 속성은 *이름 지정 속성입니다*. **naming** 속성은 항목의 고유한 이름을 설정합니다. 예를 들어 **DN uid=bjensen,ou=people,dc=example,dc=com** 에는 **RDN uid=bjensen.**

선택한 속성은 이름 지정에 따라 다릅니다.

항목의 이름을 지정할 때는 다음을 고려하십시오.

- 이름 지정에 대해 선택한 특성을 변경하지 않아야 합니다.
- 이름은 디렉터리에서 고유해야 합니다. 고유한 이름을 사용하면 **DN**이 디렉터리의 하나의 항목만 나타냅니다.

항목을 생성할 때 항목 내에 **RDN**을 정의합니다. 항목 내에 정의된 **RDN**을 사용하면 항목을 보다 쉽게 배치할 수 있습니다. 이는 검색에서 실제 **DN**을 기반으로 하지 않고 항목 자체에 저장된 속성 값을 기반으로 하는 항목을 찾기 때문입니다.

특성 이름에는 의미가 있으므로 나타내는 항목 유형과 일치하는 속성 이름을 사용하십시오. 예를 들어 조직을 나타내는 데 **l** (로케이션)을 사용하거나 조직 단위를 나타내는 **c** (국가)를 사용하지 마십시오.

4.2.3.1. 디렉터리 트리에서 사용자 항목 이름 지정

사용자 항목 이름은 고유해야 합니다. 일반적으로 사람 항목의 이름을 지정하려면 **commonName** 또

는 cn 을 사용하여 상대 고유 이름(RDN)을 형성합니다. 예를 들어 Babs Jensen 이라는 이름의 항목에는 고유 이름(DN)이 cn=Babs Jensen,dc=example,dc=com 임을 가질 수 있습니다.

RDN에서 공통 이름만 사용하면 항목 이름을 고유하게 만들고 DN 이름 충돌을 유발하는 여러 동일한 항목이 생성될 수 있습니다.

cn=Babs Jensen+employeeNumber=23,dc=example,dc=com 과 같은 일반 이름 충돌을 방지합니다. 그러나 이로 인해 대규모 디렉터리의 일반 이름이 어색하게 될 수 있으며 유지 관리하기 어려울 수 있습니다.

더 나은 방법은 cn 이외의 일부 특성을 사용하여 사람 항목을 식별하는 것입니다. 다음 속성 중 하나를 사용하는 것이 좋습니다.

uid

uid 속성을 사용하여 사용자 로그인 ID 또는 직원 번호와 같은 특정 사용자 값을 지정합니다. uid 특성을 통해 호스팅 환경에서 구독자를 식별합니다.

mail

mail 속성에는 항상 고유한 사람 이메일 주소가 포함되어 있습니다. 이 속성은 mail=bjensen@example.com,dc=example,dc=com 과 같은 중복 속성 값을 포함하는 어색한 DN으로 이어질 수 있습니다. uid 속성의 고유 값을 찾을 수 없는 경우에만 이 옵션을 사용합니다. 예를 들어 엔터프라이즈에서 임시 또는 계약직원에 직원 번호 또는 사용자 ID를 할당하지 않는 경우 uid 속성 대신 mail 속성을 사용합니다.

employeeNumber

inetOrgPerson 오브젝트 클래스의 직원의 경우 employeeNumber 특성을 사용합니다.

직접 항목 RDN에 대해 특성-데이터 쌍에 사용하는 경우 고유하고 영구 값인지 확인합니다. RDN 사용자 항목도 읽을 수 있어야 합니다. 예를 들어 DN uid=bjensen,dc=example,dc=com 은 uid=b12r56A,dc=example,dc=com 보다 더 선호되며 고유 이름을 기반으로 디렉터리 항목 변경과 같은 일부 디렉터리 작업을 간소화합니다. 또한 일부 디렉터리 클라이언트 애플리케이션에서는 uid 및 cn 속성이 사람이 읽을 수 있는 이름을 사용한다고 가정합니다.

호스트된 환경의 사용자 항목에 대한 고려 사항

사용자가 서비스에 대한 구독자인 경우 항목에 inetUser 오브젝트 클래스가 있어야 하며 uid 속성을 포함해야 합니다. 속성은 customer 하위 트리 내에서 고유해야 합니다.

사람이 호스팅 조직의 일부인 경우 nsManagedPerson 개체 클래스와 함께 inetOrgPerson 속성을

사용합니다.

디렉터리 트리에 직접 항목 배치

디렉터리 트리에 사람 항목을 배치하려면 다음 지침을 사용합니다.

- 디렉터리 트리의 조직 항목 아래에 있는 엔터프라이즈에서 사용자를 찾습니다.
- 호스팅 조직의 **ou=people** 분기 아래에 있는 호스팅 조직의 구독자를 찾습니다.

4.2.3.2. 디렉터리 트리에서 그룹 항목 이름 지정

다음 방법을 사용하여 그룹을 나타낼 수 있습니다.

- *정적 그룹은* 멤버를 명시적으로 정의합니다. **groupOfNames** 또는 **groupOfUniqueNames** 오브젝트 클래스에는 그룹 멤버의 이름을 지정하는 값이 포함되어 있습니다. 정적 그룹은 디렉터리 관리자 그룹과 같이 멤버가 적은 그룹에 적합하며 수천 명의 멤버가 있는 그룹에 적합하지 않습니다.

  uniqueMember 는 **groupOfUniqueNames** 오브젝트의 필수 특성이므로 정적 그룹 항목에는 **uniqueMember** 속성 값이 포함되어야 합니다. 이 오브젝트 클래스에는 그룹 항목의 **DN**을 형성하는 데 사용할 수 있는 **cn** 속성이 필요합니다.

- *동적 그룹은* 필터를 지정하고 필터와 일치하는 모든 항목은 이 그룹의 멤버입니다.
- *역할은* 정적 및 동적 그룹 개념을 통합합니다.

호스팅된 환경에서는 **groupOfUniqueNames** 오브젝트 클래스를 사용하여 디렉터리 관리에 사용되는 그룹의 멤버라는 값을 포함하는 것이 좋습니다.

또한 **ou=Groups** 분기에서 디렉터리 관리에 사용하는 그룹 항목을 찾습니다.

추가 리소스

- 디렉터리 항목 그룹화

4.2.3.3. 조직 항목 이름 지정

조직 항목 이름은 고유해야 합니다. 조직의 법적 이름을 다른 속성 값과 함께 사용하면 **o=example_a+st=Washington,o=ISP,c=US** 와 같이 이름이 고유한지 확인하는 데 도움이 됩니다.

상표를 사용할 수도 있지만 고유하지는 않을 수 있습니다.

호스팅 환경에서 조직 항목에 다음 속성을 포함합니다.

- **O (organizationName)**
- **top,organization, nsManagedDomain**의 값이 있는 **objectclass**

4.2.3.4. 다른 항목 이름 지정

디렉터리에는 지역, 상태, 국가, 장치, 서버, 네트워크 정보 및 기타 데이터 유형과 같은 다양한 정보를 나타내는 항목이 포함되어 있습니다. 이러한 유형의 항목에 대해 **RDN**에서 **cn** 속성을 사용합니다. 그룹 항목의 이름을 **cn=administrators,dc=example,dc=com** 으로 지정할 수도 있습니다.

경우에 따라 항목 오브젝트 클래스에서 **commonName** 속성을 지원하지 않는 경우가 있습니다. 대신 항목 오브젝트 클래스에서 지원하는 속성을 사용합니다. 이름 지정 속성은 항목에서 실제로 사용하는 속성에 해당하지 않아도 됩니다. 그러나 항목에 사용되는 **DN** 속성과 특성 간의 상관 관계가 있는 경우 디렉터리 트리를 더 쉽게 관리할 수 있습니다.

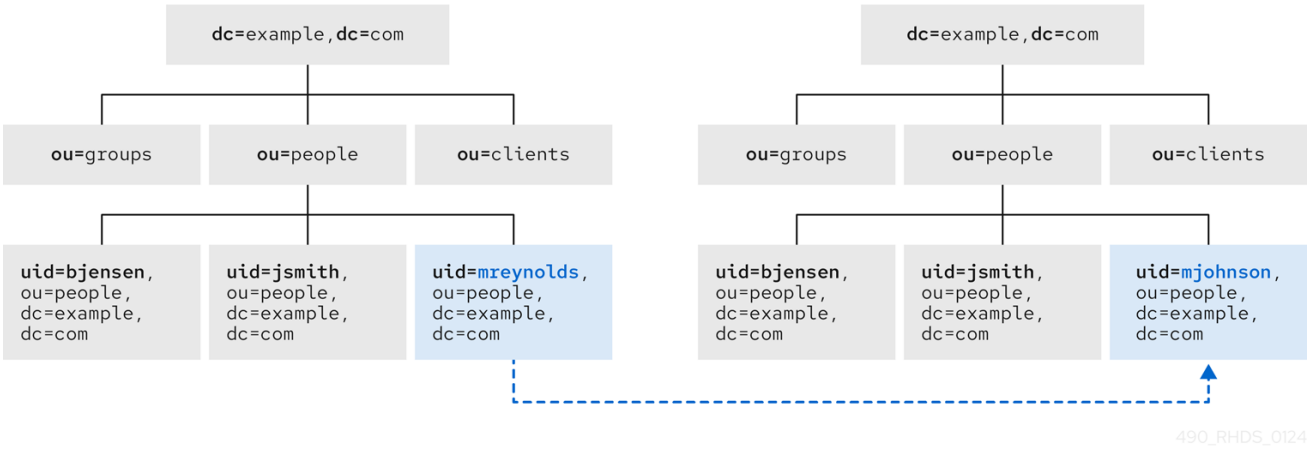
4.2.4. 항목 및 하위 트리 이름 변경

항목 이름은 디렉터리 트리 구조를 정의합니다. 각 분기 지점은 계층에 새 링크를 생성합니다.

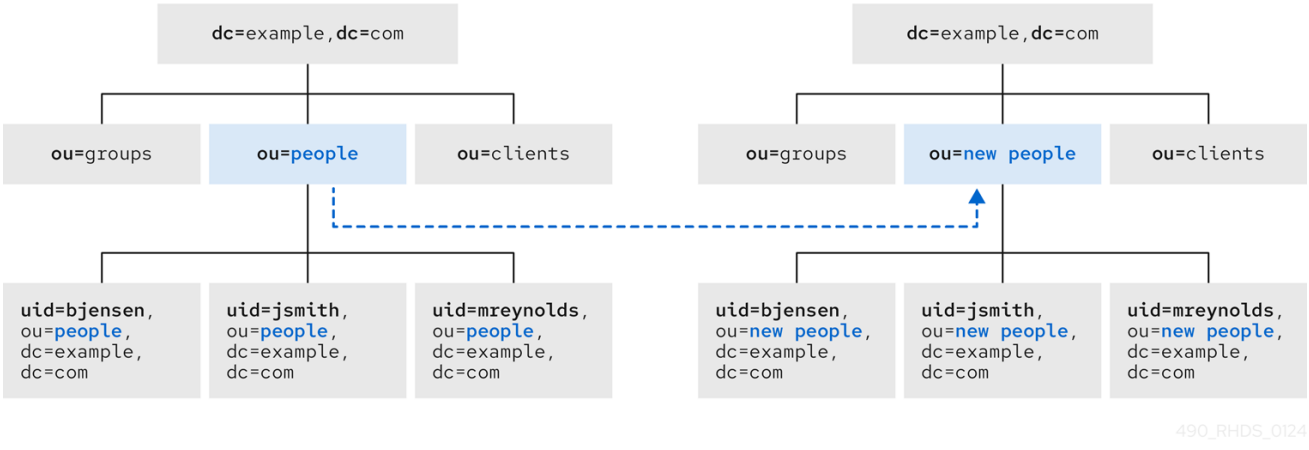
```
dc=example,dc=com  =>  root suffix
ou=People,dc=example,dc=com  =>  org unit
st=California,ou=People,dc=example,dc=com  =>  state/province
l=Mountain View,st=California,ou=People,dc=example,dc=com  =>  city
ou=Engineering,l=Mountain View,st=California,ou=People,dc=example,dc=com  =>  org
```

> **unit**
> **uid=jsmith,ou=Engineering,l=Mountain View,st=California,ou=People,dc=example,dc=com => leaf entry**

항목의 이름 지정 특성, 항목 **RDN**을 변경하면 ***modrdn*** 작업을 수행합니다. 이 수정 작업은 디렉터리 트리 내에서 항목을 이동합니다. 리프 항목(자식이 없는 항목)의 경우 **modrdn** 작업만 **RDN** 부분만 변경하므로 상위 항목은 동일하게 유지됩니다.



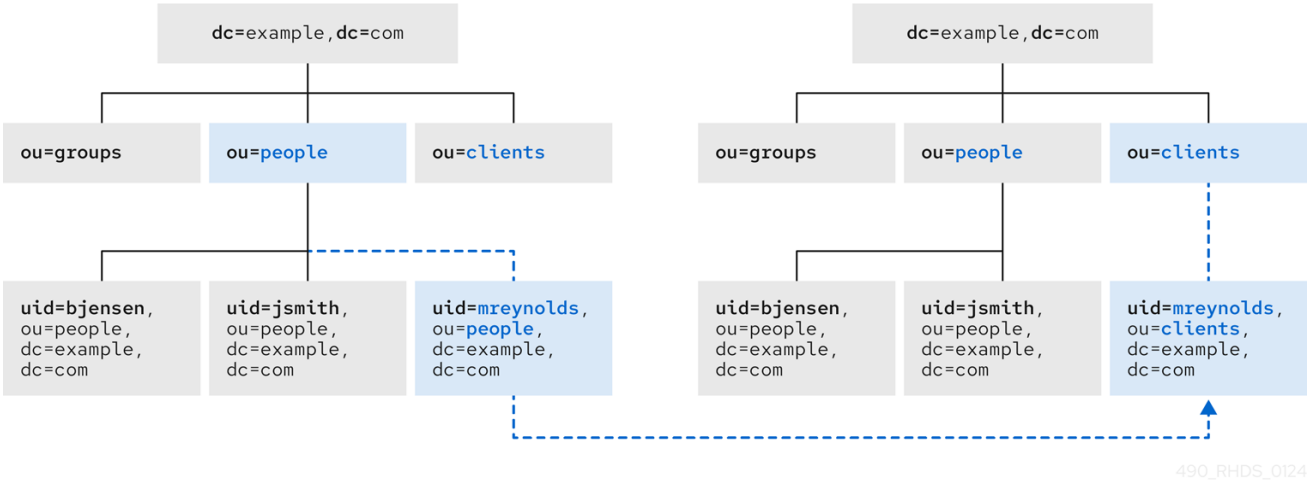
하위 트리 항목의 경우 **modrdn** 작업에서는 하위 트리 항목 자체의 이름을 바꾸고 하위 트리 아래의 모든 하위 항목의 **DN** 구성 요소를 변경합니다.



중요

하위 트리 **modrdn** 작업도 이동 및 하위 트리 항목 아래의 모든 하위 항목의 이름을 변경합니다. 대규모 하위 트리의 경우 시간과 리소스를 많이 사용하는 프로세스가 될 수 있습니다. 자주 하위 트리 이름 변경 작업이 필요하지 않도록 디렉터리 트리 계층 구조의 이름 지정 구조를 계획합니다.

하위 트리의 이름을 바꾸는 것과 유사한 작업이 한 하위 트리에서 다른 하위 트리로 항목을 이동하는 것입니다. 확장된 이러한 유형의 **modrdn** 작업은 동일한 이름인 경우에도 항목의 이름을 동시에 바꾸고 항목을 부모 간에 이동하는 **newsuperior** 특성을 설정합니다.



Directory Server는 **entryrdn.db** 인덱스를 사용하여 새로운 정상 및 하위 트리 이름 변경 작업을 수행합니다. **Directory Server**는 각 항목을 자체 링크, 상위 링크 및 하위 링크로 식별합니다. **entryrdn.db** 인덱스는 부모와 하위 항목을 항목의 속성으로 제공하고 모든 항목을 전체 **DN**이 아닌 고유 **ID**와 **RDN**으로 설명합니다.

entryrdn.db 인덱스의 형식은 다음과 같습니다.

```
numeric_id:RDN => self link
    ID: ; RDN: "rdn"; NRDN: normalized_rdn P:RDN => parent link
    ID: ; RDN: "rdn"; NRDN: normalized_rdn C:RDN => child link
    ID: #; RDN: "rdn"; NRDN: normalized_rdn
```

예를 들어 **ou=people** 하위 트리에는 **dc=example,dc=com** 상위 및 **uid=jsmith** 하위 항목이 있습니다. **entryrdn.db** 인덱스에는 다음과 같은 내용이 있습니다.

```
4:ou=people
  ID: 4; RDN: "ou=people"; NRDN: "ou=people"
P4:ou=people
  ID: 1; RDN: "dc=example,dc=com"; NRDN: "dc=example,dc=com"
C4:ou=people
  ID: 10; RDN: "uid=jsmith"; NRDN: "uid=jsmith"
```

이름 변경 작업을 수행할 때는 다음을 고려하십시오.

- 루트 접미사의 이름을 변경할 수 없습니다.

- 복제 계약을 재구성할 필요가 없습니다. Directory Server는 데이터베이스 내의 하위 트리가 아닌 전체 데이터베이스에 복제 계약을 적용합니다.

- 하위 트리 이름 변경 작업 후 모든 동기화 계약을 재구성해야 할 수 있습니다. 동기화 계약은 접미사 또는 하위 트리 수준에서 설정되므로 하위 트리의 이름을 변경하면 동기화가 중단될 수 있습니다.

- 하위 트리의 하위 항목에 대해 설정된 모든 하위 트리 수준 ACI 및 모든 항목 수준 ACI 세트를 수동으로 재구성해야 합니다.

- 하위 트리의 이름을 자식으로 변경할 수는 있지만 하위 트리를 자식으로 삭제할 수는 없습니다.

- ou 에서 dc 로 이동하는 것과 같이 하위 트리의 구성 요소를 변경하려고 하면 스키마 위반으로 실패할 수 있습니다. 예를 들어 organizationalUnit 오브젝트 클래스에는 ou 특성이 필요합니다. 작업이 organizationalUnit 개체 클래스에서 ou 특성을 제거하려고 하면 하위 트리 작업이 실패합니다.

4.3. 디렉터리 항목 그룹화

디렉터리 관리를 단순화하기 위해 생성한 그룹 항목. Directory Server는 항목 메서드를 그룹화하는 다음과 같은 방법을 지원합니다.

- 그룹

- 역할

4.3.1. Directory Server의 그룹 정보

그룹은 사용자 컬렉션입니다. Directory Server에는 고유한 멤버만 있는 인증서 그룹, URL 그룹 및 고유 그룹과 같이 허용되는 멤버십 유형을 반영하는 여러 그룹 유형이 있습니다. 각 유형의 그룹(예: groupOfUniqueNames)과 같은 해당 멤버 특성(예: uniqueMember)을 정의합니다.

그룹 유형은 멤버 유형을 식별합니다. 그룹 구성은 **Directory Server**가 그룹에 멤버를 추가하는 방법에 따라 달라집니다. **Directory Server**에는 다음 두 가지 그룹 유형이 있습니다.

정적 그룹

정적 그룹에는 유한 및 정의된 멤버 목록이 있습니다. 그룹 항목에 멤버를 수동으로 추가합니다.

동적 그룹

동적 그룹은 필터를 사용하여 그룹에 멤버를 추가합니다. 따라서 그룹 필터와 일치하는 항목 수가 변경되므로 멤버 수가 지속적으로 변경됩니다.

그룹은 항목에 대한 작업을 수행하지 않지만 **LDAP** 클라이언트는 작업을 수행하기 위해 그룹을 관리할 수 있습니다.

4.3.1.1. 사용자 항목에 그룹 멤버십 나열

그룹은 사용자 **DN**의 목록입니다. 기본적으로 그룹 항목에는 멤버십 정보가 포함되어 있으며 사용자 항목에 이 정보가 포함되지 않습니다.

MemberOf 플러그인은 그룹 멤버 항목을 사용하여 사용자 항목을 동적으로 업데이트하고 사용자가 속한 그룹을 반영합니다. 플러그인은 지정된 멤버 특성을 사용하여 그룹 항목을 자동으로 검색하고 모든 사용자 **DN**을 추적하며 그룹 이름을 사용하여 사용자 항목에 해당 **memberOf** 특성을 생성합니다.

사용자가 속한 모든 그룹의 이름은 **memberOf** 속성으로 나열되며 **memberOf** 속성 값을 관리할 수 있습니다.

참고

기본적으로 **MemberOf** 플러그인은 **Directory Server**가 그룹과 동일한 데이터베이스에 저장하는 사용자 내의 잠재적 멤버만 검색합니다. **Directory Server**가 사용자 및 그룹을 다른 데이터베이스에 저장하는 경우, 플러그인에서 사용자와 그룹 간의 관계를 정의할 수 없기 때문에 **MemberOf** 플러그인은 사용자 항목을 업데이트하지 않습니다.

memberOfAllBackends 속성을 활성화하여 구성된 모든 데이터베이스를 검색하도록 **MemberOf** 플러그인을 구성합니다.

플러그인 항목에서 다중 값 **memberofgroupattr** 을 설정하여 여러 유형의 그룹을 관리하도록 **MemberOf** 플러그인의 단일 인스턴스를 구성할 수 있습니다.

4.3.1.2. 그룹에 자동으로 새 항목 추가

그룹 멤버십에 따라 암호 정책, 액세스 제어 목록 및 기타 규칙을 적용할 수 있습니다. 그룹을 사용하면 디렉터리에 정책을 일관되고 안정적으로 적용할 수 있습니다.

새 항목이 생성될 때 새 항목을 자동으로 할당하면 **Directory Server**가 관리자 작업 없이 해당 항목에 즉시 적절한 정책과 기능을 적용할 수 있습니다.

Automembership 플러그인을 사용하면 정적 그룹이 동적 그룹처럼 작동할 수 있습니다. **Automembership** 플러그인은 항목 특성, 디렉터리 위치 및 정규식에 따라 일련의 규칙을 사용하여 사용자를 지정된 그룹에 자동으로 할당합니다.

LDAP 검색 필터와 일치하는 항목을 다른 속성의 값에 따라 다른 그룹에 추가해야 하는 인스턴스가 있을 수 있습니다. 예를 들어 **IP** 주소 또는 물리적 위치에 따라 시스템을 다른 그룹에 추가해야 합니다. 또는 직원 **ID** 번호에 따라 사용자를 다른 그룹에 배치해야 합니다.

automember 정의는 중첩된 항목 세트로, **Auto membership** 플러그인 컨테이너, **automember** 정의, 해당 정의에 대한 정규식 조건입니다.

참고

Directory Server는 항목이 **Directory Server**에 새로 추가된 경우에만 그룹에 항목을 자동으로 할당합니다. **automember** 규칙을 충족하도록 수정한 기존 항목 또는 항목의 경우 수정 작업을 실행하여 적절한 그룹 멤버십을 할당합니다.

4.3.2. Directory Server의 역할 정보

역할은 정적 그룹 및 동적 그룹으로 작동합니다. 그룹을 사용하면 **Directory Server**에서 그룹 항목에 항목을 멤버로 추가합니다. 역할로 **Directory Server**는 역할 속성을 항목에 추가한 다음 해당 속성을 사용하여 역할 항목의 멤버를 자동으로 식별합니다.

역할을 사용하면 다음과 같은 방법으로 사용자를 구성할 수 있습니다.

- 역할 멤버를 명시적으로 나열합니다. 역할을 볼 때 이 역할에 대한 전체 멤버 목록을 볼 수 있습니다. 역할을 쿼리하여 동적 그룹에서 수행할 수 없는 멤버십을 확인할 수 있습니다.

-

항목이 속한 역할을 확인합니다. 항목을 볼 때 **Directory Server**가 항목의 특성으로 역할 멤버십을 결정하므로 항목이 속한 역할을 확인할 수 있습니다. 그룹의 **memberOf** 속성과 유사합니다. 유일한 차이점은 이 기능이 작동하도록 플러그인 인스턴스를 활성화하거나 구성할 필요가 없다는 것입니다.

- 적절한 역할을 할당합니다. **Directory Server**는 역할이 아닌 항목을 통해 역할 멤버십을 할당합니다. 따라서 단일 단계에서 항목을 편집하여 사용자가 속한 역할을 쉽게 할당하고 제거할 수 있습니다.

관리 역할은 정적 그룹으로 수행할 수 있는 모든 작업을 수행할 수 있습니다. 동적 그룹으로 필터링하는 것과 유사하게 필터링된 역할을 사용하여 역할 멤버를 필터링할 수 있습니다. 역할은 구현에 더 유연하고 클라이언트의 복잡성을 줄이기 때문에 그룹보다 쉽게 사용할 수 있습니다.

역할 유형을 사용하여 명시적으로 또는 동적으로 멤버를 지정할 수 있습니다. **Directory Server**는 다음 유형의 역할을 지원합니다.

관리 역할

관리 역할에는 명시적 멤버 목록이 있습니다.

필터링된 역할

항목에 역할에 정의된 특정 특성이 있는 경우 **Directory Server**는 필터링된 역할에 항목을 할당합니다. 역할 정의는 대상 속성에 대한 **LDAP** 필터를 지정합니다. 필터와 일치하는 항목에는 역할이 있음(의 멤버임)입니다.

중첩된 역할

중첩된 역할은 다른 역할을 포함하는 역할입니다.

하나의 작업에서 전체 항목 그룹을 활성화하거나 비활성화할 수 있습니다. 해당 역할이 속한 역할을 비활성화하여 역할 멤버를 일시적으로 비활성화할 수 있습니다.

역할을 비활성화하면 사용자는 역할 항목을 사용하여 서버에 계속 바인딩할 수 있습니다. 그러나 사용자는 이 역할에 속하는 항목을 사용하여 서버에 바인딩할 수 없습니다. 비활성화된 역할에 속하는 항목에는 **nsAccountLock** 속성이 **true** 로 설정되어 있습니다.

중첩된 역할을 비활성화하면 중첩된 역할 내의 모든 역할의 멤버인 경우 사용자가 서버에 바인딩할 수 없습니다. 중첩된 역할의 직접 또는 간접 멤버인 역할에 속하는 모든 항목은 **nsAccountLock** 을 **true** 로 설정합니다. 중첩의 어느 시점에서나 중첩 역할을 비활성화하면 해당 아래의 모든 역할과 사용자가 비활성화됩니다.

4.3.3. 그룹과 역할 결정

역할과 그룹은 동일한 목표를 달성할 수 있습니다. 관리되는 역할은 정적 그룹이 수행할 수 있는 모든 작업을 수행할 수 있지만 필터링된 역할은 동적 그룹과 동일한 방식으로 멤버를 필터링하고 식별할 수 있습니다. 역할과 그룹 모두 장단점이 있습니다. 역할 또는 그룹 또는 혼합 사용 여부를 결정하는 것은 요구 사항과 서버 리소스의 균형을 조정하는 데 따라 달라집니다.

역할은 클라이언트 측의 복잡성을 줄입니다. 역할을 사용하면 클라이언트 애플리케이션에서 항목에서 **nsRole** 작동 속성을 검색하여 역할 멤버십을 확인할 수 있습니다. 이 다중 값 속성은 항목이 속한 모든 역할을 식별합니다. 클라이언트 애플리케이션 관점에서 멤버십을 확인하는 방법은 균일하며 서버 측에서 수행됩니다.

그러나 역할에는 서버 복잡성이 증가해야 합니다. 역할을 평가하는 것은 서버가 클라이언트 애플리케이션에 대해 작업을 수행하기 때문에 그룹을 평가하는 것보다 리소스 집약적입니다.

그룹은 효과적으로 사용하기 위해 더 스마트하고 복잡한 클라이언트가 필요합니다. 예를 들어 애플리케이션 관점에서 동적 그룹은 그룹 멤버 목록을 제공하기 위해 서버에서 지원하지 않습니다. 대신 애플리케이션은 그룹 정의를 검색한 다음 필터를 실행합니다. 사용자 항목에는 적절한 플러그인을 구성하는 경우에만 그룹 멤버십 정보가 포함됩니다.

참고

MemberOf 플러그인을 사용하여 그룹 멤버십 관리의 균형을 조정할 수 있습니다. **MemberOf** 플러그인은 사용자가 그룹에 추가될 때마다 사용자 항목에 **memberOf** 속성을 동적으로 생성합니다. 클라이언트는 그룹 항목에서 단일 검색을 실행하여 모든 멤버 목록을 가져오거나 사용자 항목에서 단일 검색을 실행하여 속하는 모든 그룹의 전체 목록을 가져올 수 있습니다.

멤버십이 수정되는 경우에만 서버에는 유지 관리 오버헤드가 있습니다. **Directory Server**는 지정된 멤버 (그룹) 및 **memberOf** (사용자) 속성을 모두 데이터베이스에 저장하므로 검색에는 추가 처리가 필요하지 않으므로 클라이언트의 검색이 매우 효율적입니다.

추가 리소스

- 사용자 항목에 그룹 멤버십 나열
- 그룹에 자동으로 새 항목 추가

4.4. 가상 디렉터리 정보 트리 뷰

Directory Server는 *가상 뷰인 가상 디렉터리 정보 트리* 보기를 지원합니다. 가상 뷰는 항목을 분류하고 검색하는 표준 디렉터리 트리 외에도 선택적 구조 계층입니다.

참고

가상 보기는 여러 백엔드와 완전히 호환되지 않습니다. 검색은 하나의 백엔드로 제한되므로 가상 뷰가 반환하는 항목은 동일한 백엔드에 있어야 합니다.

가상 DIT 뷰에 대한 자세한 내용은 뷰를 사용하여 가상 디렉터리 계층 구조 만들기를 참조하십시오.For more information about virtual DIT views, see Using views to create a virtual directory hierarchy.

4.4.1. 가상 DIT 보기 예

아래의 LDIF 항목은 위치를 기반으로 하는 가상 뷰 계층 구조를 보여줍니다. dc=example,dc=com 아래에 있고 view 설명에 맞는 모든 항목이 위치별로 구성된 이 보기에 표시됩니다.

```
dn: ou=Location Views,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
objectclass: nsView
ou: Location Views
description: views categorized by location


dn: ou=Sunnyvale,ou=Location Views,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
objectclass: nsView
ou: Sunnyvale
nsViewFilter: (l=Sunnyvale)
description: views categorized by location


dn: ou=Santa Clara,ou=Location Views,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
objectclass: nsView
ou: Santa Clara
nsViewFilter: (l=Santa Clara)
description: views categorized by location


dn: ou=Cupertino,ou=Location Views,dc=example,dc=com
objectclass: top
objectclass: organizationalUnit
objectclass: nsView
```

```
ou: Cupertino
nsViewFilter: (l=Cupertino)
description: views categorized by location
```

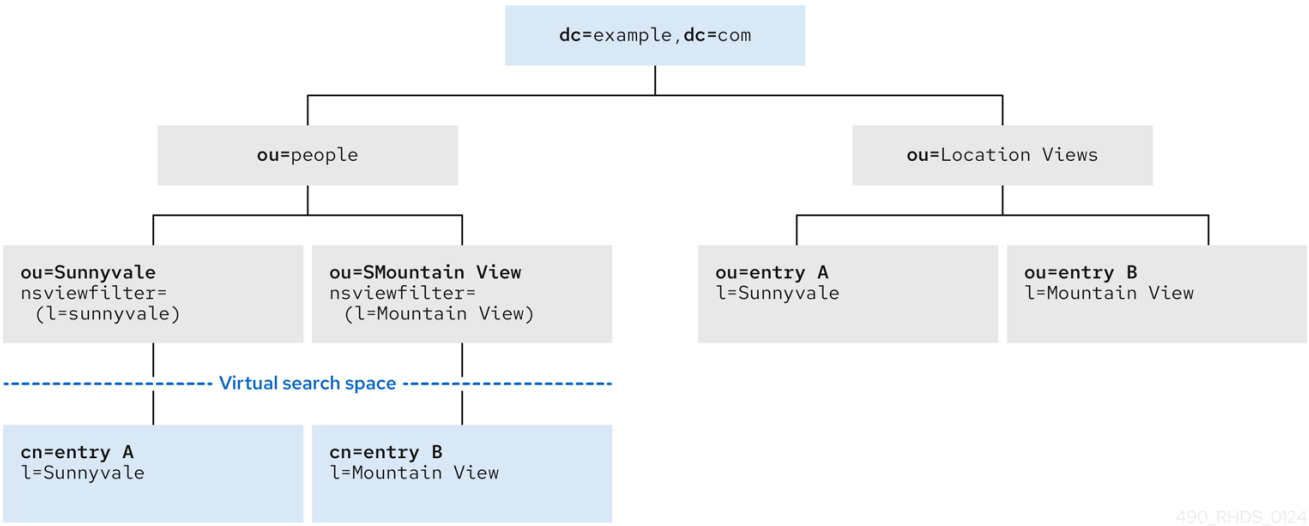
ou=Location Views,dc=example,dc=com 에 기반한 하위 트리 검색은 필터 **(l=Sunnyvale), (l= Santa Cextensiona)** 또는 **(l=Cupertino)** 와 일치하는 **dc= example,dc=com** 아래의 모든 항목을 반환합니다. 그러나 한 수준 검색에서는 모든 적격 항목이 세 하위 뷰에 있기 때문에 하위 뷰 항목이 아닌 항목을 반환하지 않습니다.

ou=Location Views,dc=example,dc=com view 항목 자체에는 필터가 포함되어 있지 않습니다. 이 기능은 보기에 포함된 항목을 추가로 제한해야 하는 요구 없이 계층적 조직을 용이하게 합니다. 모든 보기에서 필터를 생략할 수 있습니다.

예제 필터는 매우 간단하지만 사용하는 필터는 필요에 따라 복잡할 수 있습니다. 뷰에 포함되어야 하는 항목 유형을 제한할 수 있습니다. 예를 들어 이 계층 구조에 사람 항목만 포함하도록 제한하려면 필터 값 **(objectclass=organizationalperson)** 을 사용하여 **ou=Location Views,dc=example,dc=com** 에 **nsfilter** 특성을 추가합니다.

필터가 있는 각 뷰는 모든 하위 뷰의 콘텐츠를 제한하지만 필터가 있는 하위 뷰도 상위 콘텐츠를 제한합니다. 예를 들어 위에서 언급한 새 필터와 먼저 상위 뷰 **ou=Location Views** 를 생성하면 조직 오브젝트 클래스가 포함된 모든 항목이 있는 뷰가 생성됩니다. 하위 뷰가 추가되어 항목을 추가로 제한할 때 하위 보기에 이제 하위 보기가 표시되는 항목이 상위 보기에서 제거됩니다. 이는 가상 **DIT** 뷰가 기존 **DIT**의 동작을 에뮬레이션하는 방법을 보여줍니다.

가상 **DIT** 보기는 기존 **DIT**의 동작을 에뮬레이션하지만 기존 **DIT**는 할 수 없는 작업을 수행할 수 있습니다: 항목이 두 개 이상의 위치에 표시될 수 있습니다. 예를 들어 **Entry B** 를 **192.0.2. View** 및 **Sunnyvale** 모두와 연결하려면 **Sunnyvale** 값을 **location** 속성에 추가하고 항목이 두 보기 모두에 표시됩니다.



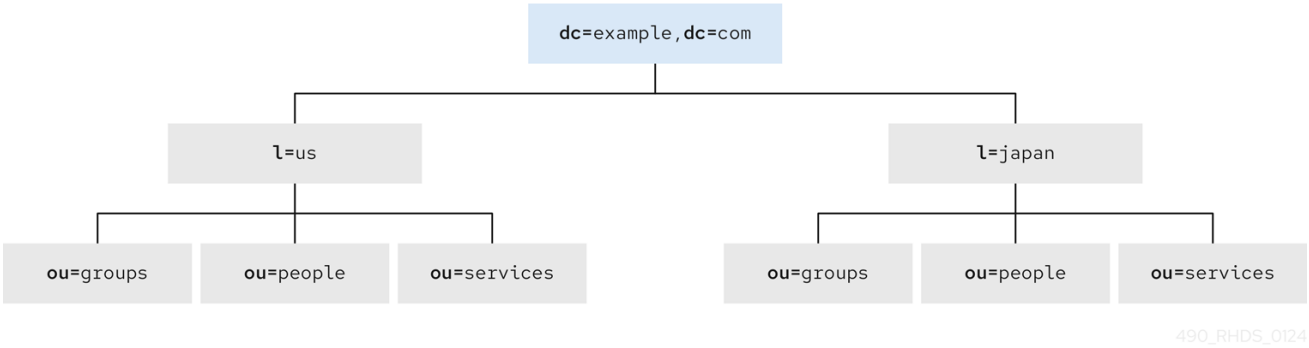
4.5. 디렉터리 트리 설계 예

국제 엔터프라이즈 및 Cryostat용 디렉터리 트리의 예를 찾습니다.

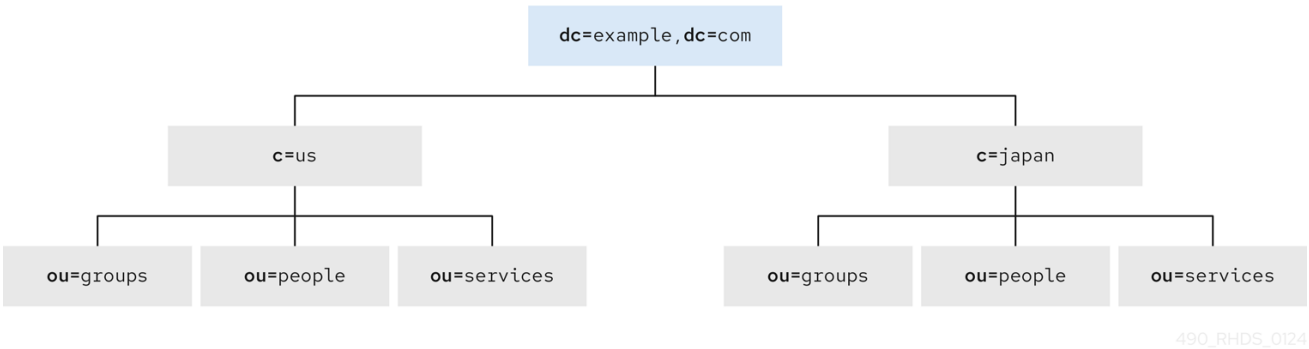
국제 엔터프라이즈용 디렉터리 트리

인터넷 도메인 이름을 디렉터리 트리의 루트 항목으로 사용합니다. 그런 다음 엔터프라이즈가 작업을 수행하는 각 국가의 루트 항목 아래에 트리를 분기합니다.

다른 국가를 표현하려면 l (location) 특성을 사용합니다.



그러나 c (country) 속성은 각 국가 분기를 나타낼 수도 있습니다.



LDAP에서는 DN의 속성 순서에 제한이 없습니다.

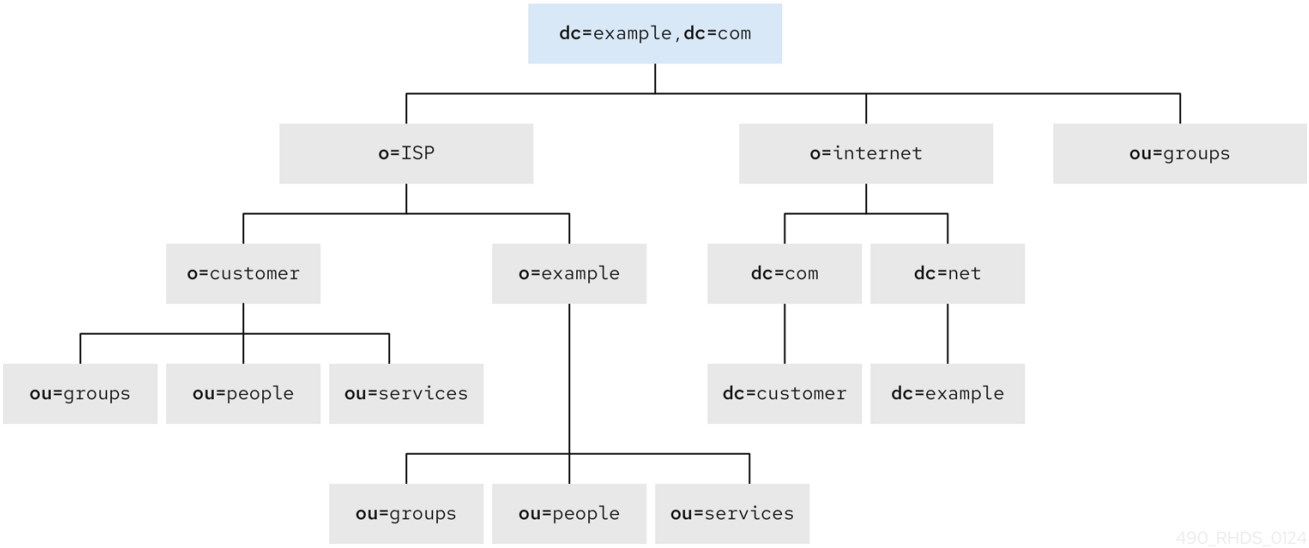
Cryostat의 디렉터리 트리

인터넷 서비스 공급자(ISP)는 해당 디렉토리를 통해 여러 엔터프라이즈를 지원할 수 있습니다. Cryostat는 각 고객을 고유한 엔터프라이즈로 간주하고 그에 따라 디렉터리 트리를 설계해야 합니다. 보

안상의 이유로 각 고객에 대한 고유한 접미사와 독립적인 보안 정책을 포함하는 고유한 디렉터리 트리를 제공합니다.

각 고객에게 별도의 데이터베이스를 할당하고 이러한 데이터베이스를 별도의 서버에 저장합니다. 각 디렉터리 트리를 자체 데이터베이스에 배치하면 다른 고객에 영향을 주지 않고 각 디렉터리 트리의 데이터를 백업하고 복원할 수 있습니다.

또한 파티션은 디스크 경합으로 인한 성능 문제와 디스크 중단이 잠재적으로 영향을 미칠 수 있는 고객 수를 줄입니다.



4.6. 추가 리소스

- RFC 2247: LDAP/X.500 고유 이름에 도메인 사용
- RFC 2253: LDAPv3, UTF-8 문자열 구분

5장. 디렉터리 토폴로지 설계

Red Hat Directory Server는 많은 수의 항목을 저장할 수 있으며 결과적으로 여러 서버에 항목을 배포해야 할 수 있습니다. 디렉터리 토폴로지는 디렉터리 트리를 여러 물리적 디렉터리 서버로 나누는 방법과 이러한 서버가 연결되는 방법을 설명합니다.

5.1. 토폴로지 개요

Directory Server는 여러 물리적 *디렉터리 서버에서 Designing-the-directory-tree 에서 설계한 디렉터리 트리를 분배하는 분산* 디렉터리를 지원합니다. 해당 서버에서 디렉토리를 나누는 방법은 다음과 같은 성능 관련 지점에 영향을 미칩니다.

- 디렉터리 지원 애플리케이션을 위한 성능.
- 디렉터리 서비스의 가용성.
- 디렉터리 서비스 관리.

디렉터리 토폴로지에는 다음과 같은 주요 의미가 있습니다.

데이터베이스

*데이터베이스*는 복제, 백업 및 데이터 복원과 같은 작업의 기본 단위입니다. 단일 디렉터리를 여러 조각으로 분할하고 별도의 데이터베이스에 할당할 수 있습니다. 그런 다음 이러한 데이터베이스를 서버 간에 배포하여 각 서버의 워크로드를 줄일 수 있습니다. 단일 서버에 둘 이상의 데이터베이스를 저장할 수 있습니다. 예를 들어 하나의 서버에는 세 개의 서로 다른 데이터베이스가 포함될 수 있습니다.

여러 데이터베이스에 대한 자세한 내용은 여러 데이터베이스 사용 정보.

접미사

디렉터리 트리를 여러 데이터베이스로 나눌 때 각 데이터베이스에 *접미사*라는 디렉터리 트리의 일부가 포함되어 있습니다. 예를 들어 한 데이터베이스를 사용하여 디렉터리 트리의 **ou=people,dc=example,dc=com** 접미사(branch)에 항목만 저장할 수 있습니다.

접미사에 대한 자세한 내용은 접미사 정보를 참조하십시오. ???

기술 자료 참조 (referrals 및 chaining)

Directory Server는 다른 데이터베이스에 저장된 디렉터리 데이터를 연결하는 데 필요한 정보 참조 메커니즘(예: 추천 및 연결)을 제공합니다.

추천 및 체인에 대한 자세한 내용은 참조 및 체인 사용을 참조하십시오.

5.2. 디렉터리 데이터 배포

디렉터리 데이터를 분산하면 엔터프라이즈의 각 서버에 대한 디렉터리 항목을 물리적으로 포함하지 않고도 디렉터리를 여러 서버에 확장할 수 있습니다. 따라서 분산 디렉터리는 단일 서버에서 가능한 것보다 훨씬 많은 수의 항목을 보유할 수 있습니다.

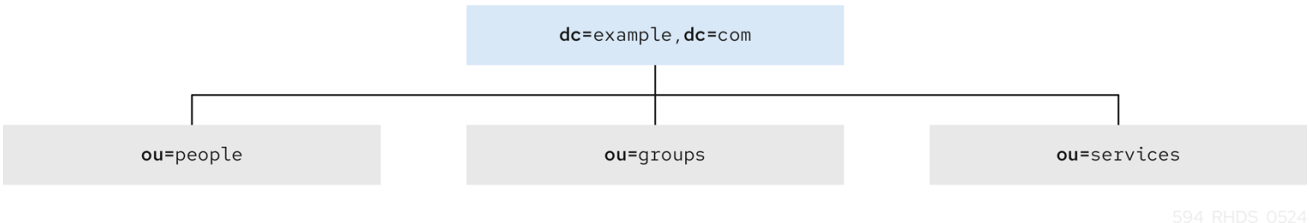
또한 사용자가 배포 세부 정보를 숨기도록 디렉터리를 구성할 수 있습니다.

5.2.1. Directory Server에서 여러 데이터베이스 사용

Directory Server는 Lightning Memory-Mapped Databases(LMDB)에 데이터를 저장합니다. 각 데이터베이스는 할당된 모든 데이터를 포함하는 큰 파일 세트로 구성됩니다.

디렉터리 트리의 다른 부분을 다른 데이터베이스에 저장할 수 있습니다. 예를 들어 디렉터리 트리는 다음과 같은 방식으로 표시될 수 있습니다.

그림 5.1. 디렉터리 트리 예



예제의 디렉터리는 다음 세 개의 하위 요소로 구성됩니다.

- ou=people,dc=example,dc=com

- ou=groups,dc=example,dc=com
- ou=services,dc=example,dc=com

다음과 같은 방식으로 3개의 하위 데이터베이스로 데이터를 저장할 수 있습니다.

그림 5.2. 별도의 데이터베이스에 접미사 데이터 저장

- ou=people,dc=example,dc=com의 DB1
- ou=groups,dc=example,dc=com의 DB2
- ou=services,dc=example,dc=com의 DB3

디렉터리 트리를 여러 데이터베이스로 나눌 때 이러한 데이터베이스를 여러 서버에 배포하여 각 서버의 워크로드를 줄일 수 있습니다. 예를 들어 두 서버(서버A 및 서버 B)에 세 개의 데이터베이스(DB1,DB2 및 DB3)를 저장할 수 있습니다.

그림 5.3. 별도의 서버 간에 접미사 데이터베이스 분할

서버 A 에는 DB1 및 DB2 가 포함되어 있으며 서버 B 에는 DB3 가 포함되어 있습니다.

Directory Server는 전체 디렉터리 서비스를 중지하지 않고 동적으로 데이터베이스 추가를 지원합니다.

5.2.2. Directory Server의 접미사

데이터베이스에는 특정 접미사(디렉터리 트리의 일부)에 대한 데이터가 포함되어 있습니다. Directory Server에서는 루트 접미사 또는 sub-suffix를 만들 수 있습니다.

루트 접미사

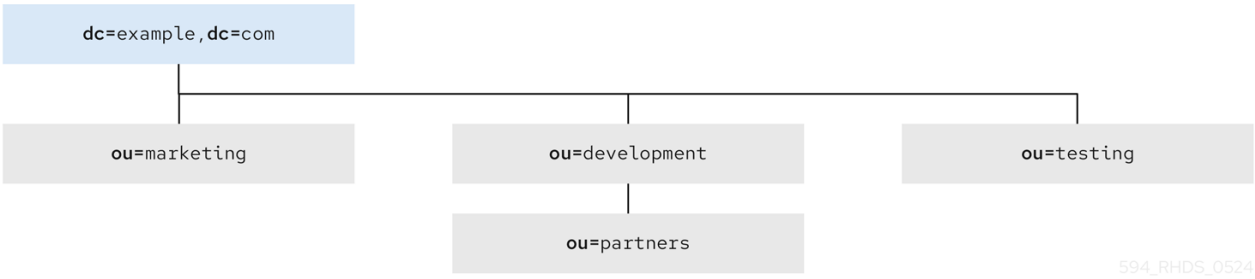
루트 접미사는 트리 상단에 있는 항목입니다. 디렉터리 트리의 루트이거나 디렉터리 서버를 위해 설계된 큰 트리의 일부일 수 있습니다.

sub-suffix

sub-suffix는 루트 접미사 아래에 있는 분기입니다.

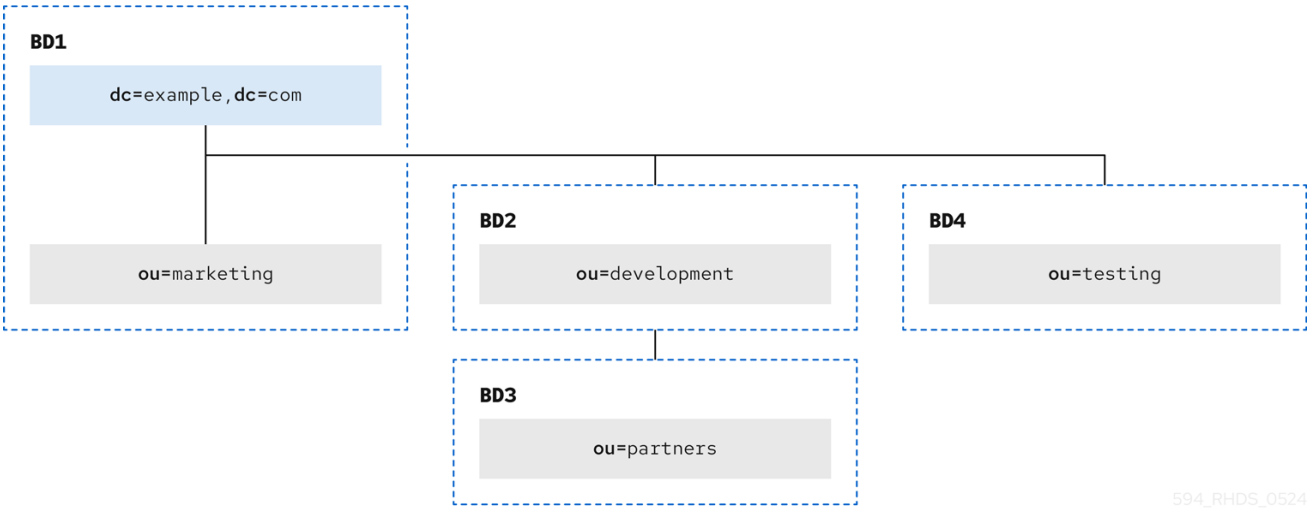
예를 들어 ExampleCom은 접미사를 생성하여 다음과 같은 방식으로 디렉터리 데이터의 배포를 나타냅니다.

그림 5.4. ExampleCom의 디렉터리 트리



ExampleCom은 다음과 같은 방식으로 네 가지 데이터베이스에 디렉터리 트리를 분배합니다.

그림 5.5. 여러 데이터베이스에 디렉터리 트리 분산



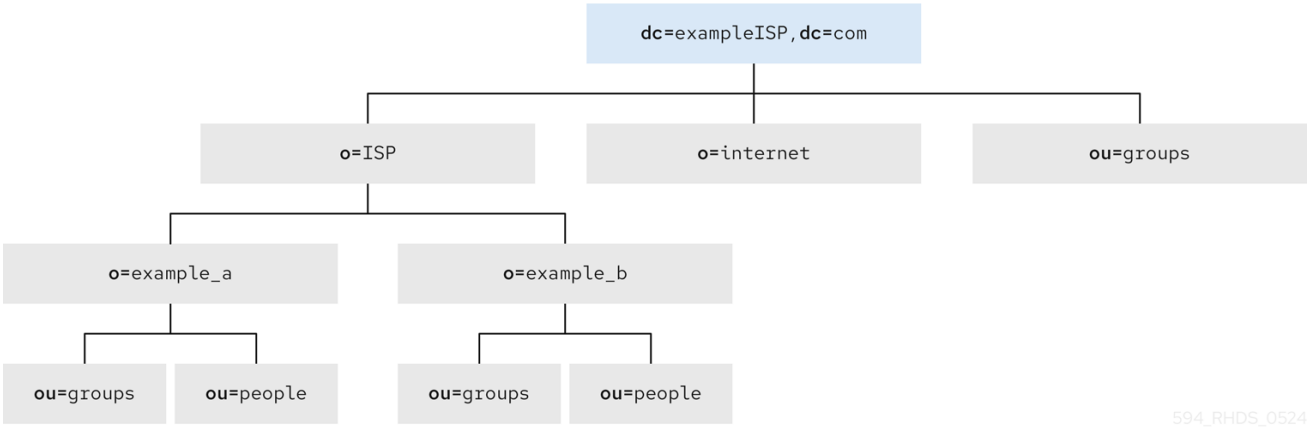
4개의 데이터베이스에는 다음 접미사에 대한 데이터가 포함되어 있습니다.

- 루트 접미사 **dc=example,dc=com**. 이 데이터베이스에는 **dc=example,dc=com** 데이터와 함께 이 데이터베이스에는 원래 디렉터리 트리의 **ou=marketing,dc=example,dc=com** 분기에 대한 데이터가 포함되어 있습니다.
- **ou=testing,dc=example,dc=com sub-suffix.**
- **ou=developer,dc=example,dc=com** 하위 기능.
- **ou=partners,ou=developer,dc=example,dc=com** 하위 기능.

여러 루트 접미사 사용

디렉터리 서비스에는 루트 접미사가 두 개 이상 포함될 수 있습니다. 예를 들어, **Rackspace**라는 이름은 **example_a.com** 과 **example_b.com** 과 같은 여러 웹 사이트를 호스팅합니다. **ExampleISP**에는 다음과 같은 디렉터리 구조가 있습니다.

그림 5.6. 여러 루트 접미사가 있는 디렉터리 트리



Cryostat는 다음 루트 접미사를 생성합니다.

- **DC=exampleISP,dc=com** 과 다음 항목에 대한 데이터가 있습니다.
  - **dc=exampleISP,dc=com**
  - **o=ISP,dc=exampleISP,dc=com**
  - **o=internet,dc=exampleISP,dc=com**
  - **ou=groups,dc=exampleISP,dc=com**
- **O =example_a.com** 과 다음 항목에 대한 데이터가 있습니다.
  - **o=example_a.com,o=ISP,dc=exampleISP,dc=com**
  - **ou=people,o=example_a.com,o=ISP,dc=exampleISP,dc=com**
  - **ou=groups,o=example_a.com,o=ISP,dc=exampleISP,dc=com**

- O =example_b.com 과 다음 항목에 대한 데이터가 있습니다.
  - o=example_b.com,o=ISP,dc=exampleISP,dc=com
  - ou=people,o=example_b.com,o=ISP,dc=exampleISP,dc=com
  - ou=groups,o=example_b.com,o=ISP,dc=exampleISP,dc=com

추가 리소스

- 별도의 데이터베이스에 접미사 저장

5.3. DIRECTORY SERVER의 지식 참조

기술 참조는 분산 데이터 간의 관계를 정의합니다. 기술 참조는 다른 데이터베이스에 보관된 디렉터리 정보에 대한 포인터입니다. Directory Server는 분산 데이터를 단일 디렉터리 트리에 연결하기 위해 다음과 같은 유형의 지식 참조를 제공합니다.

추천

Directory Server는 클라이언트 애플리케이션이 요청을 수행하기 위해 다른 서버에 연결해야 함을 나타내는 정보를 클라이언트 애플리케이션에 반환합니다.

연결

Directory Server는 클라이언트 애플리케이션을 대신하여 다른 서버에 연결하고 작업이 완료되면 결합된 결과를 클라이언트 애플리케이션에 반환합니다.

5.4. DIRECTORY SERVER에서 참조 사용

추천 은 클라이언트 애플리케이션에 요청을 진행하기 위해 연결할 서버를 알려주는 Directory Server에서 반환하는 정보입니다. 이 리디렉션 메커니즘은 클라이언트 애플리케이션이 로컬 서버에 포함되지 않은 디렉터리 항목을 요청할 때 발생합니다.

Directory Server는 다음 유형의 추천을 지원합니다.

기본 추천

클라이언트 애플리케이션이 로컬 트리에 속하지 않는 항목에 대해 요청할 때 디렉터리가 기본 추천을 반환합니다. 서버 및 접미사 수준에서 기본 추천을 구성할 수 있습니다.

스마트 추천

Directory Server는 디렉터리 내의 항목에 대한 스마트 추천을 저장합니다. 스마트 추천은 스마트 추천이 포함된 항목의 DN과 일치하는 하위 트리에 대한 정보가 포함된 서버를 가리킵니다.

Directory Server는 LDAP 균일한 리소스 Cryostat 또는 LDAP URL 형식의 모든 추천을 반환합니다.

추가 리소스

- Directory Server의 기본 참조
- Directory Server의 스마트 참조

5.4.1. LDAP 참조 구조

Directory Server는 LDAP URL 형식의 모든 추천을 반환합니다. LDAP URL에는 다음 정보가 포함되어 있습니다.

- 연결할 서버의 호스트 이름입니다.
- LDAP 요청을 수신 대기하도록 구성된 서버의 포트 번호입니다.
- 기본 DN(검색 작업용) 또는 대상 DN(추가, 삭제 및 수정용)입니다.

예를 들어 클라이언트 애플리케이션은 dc=example,dc=com 분기에서 이름이 Jensen 인 항목을 검색합니다. 그러나 디렉터리 트리의 일부는 유럽 서버에 저장됩니다. 추천은 클라이언트 애플리케이션에 다음 LDAP URL을 반환합니다.

```
ldap://europe.example.com:389/ou=people,l=europe,dc=example,dc=com
```

이 추천은 클라이언트 애플리케이션이 포트 389에서 host europe.example.com 에 연락하고 European branch ou=people,l=europe,dc=example,dc=com 을 통해 새 검색을 제출하도록 지시합니

다.

사용하는 **LDAP** 클라이언트 애플리케이션에 따라 추천 처리 방법이 결정됩니다. 일부 클라이언트 애플리케이션은 참조된 서버에서 작업을 자동으로 재시도합니다. 다른 클라이언트 애플리케이션은 추천 정보를 사용자에게 반환합니다. **Red Hat Directory Server**에서 제공하는 대부분의 **LDAP** 클라이언트 애플리케이션(예: 명령줄 유틸리티)은 자동으로 추천을 따릅니다. **Directory Server**는 초기 디렉터리 요청에 제공된 동일한 바인딩 자격 증명을 사용하여 서버에 액세스합니다.

대부분의 클라이언트 애플리케이션은 제한된 수의 추천 또는 *홉*을 따릅니다. 추천 횟수에 대한 제한은 클라이언트 애플리케이션이 디렉터리 조회 요청을 완료하는 데 걸리는 시간을 줄이고 순환 참조 패턴으로 인한 중단 프로세스를 제거하는 데 도움이 됩니다.

5.4.2. Directory Server의 기본 참조

연결된 서버 또는 데이터베이스에 요청된 데이터가 포함되지 않은 경우 **Directory Server**는 클라이언트에 대한 *기본 추천을* 반환합니다.

예를 들어 클라이언트는 **uid=bjensen,ou=people,dc=example,dc=com** 디렉터리 항목을 요청합니다.

그러나 서버는 **dc=europe,dc=example,dc=com** 접미사 아래에 저장된 항목만 관리합니다. 디렉터리는 **dc=example,dc=com** 접미사 아래에 저장된 항목에 대해 연결할 수 있는 정보를 사용하여 클라이언트에 대한 참조를 반환합니다. 그런 다음 클라이언트는 적절한 서버에 연결하여 원래 요청을 다시 제출합니다.

서버 및 접미사 수준에서 기본 추천을 구성할 수 있습니다.

- 서버 수준 추천을 설정하려면 서버 수준 구성 속성 **nsslapd-referral** 을 사용합니다. **Directory Server**는 **dse.ldif** 구성 파일에 특성 값을 저장합니다. 서버를 사용할 수 없거나 클라이언트가 로컬 서버의 데이터에 액세스할 수 있는 권한이 없는 경우 **Directory Server**는 기본 추천을 반환합니다.

- 접미사 수준 추천을 설정하려면 접미사 구성 속성 **nsslapd-referral** 및 **nsslapd-state** 를 사용합니다. 전체 접미사가 오프라인 상태가 되면 **Directory Server**는 해당 접미사에 대한 클라이언트 요청에 대한 추천을 반환합니다.

5.4.3. Directory Server의 스마트 참조

디렉터리 서버는 기본 참조 외에도 디렉터리 항목 또는 디렉터리 트리를 특정 **LDAP URL**과 연결하는 *스마트 추천을* 지원합니다. 따라서 **Directory Server**는 다음 중 하나로 클라이언트 요청을 전달할 수 있습니다.

- 다른 서버에 포함된 동일한 네임스페이스입니다.
- 다른 서버의 네임스페이스가 다릅니다.
- 동일한 서버의 다른 네임스페이스입니다.

기본 참조와 달리 **Directory Server**는 구성 파일이 아닌 디렉터리에 스마트 참조를 저장합니다.

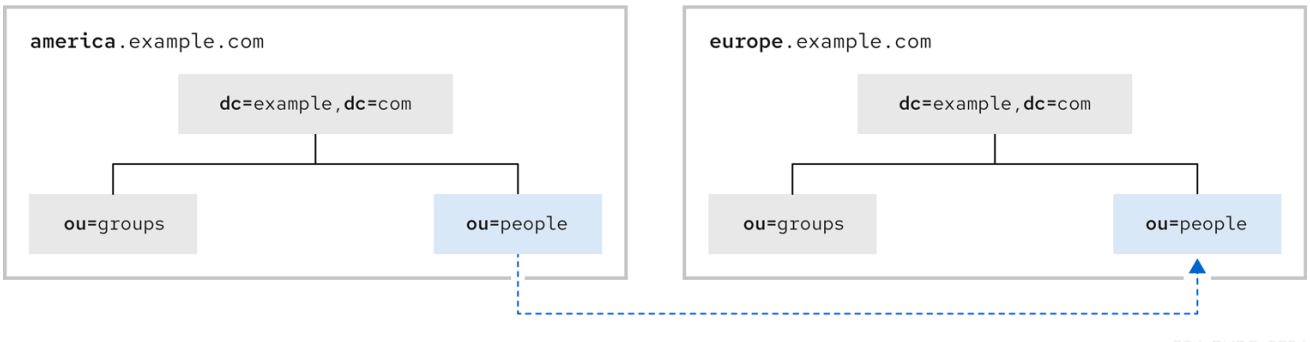
예를 들어, **ExampleCom**의 미국 사무실용 디렉터리에는 **ou=people,dc=example,dc=com** 디렉토리 분기 포인트가 포함되어 있습니다.

이 분기의 요청을 **ExampleCom**의 유럽 사무실의 **ou=people** 분기로 리디렉션하려면 **ou=people** 항목 자체에서 스마트 추천을 지정할 수 있습니다. 스마트 추천에는 다음과 같은 값이 있습니다.

ldap://europe.example.com:389/ou=people,dc=example,dc=com

미국 디렉터리의 **ou=people** 분기에 대한 요청은 다음과 같은 방식으로 유럽 디렉터리로 리디렉션됩니다.

그림 5.7. 스마트 추천을 사용하여 요청 리디렉션



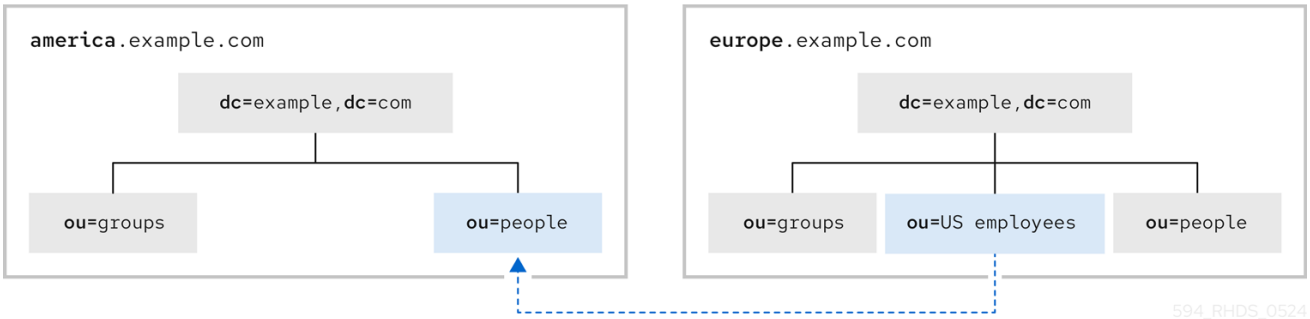
동일한 메커니즘을 사용하여 다른 네임스페이스를 사용하는 다른 서버로 쿼리를 리디렉션할 수 있습니다. 예를 들어, **ExampleCom**의 이탈리아 사무실에서 근무하는 직원은 미국에서 **ExampleCom** 직원의

전화 번호에 대한 유럽 디렉터리 서비스에 요청합니다. **Directory Server**는 다음과 같은 추천을 반환합니다.

ldap://america.example.com:389/ou=people,dc=example,dc=com

다음 다이어그램은 다른 네임스페이스에 대한 참조가 작동하는 방법을 보여줍니다.

그림 **5.8.** 쿼리를 다른 서버 및 네임스페이스로 리디렉션



마지막으로 동일한 서버에서 여러 접미사를 제공할 때 한 네임스페이스의 쿼리를 동일한 서버에서 제공된 다른 네임스페이스로 리디렉션할 수 있습니다. 예를 들어 로컬 서버의 **o=example,c=us** 에 대한 모든 쿼리를 **dc=example,dc=com** 으로 리디렉션하려면 **o=example,c=us** 항목에서 스마트 추천 **ldap:///dc=example,dc=com** 을 설정합니다. **LDAP URL**의 세 번째 슬래시는 **URL**이 동일한 서버를 가리키는 것을 나타냅니다.

참고

한 네임스페이스에서 다른 네임스페이스로의 추천은 검색이 해당 고유 이름을 기반으로 하는 클라이언트에 대해서만 작동합니다. **ou=people,o=example,c=US** 아래의 검색과 같은 기타 작업 유형은 올바르게 수행되지 않습니다.

5.4.4. 스마트 추천 사용 시 고려 사항

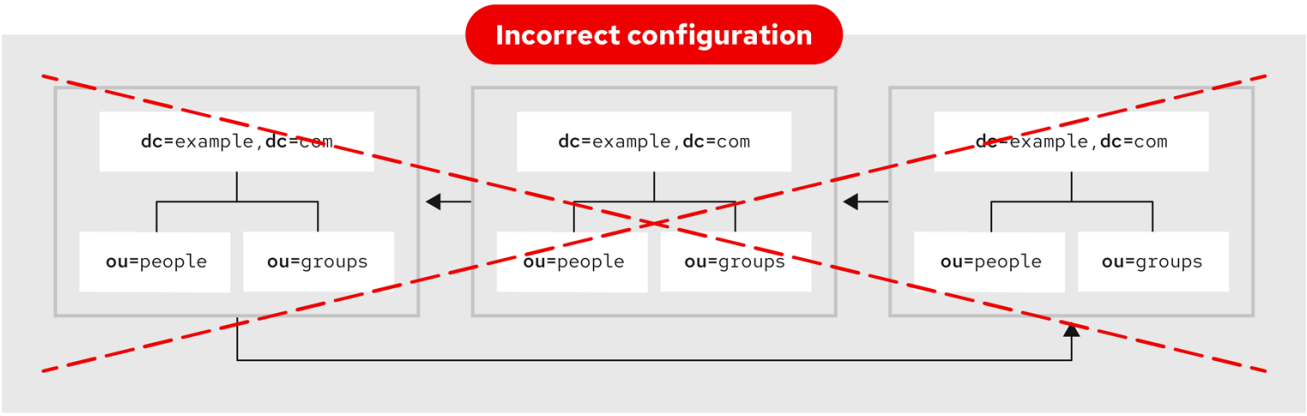
스마트 추천을 사용하기 전에 다음 사항을 고려하십시오.

- 설계를 간단하게 유지합니다.

복잡한 추천 웹은 관리를 어렵게 만듭니다. 스마트 추천 과도한 사용은 또한 순환 추천 패턴을 유발할 수 있습니다. 예를 들어, 추천은 다른 **LDAP URL**을 가리키는 **LDAP URL**을 가리키므

로 체인의 어느 곳에서는 다시 원래 서버를 가리킬 때까지 마찬가지입니다. 다음 다이어그램은 원형 참조 패턴을 보여줍니다.

그림 5.9. 순환 추천 패턴



- 주요 분기 지점에서 리디렉션합니다.

보안을 개선하고 유지 관리 비용을 줄이려면 접미사 및 주요 분기 지점에서 리디렉션을 처리하도록 추천 사용량을 제한합니다. 스마트 추천을 별칭 메커니즘으로 사용하지 마십시오.

- 보안에 미치는 영향을 고려하십시오.

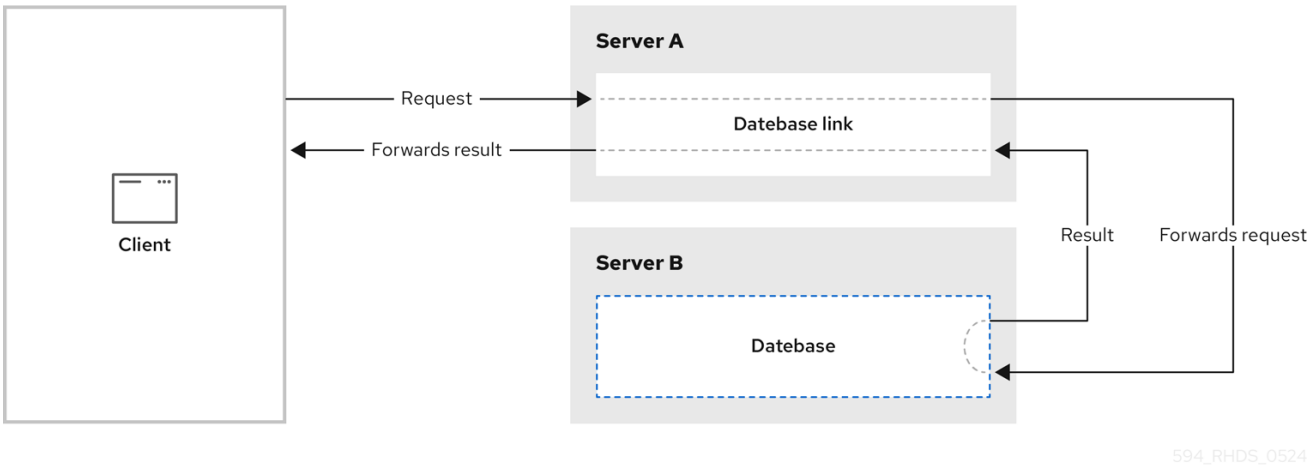
액세스 제어는 추천 경계를 교차하지 않습니다. 요청이 원래 전송된 서버가 항목에 대한 액세스를 허용하는 경우에도 스마트 추천이 다른 서버로 클라이언트 요청을 보내면 클라이언트 애플리케이션이 액세스를 거부할 수 있습니다.

또한 클라이언트 애플리케이션은 클라이언트를 참조하는 서버를 인증하기 위해 자격 증명이 필요합니다.

5.5. 연결 사용

체인은 클라이언트 애플리케이션을 대신하여 다른 서버로 요청을 리디렉션하는 방법입니다. 체인은 서버에서 플러그인으로 구현됩니다. 플러그인은 기본적으로 활성화되어 있습니다. 이 플러그인을 사용하여 원격으로 저장된 데이터를 가리키는 데이터베이스 링크, 특수 항목을 만듭니다. 클라이언트 애플리케이션이 데이터베이스 링크에서 데이터를 요청하면 데이터베이스 링크가 원격 데이터베이스에서 데이터를 검색하여 클라이언트로 반환합니다.

그림 5.10. 연결 을 사용하여 클라이언트 요청을 서버로 전송



각 데이터베이스 링크는 데이터를 보유하는 원격 서버와 연결됩니다. 오류가 발생할 때 사용할 데이터베이스 링크의 데이터 복제본을 포함하는 대체 원격 서버를 구성할 수도 있습니다.

데이터베이스 링크 구성에 대한 자세한 내용은 데이터베이스 링크 생성 및 유지 관리를 참조하십시오.

데이터베이스 링크는 다음과 같은 기능을 제공합니다.

- 원격 데이터에 대한 보이지 않는 액세스

  데이터베이스 링크는 클라이언트 요청을 해결하고 클라이언트에서 데이터 배포를 완전히 숨깁니다.

- 동적 관리

  시스템에서 디렉터리의 일부를 추가하거나 제거할 수 있지만 전체 시스템은 클라이언트 애플리케이션에서 계속 사용할 수 있습니다. 디렉터리에서 항목을 재배포할 때까지 데이터베이스 링크를 사용하여 애플리케이션에 대한 추천을 일시적으로 반환할 수 있습니다.

  클라이언트 애플리케이션을 데이터베이스로 전달하는 대신 추천을 반환하는 접미사를 사용하여 이를 구현할 수도 있습니다.

- 액세스 제어

데이터베이스 링크는 클라이언트 애플리케이션을 가장하여 원격 서버에 적절한 권한 부여 ID를 제공합니다. 액세스 제어 평가가 필요하지 않은 경우 원격 서버에서 사용자 가장을 비활성화할 수 있습니다.

데이터베이스 링크 및 액세스 제어 평가에 대한 자세한 내용은 데이터베이스 링크 및 액세스 제어 평가를 참조하십시오.

5.6. 추천 및 체인 간 결정

디렉터리의 특정 요구 사항에 따라 추천과 체인 중에서 선택합니다.

- 연결로 인해 서버 복잡성 증가 시 클라이언트의 복잡성이 감소합니다. 그러나 연결 기능을 사용하면 클라이언트 애플리케이션이 단일 서버와 상호 작용하고 여러 서버에 저장된 데이터에 계속 액세스할 수 있습니다. 클라이언트 애플리케이션은 요청이 연결되는 서버를 인증할 필요가 없습니다.

- 추천 기능을 사용하면 클라이언트 애플리케이션이 추천을 찾고 검색 결과를 다시 제출해야 합니다. 클라이언트는 참조된 서버에 올바르게 인증할 수도 있어야 합니다.

  또한 회사 네트워크에서 프록시를 사용하는 경우 추천이 실패하는 경우가 있습니다. 예를 들어 클라이언트 애플리케이션에는 방화벽 내의 하나의 서버와만 통신할 수 있는 권한이 있을 수 있습니다. 해당 애플리케이션을 다른 서버를 참조하면 애플리케이션에 연결할 수 없습니다.

  그러나 추천은 클라이언트 애플리케이션 작성자에게 더 많은 유연성을 제공하며 개발자는 분산 디렉터리 작업의 진행 상황에 대해 사용자에게 더 나은 피드백을 제공할 수 있습니다.

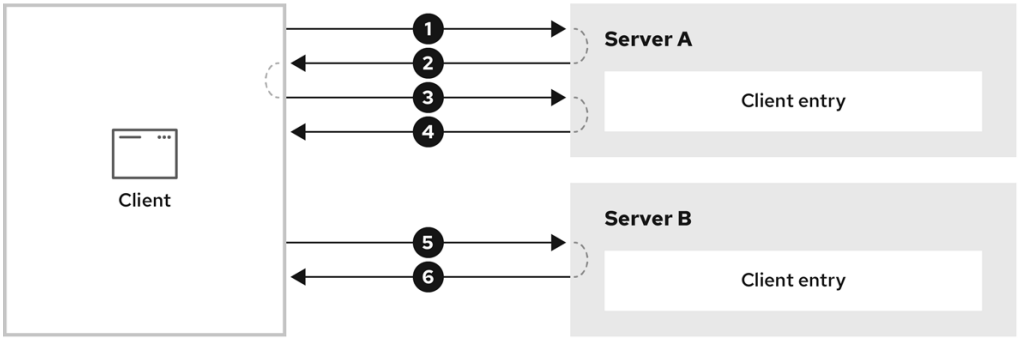
5.6.1. 액세스 제어 평가

체인은 추천과 다르게 액세스 제어를 평가합니다. 추천을 사용하면 클라이언트 항목(바인드 DN)이 모든 대상 서버에 있어야 합니다. 연결에서 클라이언트 항목이 모든 대상 서버에 있을 필요는 없습니다.

5.6.1.1. 추천을 사용하여 검색 요청 수행

다음 다이어그램에서는 추천을 사용하여 서버에 대한 클라이언트 요청을 보여줍니다.

그림 5.11. 추천을 사용하여 클라이언트 요청을 서버로 전송



검색 요청이 다음과 같은 방식으로 수행됩니다.

1. 클라이언트 애플리케이션은 먼저 서버 **A** 와 바인딩합니다.

2. 서버 **A** 에는 사용자 이름과 암호를 제공하는 클라이언트에 대한 항목이 포함되어 있어 바인딩 수락 메시지를 반환합니다. 추천이 작동하려면 클라이언트 항목이 서버 **A** 에 있어야 합니다.

3. 클라이언트 애플리케이션에서 서버 **A** 로 작업 요청을 보냅니다.

4. 그러나 서버 **A** 에는 요청된 정보가 포함되어 있지 않습니다. 대신 **Server A** 는 서버 **B**에 연결하도록 지시하는 클라이언트 애플리케이션에 대한 참조를 반환합니다.

5. 그러면 클라이언트 애플리케이션이 서버 **B** 로 **bind** 요청을 보냅니다. 성공적으로 바인딩하려면 서버 **B** 에도 클라이언트 애플리케이션에 대한 항목도 포함되어야 합니다.

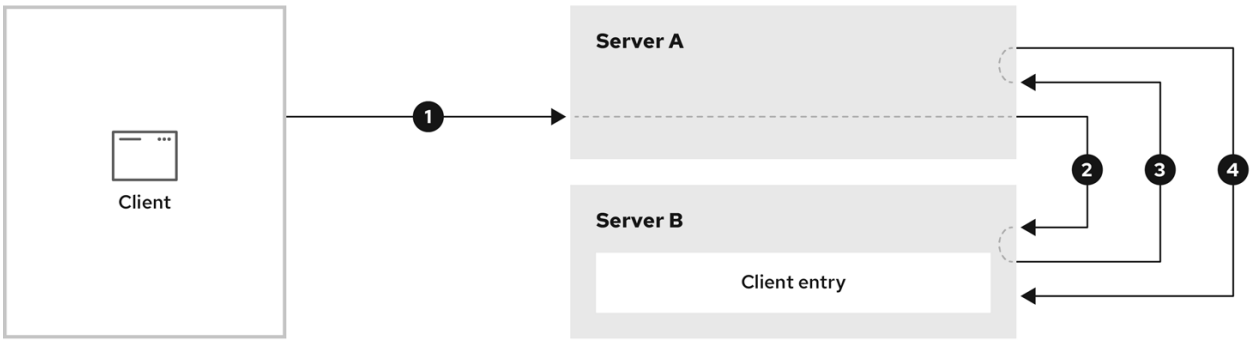
6. 바인딩에 성공하고 클라이언트 애플리케이션에서 검색 작업을 **Server B** 로 다시 제출할 수 있습니다.

이 방법을 사용하려면 서버 **B** 에 서버 **A** 에서 클라이언트 항목의 복제 사본이 있어야 합니다.

5.6.1.2. 체인을 사용하여 검색 요청 수행

연결을 통해 서버 간에 클라이언트 항목을 복제하는 문제를 해결할 수 있습니다.

그림 5.12. 연결 을 사용하여 클라이언트 요청을 서버로 전송



검색 요청이 다음과 같은 방식으로 수행됩니다.

1. 클라이언트 애플리케이션이 서버 **A** 와 바인딩되고 서버 **A** 는 사용자 이름과 암호가 올바른지 확인합니다.

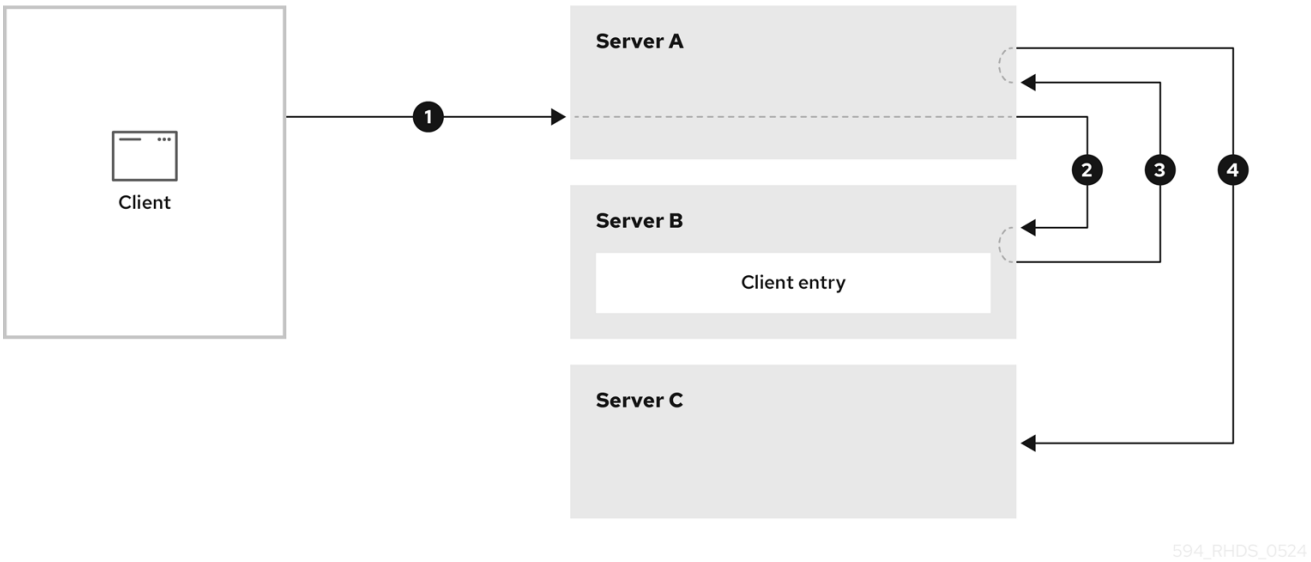
2. 서버 **A** 에는 클라이언트 애플리케이션에 해당하는 항목이 포함되어 있지 않습니다. 대신 클라이언트의 실제 항목이 포함된 서버 **B** 에 대한 데이터베이스 링크가 포함되어 있습니다. 서버 **A** 는 서버 **B** 에 바인딩 요청을 보냅니다.

3. 서버 **B** 는 서버 **A** 에 수락 응답을 보냅니다.

4. 서버 **A** 는 데이터베이스 링크를 사용하여 클라이언트 애플리케이션 요청을 처리합니다. 데이터베이스 링크는 서버 **B** 에 있는 원격 데이터 저장소에 연결하여 검색 작업을 처리합니다.

연결된 시스템에서 클라이언트 애플리케이션에 해당하는 항목을 클라이언트가 요청하는 데이터와 동일한 서버에 배치할 필요가 없습니다. 다음 다이어그램은 클라이언트 검색 요청을 완료하는 데 두 개의 체인 서버를 사용하는 방법에 대해 설명합니다.

그림 5.13. 두 개의 다른 서버를 사용하여 클라이언트 인증 및 데이터 검색



검색 요청이 다음과 같은 방식으로 수행됩니다.

1. 클라이언트 애플리케이션이 서버 **A** 에 바인딩되고 서버 **A** 는 사용자 이름과 암호가 올바른지 확인합니다.

2. 서버 **A** 에는 클라이언트 애플리케이션에 해당하는 항목이 포함되어 있지 않습니다. 대신 클라이언트의 실제 항목이 포함된 서버 **B** 에 대한 데이터베이스 링크가 포함되어 있습니다. 서버 **A** 는 서버 **B** 에 바인딩 요청을 보냅니다.

3. 서버 **B** 는 서버 **A** 에 수락 응답을 보냅니다.

4. 그러면 서버 **A** 가 다른 데이터베이스 링크를 사용하여 클라이언트 요청을 처리합니다. 데이터베이스 링크는 검색 작업을 처리하기 위해 서버 **C** 에 있는 원격 데이터 저장소에 연결합니다.

5.6.1.3. 지원되지 않는 액세스 제어

데이터베이스 링크는 다음 액세스 제어를 지원하지 않습니다.

- 사용자 항목이 다른 서버에 있는 경우 사용자 항목의 콘텐츠에 액세스해야 하는 컨트롤입니다. 여기에는 그룹, 필터 및 역할을 기반으로 하는 액세스 제어가 포함됩니다.

-

클라이언트 IP 주소 또는 DNS 도메인을 기반으로 하는 제어는 거부될 수 있습니다. 이는 데이터베이스 링크가 원격 서버에 연결할 때 클라이언트를 가장하기 때문입니다. 원격 데이터베이스에 IP 기반 액세스 제어가 포함된 경우 원래 클라이언트 도메인이 아닌 데이터베이스 링크 도메인을 사용하여 평가합니다.

5.7. 인덱스를 사용하여 데이터베이스 성능 개선

데이터베이스 크기에 따라 클라이언트 애플리케이션에서 수행하는 검색에는 많은 시간과 리소스가 걸릴 수 있습니다. 따라서 검색 성능을 개선하기 위해 인덱스를 사용할 수 있습니다.

인덱스는 데이터베이스가 저장하는 파일입니다. 디렉터리의 각 데이터베이스에 대해 별도의 인덱스 파일이 유지 관리됩니다. 각 파일의 이름은 인덱스에 따라 이름이 지정됩니다. 특정 속성에 대한 인덱스 파일에는 여러 유형의 인덱스가 포함될 수 있습니다. 예를 들어 cn.db 라는 파일에는 공통 이름(cn) 속성에 대한 모든 인덱스가 포함되어 있습니다.

디렉터리를 사용하는 애플리케이션 유형에 따라 다양한 유형의 인덱스를 사용합니다. 다른 애플리케이션에서는 특정 속성을 자주 검색하거나 다른 언어로 디렉터리를 검색하거나 특정 형식의 데이터가 필요할 수 있습니다.

5.7.1. 디렉터리 인덱스 유형 개요

Directory Server는 다음 인덱스 유형을 지원합니다.

존재 인덱스

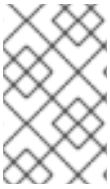
uid 와 같은 특정 속성을 보유한 항목을 나열합니다.

같음 인덱스

cn=Babs Jensen 과 같은 특정 특성 값이 포함된 항목을 나열합니다.

대략적인 인덱스

대략적인(또는 "sounds-like") 검색을 허용합니다. 예를 들어 항목에 cn=Babs L. Jensen 속성 값이 포함될 수 있습니다. 대략적인 검색에서는 cn~=Babs Jensen,cn~=Babs, cn~=Jensen 에 대한 검색 값을 반환합니다.



참고

대략적인 인덱스는 ASCII 문자를 사용하여 이름을 영어로 작성해야 합니다.

하위 문자열 인덱스

항목 내의 하위 문자열을 검색할 수 있습니다. 예를 들어 **cn=*derson** 검색은 **bill Anderson, Norma Henderson** 및 이 문자열을 포함하는 **Sanderson**과 같은 공통 이름과 일치합니다.

국제 인덱스

국제 디렉터리의 정보 검색 성능을 향상시킵니다. 로케일(국제 **OID**)을 인덱싱하는 특성과 연결하여 일치 규칙을 적용하도록 인덱스를 구성할 수 있습니다.

인덱스 또는 **VLV**(가상 목록 보기) 인덱스 검색

웹 콘솔의 항목 표시 성능을 향상시킵니다. 디렉터리 트리 분기에 *검색 인덱스*를 생성하여 표시 성능을 향상시킬 수 있습니다.

추가 리소스

- 인덱스 관리

5.7.2. 인덱싱 비용 평가

인덱스를 사용하여 검색 성능을 향상시킬 때 다음 사항을 고려하십시오.

- 인덱스를 사용하면 항목을 수정하는 데 걸리는 시간이 증가합니다.

  유지 관리하는 인덱스가 많을수록 데이터베이스를 업데이트하는 데 더 오래 걸립니다.

- 인덱스 파일은 디스크 공간을 사용합니다.

  더 많은 속성을 인덱싱할수록 더 많은 파일을 만들 수 있습니다. 또한 긴 문자열이 포함된 특성에 대한 대략적인 및 하위 문자열 인덱스를 만드는 경우 인덱스 파일이 빠르게 증가할 수 있습니다.

- 인덱스 파일은 메모리를 사용합니다.

  더 효율적으로 실행하기 위해 **Directory Server**는 가능한 한 많은 인덱스 파일을 메모리에 넣습니다. 인덱스 파일은 데이터베이스 캐시 크기에 따라 사용 가능한 풀에서 메모리를 사용합니다. 많은 인덱스 파일을 사용하려면 더 큰 데이터베이스 캐시가 필요합니다.

- 인덱스 파일을 만드는 데 시간이 걸립니다.

  인덱스 파일은 검색 중에 시간을 절약하지만 불필요한 인덱스를 유지하면 시간이 소모될 수 있습니다. 디렉터리를 사용할 때 클라이언트 애플리케이션에 필요한 파일만 유지합니다.

6장. 복제 프로세스 설계

디렉터리 정보를 복제하면 디렉터리의 가용성과 성능이 향상됩니다. 복제 프로세스를 설계하여 필요한 시기와 위치를 데이터를 사용할 수 있는지 확인합니다.

6.1. 복제 소개

복제는 한 디렉터리 서버에서 다른 **Directory** 서버로 디렉터리 데이터를 자동으로 복사하는 메커니즘입니다. 복제를 사용하면 자체 데이터베이스(복제본)에 저장된 디렉터리 트리 또는 하위 트리를 서버 간에 복사할 수 있습니다. 정보의 주요 사본을 보유한 서버는 모든 업데이트를 모든 복제본에 자동으로 복사합니다.

복제는 고가용성 디렉터리 서비스를 제공하며 지리적으로 데이터를 배포할 수 있습니다. 다음은 복제 이점 목록입니다.

- 내결함성 및 페일오버

  디렉터리 트리를 여러 서버에 복제하면 하드웨어, 소프트웨어 또는 네트워크 문제로 인해 클라이언트 애플리케이션이 특정 **Directory Server**에 액세스할 수 없는 경우에도 디렉터리를 사용할 수 있습니다. 클라이언트는 읽기 및 쓰기 작업을 위해 다른 **Directory Server**라고 합니다.

  참고

  추가,수정, 삭제 작업을 위한 장애 조치(failover)는 다중 제공 복제 를 사용하는 경우에만 가능합니다.

- 로드 밸런싱

  서버 간에 디렉터리 트리를 복제하면 지정된 서버의 액세스 로드가 줄어들어 서버 응답 시간이 단축됩니다.

- 성능 향상

  사용자에게 가까운 위치에 디렉터리 항목을 복제하면 **Directory Server** 성능이 향상됩니다.

- 로컬 데이터 관리

  복제를 사용하면 엔터프라이즈 전체의 다른 **Directory Server**와 공유하는 동안 로컬에서 정보를 소유하고 관리할 수 있습니다.

6.1.1. 복제 개념

복제 구현을 고려할 때 다음과 같은 기본 질문에 대답하십시오.

- 어떤 정보를 복제해야 합니까?
- 어떤 서버가 해당 정보의 주요 사본 또는 공급자 복제를 보유하고 있습니까?
- 해당 정보의 읽기 전용 복사 또는 소비자 복제본을 보유하는 서버는 무엇입니까.
- 소비자 복제본이 클라이언트 애플리케이션에서 수정 요청을 수신하면 어떻게 됩니까? 어떤 서버로 요청을 리디렉션해야 합니까?

Directory Server가 복제를 구현하는 방법에 대한 이해를 제공하는 개념에 대해 알아봅니다.

- replica
- 복제 단위
- 공급업체 및 소비자
- 변경 로그
- 복제 계약

6.1.1.1. replica

복제본 은 복제에 참여하는 데이터베이스입니다. Directory Server는 다음 유형의 복제본을 지원합니다.

vendor replica (read-write)

디렉터리 데이터의 주요 복사본이 포함된 읽기-쓰기 데이터베이스입니다. 공급자 복제본 프로세스만 디렉터리 클라이언트의 요청을 수정합니다.

소비자 복제본(읽기 전용)

공급자 복제본에 보관된 정보의 다른 복사본을 포함하는 읽기 전용 데이터베이스입니다.A read-only database that contains another copy of the information held on the provider replica. 소비자 복제본은 디렉터리 클라이언트의 검색 요청을 처리할 수 있지만 공급자 복제본에 대한 수정 요청을 나타냅니다.

Directory Server는 복제에서 다양한 역할을 가진 여러 데이터베이스를 관리할 수 있습니다. 예를 들어 공급자 복제본에 **dc=accounting,dc=example,dc=com** 접미사를 저장하고 소비자 복제본에 **dc=sales,dc=example,dc=com** 접미사를 사용할 수 있습니다.

6.1.1.2. 복제 단위

가장 작은 복제 단위는 접미사(네임스페이스)입니다. 복제 메커니즘을 사용하려면 하나의 접미사가 하나의 데이터베이스에 일치해야 합니다. Directory Server는 사용자 지정 배포 논리를 사용하여 두 개 이상의 데이터베이스를 통해 배포되는 접미사를 복제할 수 없습니다.

6.1.1.3. 공급업체 및 소비자

vendor server

공급자 서버는 다른 서버에 업데이트를 복제하는 서버입니다. 공급자 서버는 각 업데이트 작업의 레코드를 포함하는 변경 로그를 유지 관리합니다.

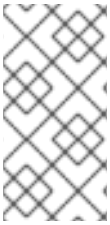
소비자 서버

소비자 서버는 다른 서버에서 업데이트를 수신하는 서버입니다.

서버는 다음과 같은 상황에서 공급자와 소비자의 역할을 동시에 수행할 수 있습니다.

- 계단식 복제에서 일부 서버가 *허브 서버의* 역할을 수행하는 경우 . 자세한 내용은 오스 케이딩 복제를 참조하십시오.
-

다중 제공 복제에서는 여러 서버가 공급업체 읽기-쓰기 복제본을 관리하는 경우입니다. 각 서버는 다른 서버에서 업데이트를 전송하고 수신합니다. 자세한 내용은 **Multi-supplier 복제** 를 참조하십시오.



참고

Red Hat Directory Server에서 공급업체 서버는 항상 사용자가 아닌 복제를 시작합니다.

공급자 서버는 다음 작업을 수행해야 합니다.

- 디렉터리 클라이언트의 읽기 요청 및 업데이트 요청에 응답합니다.
- 복제본의 상태 정보 및 변경 로그를 유지 관리합니다. 공급자 서버는 항상 자신이 관리하는 읽기-쓰기 복제본에 대한 변경 사항을 기록합니다. 이렇게 하면 모든 변경 사항이 소비자 서버에 복제됩니다.
- 소비자 서버에 대한 복제를 시작합니다.

소비자 서버는 다음 작업을 수행해야 합니다.

- 읽기 요청에 응답합니다.
- 복제본에 대한 공급자 서버에 대한 업데이트 요청을 참조하십시오. 소비자 서버가 항목을 추가, 삭제 또는 변경하라는 요청을 수신하면 해당 요청을 공급자 서버라고 합니다. 그런 다음 공급자 서버는 요청을 수행하고 이러한 변경 사항을 복제합니다.

캐스케이딩 복제 특수한 경우 허브 서버는 다음 작업을 수행합니다.

- 읽기 요청에 응답합니다.
- 공급자 서버에 대한 업데이트 요청을 참조하십시오.

- 소비자 서버에 대한 복제를 시작합니다.

6.1.1.4. 변경 로그

모든 공급자 서버는 *변경 로그*를 유지합니다. 변경 로그는 공급자 복제본에서 발생한 수정 사항에 대한 레코드입니다. 공급자 서버는 이러한 수정 사항을 다른 서버에 저장된 복제본에 푸시합니다.

항목이 추가, 수정 또는 삭제되면 **Directory Server**는 변경 로그 파일에 수행된 **LDAP** 작업을 기록합니다.

changelog는 서버의 내부 용도로만 사용됩니다. 변경 로그를 읽어야 하는 애플리케이션이 있는 경우 이전 버전과의 호환성을 위해 **Retro Changelog** 플러그인을 사용해야 합니다.

변경 로그 속성에 대한 자세한 내용은 **cn=changelog,cn=database_name,cn=ldbm** 데이터베이스,**cn=plugins,cn=config** 아래의 데이터베이스 속성을 참조하십시오.

6.1.1.5. 복제 계약

서버는 복제 계약을 사용하여 두 서버 간에 복제를 수행하는 방법을 정의합니다. 복제 계약에는 *하나의 공급업체와 하나의* 소비자 간 복제가 설명되어 있습니다. 본 약관은 공급자 서버에 구성되며 다음 정보를 식별합니다.

- 복제할 데이터베이스입니다.
- 데이터가 푸시되는 소비자 서버입니다.
- 복제가 발생할 수 있는 시간입니다.
- 공급자 서버가 *복제 관리자* 항목 또는 *공급업체 바인딩 DN*이라는 소비자에 바인딩하는 데 *사용해야 하는 DN 및 인증 정보입니다*.
- 연결 보안 방법(예: **TLS**, **StartTLS**, 클라이언트 인증, **SASL** 또는 간단한 인증)
-

복제할 속성입니다. 소수 복제에 대한 자세한 내용은 Fractional replication 을 참조하십시오.

6.1.2. 데이터 일관성

데이터 일관성은 복제된 데이터베이스 내용이 지정된 시간에 서로 얼마나 밀접하게 일치하는지 나타냅니다. 공급자는 소비자를 업데이트해야 하는 시기를 결정하고 복제를 시작합니다. 소비자를 초기화한 후에만 복제를 시작할 수 있습니다.

Directory Server는 항상 특정 요일 또는 요일에 대한 복제본을 동기화된 상태로 유지하거나 업데이트를 예약할 수 있습니다.

지속적으로 동기화된 복제본

지속적으로 동기화된 복제본은 데이터 일관성을 개선하지만 자주 업데이트되므로 네트워크 트래픽을 늘립니다.

다음과 같은 경우 지속적으로 동기화된 복제본을 사용합니다.

- 서버 간에 안정적인 고속 연결이 가능합니다.
- 클라이언트 애플리케이션은 주로 검색을 전송,읽기, 디렉터리 서버와 비교하여 몇 가지 업데이트 작업만 보냅니다.

소비자 업데이트 예약

디렉터리에 낮은 수준의 데이터 일관성이 있을 수 있고 네트워크 트래픽에 미치는 영향을 낮추려는 경우 업데이트를 예약하도록 선택합니다.

다음의 경우 예약된 업데이트를 사용합니다.

- 신뢰할 수 없거나 주기적으로 사용 가능한 네트워크 연결이 있습니다.
- 클라이언트 애플리케이션은 주로 Directory Server에 추가 및 수정 작업을 보냅니다.

- 연결 비용을 줄여야 합니다.

다중 제공 복제의 데이터 일관성

다중 제공 복제가 있는 경우, 각 공급 업체마다 복제본이 지속적으로 동기화되어도 저장된 데이터에 차이가 있을 수 있으므로 각 공급 업체마다 일관적인 복제본이 있습니다.

느슨한 일관성의 주요 이유는 다음과 같습니다.

- 공급업체 간의 수정 작업 전파에는 대기 시간이 있습니다.
- 수정 작업을 서비스한 공급자는 "운영 성공" 메시지를 고객에게 반환하기 전에 두 번째 공급 업체가 유효성을 검증할 때까지 기다리지 않습니다.

6.2. 일반적인 복제 시나리오

다음 일반적인 시나리오를 사용하여 필요에 가장 적합한 복제 토폴로지를 빌드할 수 있습니다.

- 단일 공급 업체 복제
- 다중 제공 복제
- 계단식 복제
- 혼합 환경

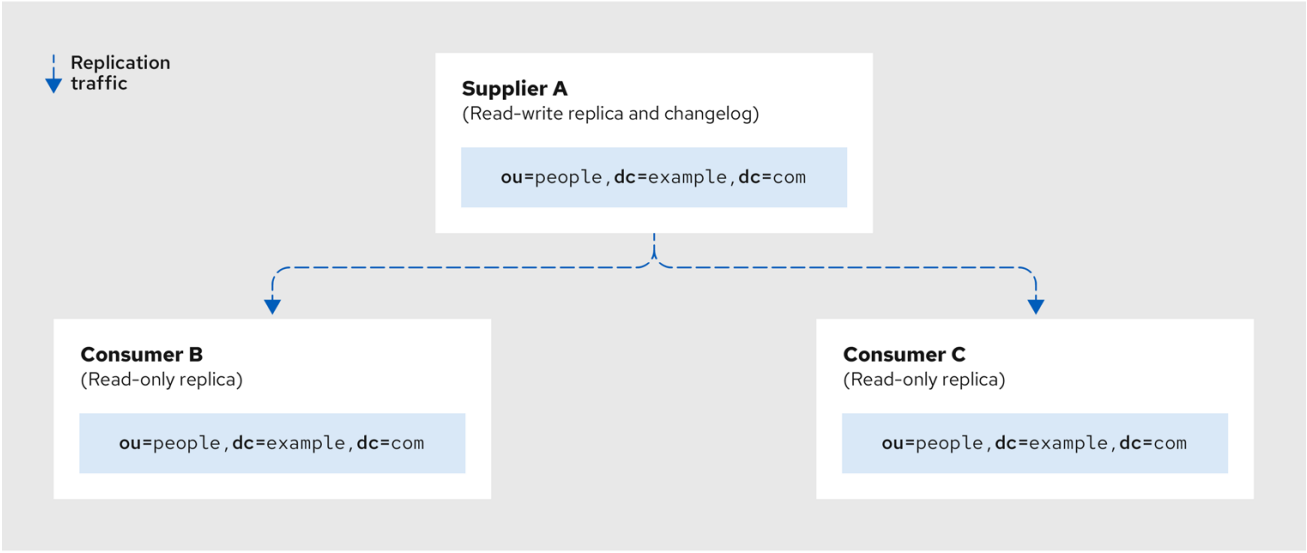
6.2.1. 단일 공급 업체 복제

단일 제공 복제 시나리오에서는 공급자 서버가 디렉토리 데이터(읽기-쓰기 복제본)의 주요 복사본을 유지 관리하고 이 데이터의 업데이트를 하나 이상의 소비자 서버로 보냅니다. 모든 디렉터리 수정은 공급자 서버의 읽기-쓰기 복제본에서 수행되며 소비자 서버에는 데이터의 읽기 전용 복제본이 포함됩니다.

공급자 서버는 공급자 복제본의 모든 변경 사항을 기록하는 변경 로그를 유지합니다.

다음 다이어그램은 단일 공급 업체 복제 시나리오를 보여줍니다.

그림 6.1. 단일 공급 업체 복제



단일 공급자 서버가 관리할 수 있는 총 소비자 서버 수는 네트워크의 속도와 매일 수정된 총 항목 수에 따라 달라집니다.

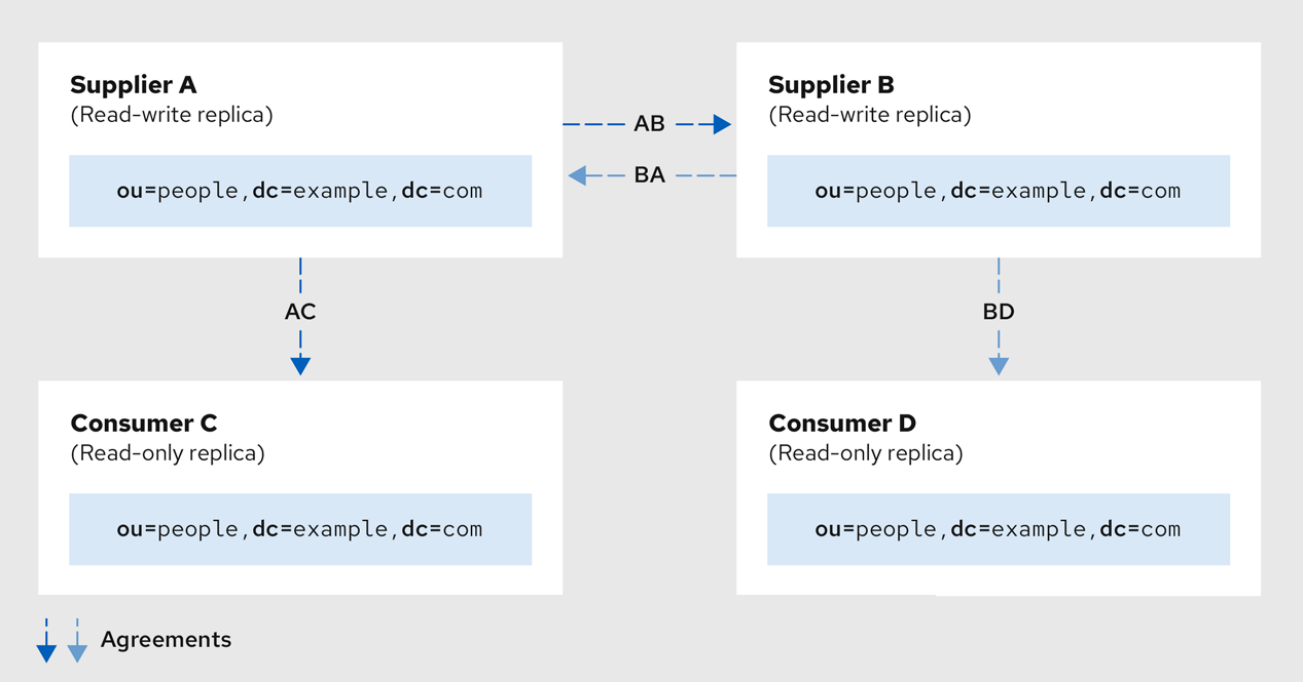
6.2.2. 다중 제공 복제

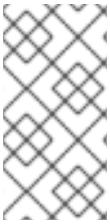
다중 제공 복제 환경에서는 동일한 정보의 주요 사본이 여러 서버에 존재할 수 있으며 디렉터리 데이터를 다른 위치에서 동시에 업데이트할 수 있습니다. 각 서버에서 발생하는 변경 사항은 다른 서버에 복제되므로 각 서버가 공급업체와 소비자 역할을 합니다.

여러 서버에서 동일한 데이터를 수정하면 복제 충돌이 발생합니다. 충돌 해결 절차를 사용하여 Directory Server는 최근 변경 사항을 유효한 작업으로 사용합니다.

멀티 공급 업체 환경에서 각 공급 업체는 소비자와 다른 공급 업체를 가리키는 복제 계약이 있어야합니다. 예를 들어 두 공급업체, provider A 및 provider B 와 소비자 C 및 소비자 D 두 개로 복제를 구성합니다. 또한 하나의 공급업체가 하나의 소비자만 업데이트한다고 결정합니다. 공급업체 A에서 공급업체 B 및 소비자 C 를 가리키는 복제 계약을 생성합니다. 공급업체 B에서 공급업체 A 및 소비자 D 를 가리키는 복제 계약을 생성합니다. 다음 다이어그램에서는 복제 계약을 보여줍니다.

그림 6.2. 두 공급업체를 통한 멀티 공급 업체 복제





참고

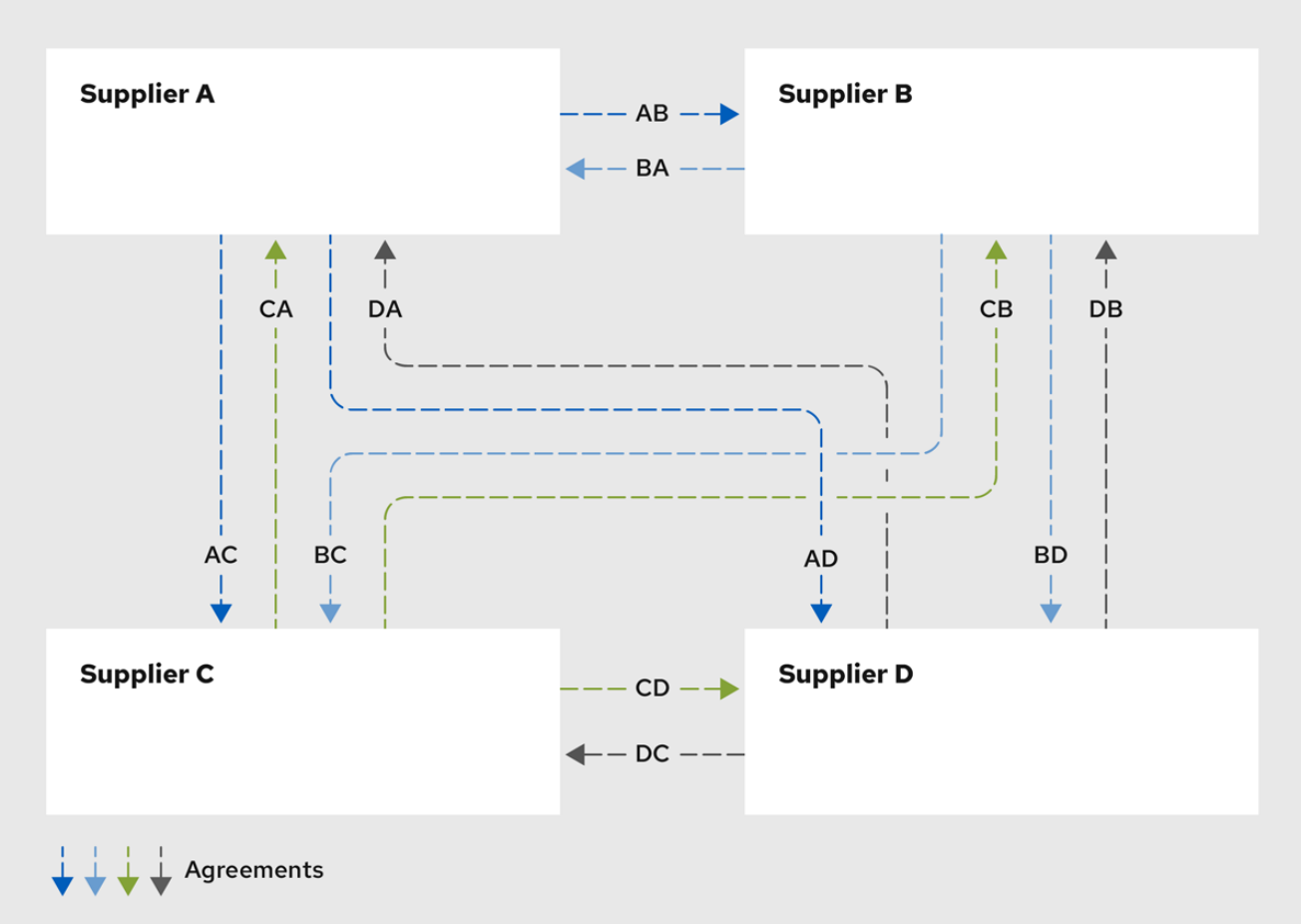
Red Hat Directory Server는 모든 복제 환경에서 최대 **20**개의 공급자 서버와 무제한의 허브 및 소비자 서버를 지원합니다.

많은 공급업체를 사용하려면 다양한 복제 계약을 생성해야 합니다. 또한 각 공급업체는 서로 다른 토폴로지로 구성할 수 있으므로 디렉터리 서버 환경에 **20**개의 디렉토리 트리와 스키마 차이가 있을 수 있습니다. 다른 많은 변수는 토폴로지 선택에 직접적인 영향을 미칠 수 있습니다.

공급업체는 다른 모든 공급 업체 또는 일부 공급 업체에 업데이트를 보낼 수 있습니다. 업데이트가 모든 공급업체에 전송되면 변경 사항이 더 빨라지고 전체 시나리오가 장애 허용성이 향상됩니다. 그러나 공급업체 구성의 복잡성이 증가하고 높은 네트워크 및 높은 서버 수요를 도입합니다. 공급업체의 하위 집합에 업데이트를 보내는 것은 네트워크 및 서버 로드를 구성하고 줄이는 것이 훨씬 쉽지만 여러 서버 오류가 발생하면 데이터 손실 위험이 증가합니다.

완전히 연결된 메시 토폴로지

다음 다이어그램은 **4**개의 공급업체 서버가 다른 모든 공급자 서버에 데이터를 복제하는 완전히 연결된 메시 토폴로지를 보여줍니다. 하나의 복제 계약에는 하나의 공급업체와 하나의 소비자 간의 관계를 설명하기 때문에 총 **4**개의 공급자 서버 간에 복제 계약이 존재합니다.

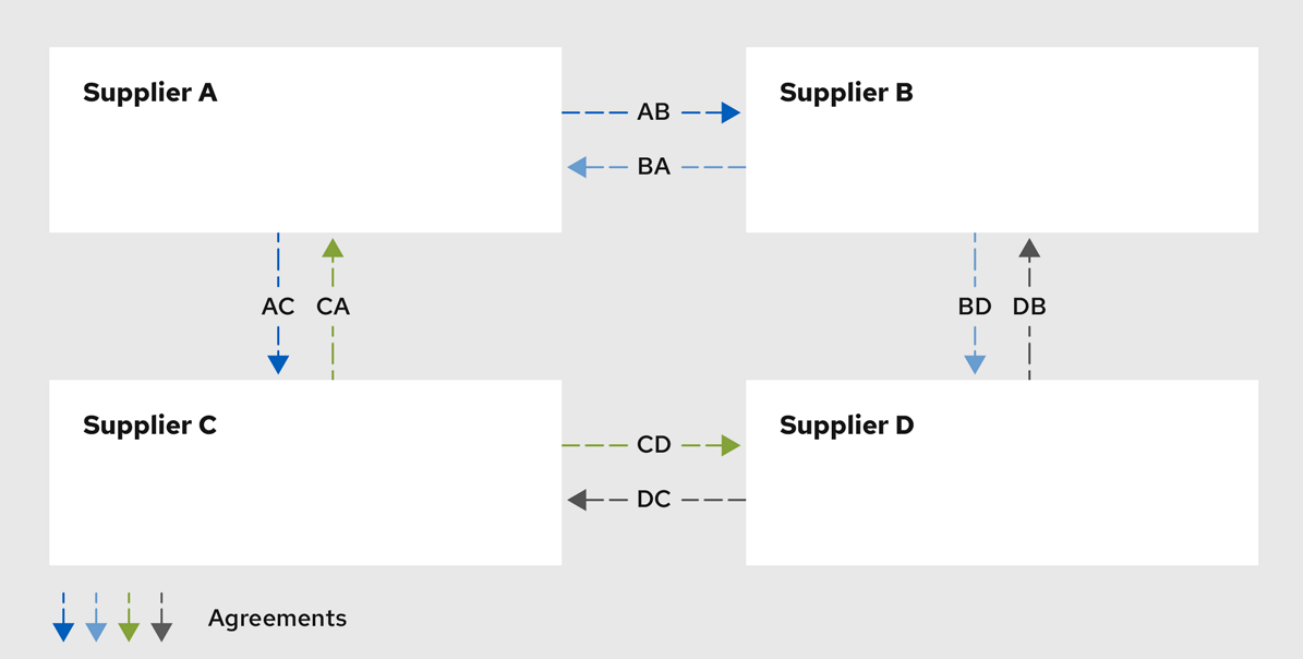


20개의 공급업체가 있는 경우 총 20개의 공급업체(각각 19개의 공급업체)에 380개의 복제 계약을 생성해야 합니다.

두 개 이상의 서버가 동시에 실패할 가능성이 적거나 특정 공급 업체 간의 연결이 작으면 부분적으로 연결된 토폴로지를 사용하는 것이 좋습니다.

부분적으로 연결된 토폴로지

다음 다이어그램은 각 공급자 서버가 두 공급자 서버에 데이터를 복제하는 토폴로지를 보여줍니다. 이전 예제 토폴로지에 비해 4개의 공급업체 서버 간에는 8개의 복제 계약만 존재합니다.



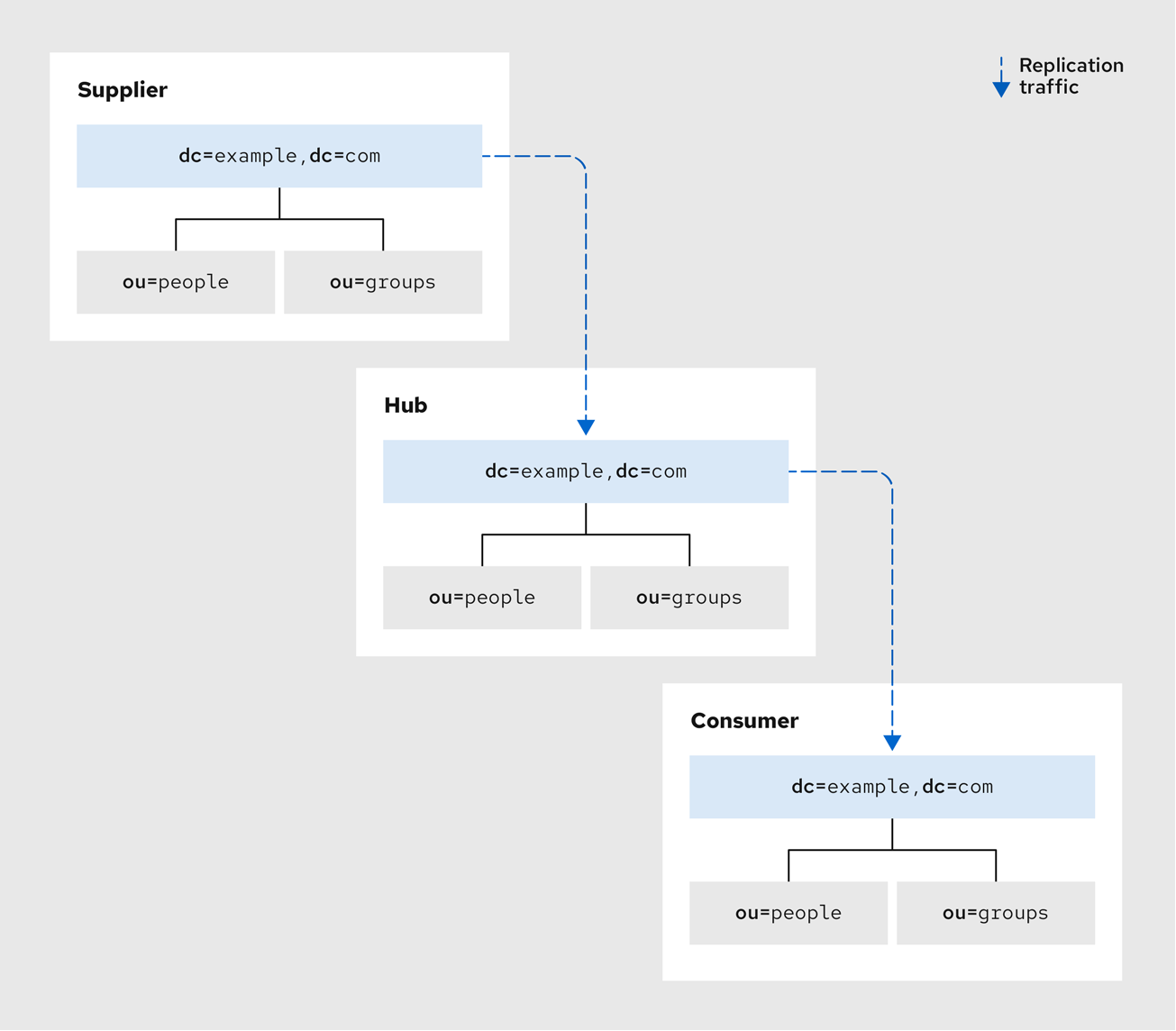
6.2.3. 계단식 복제

계단식 복제 시나리오에서는 허브 서버가 공급자 서버에서 업데이트를 수신하고 이러한 업데이트를 소비자 서버로 보냅니다. 허브 서버는 일반적인 소비자 서버와 같은 읽기 전용 복제본을 보유하고 있으며 일반적인 공급자 서버와 같은 변경 로그를 유지 관리하기 때문에 하이브리드입니다.

Hub 서버는 공급자 데이터를 소비자에게 전달하고 디렉터리 클라이언트의 업데이트 요청을 공급자에게 참조합니다.

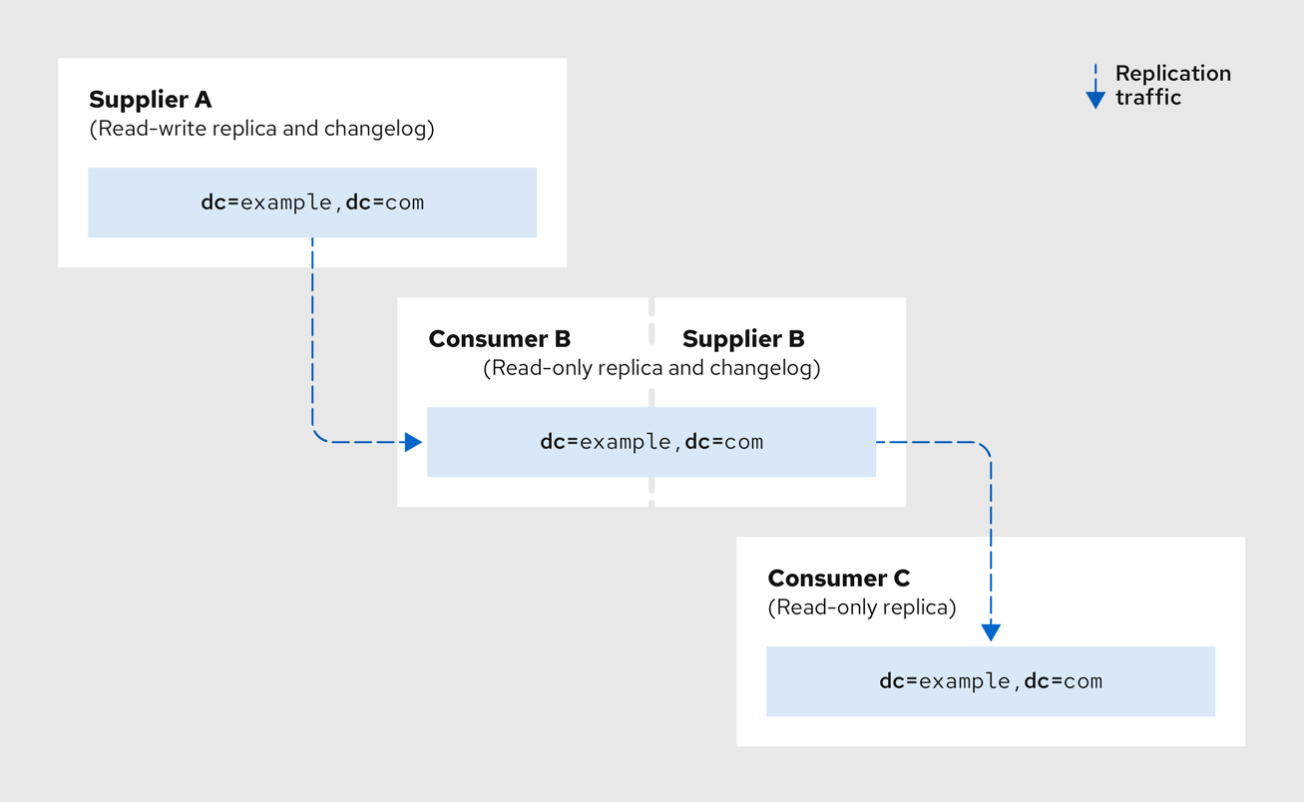
다음 다이어그램에서는 계단식 복제 시나리오를 보여줍니다.

그림 6.3. 계단식 복제 시나리오



다음 다이어그램은 각 서버에 복제본을 구성하는 방법과 변경 로그를 유지 관리하는 서버(읽기-쓰기 또는 읽기 전용)를 보여줍니다.

그림 6.4. cascading 복제의 복제 트래픽 및 변경 로그



계단식 복제는 다음과 같은 경우에 유용합니다.

- 많은 트래픽 부하를 분산합니다. 복제 토폴로지의 공급자는 모든 업데이트 트래픽을 관리하므로 소비자에게 복제 트래픽을 지원하기 위해 많은 부하가 있을 수 있습니다. 많은 수의 소비자에 대한 복제 업데이트를 서비스할 수 있는 허브로 복제 트래픽을 리디렉션할 수 있습니다.

- 지리적으로 분산된 환경에서 로컬 허브 공급업체를 사용하여 연결 비용을 절감하려면 다음을 수행하십시오.

- 디렉터리 서비스의 성능을 높이기 위해 다음을 수행합니다. 모든 읽기 작업을 소비자에게 전달하고 모든 업데이트 작업을 공급자에게 보내는 경우 허브 서버에서 모든 인덱스(시스템 인덱스 제외)를 제거할 수 있습니다. 이렇게 하면 공급자와 허브 서버 간 복제 속도가 크게 증가합니다.

추가 리소스

- 로드 밸런싱에 복제 사용

- 고가용성에 복제 사용

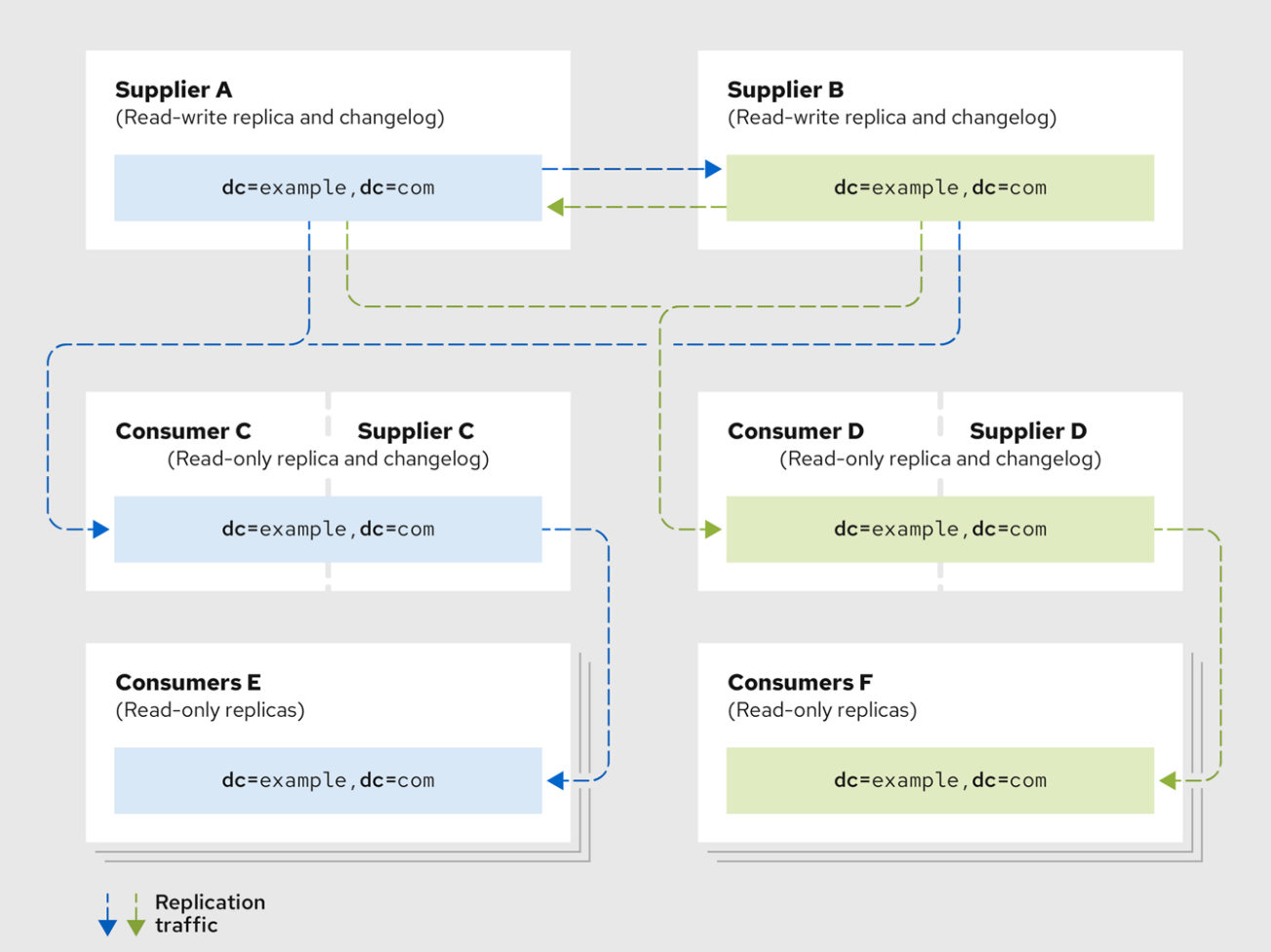
- 로컬 가용성에 복제 사용

6.2.4. 혼합 시나리오

복제 시나리오를 네트워크 및 디렉터리 환경의 요구 사항에 맞게 결합할 수 있습니다. 일반적인 조합 중 하나는 계단식 구성과 함께 다중 제공 구성을 사용하는 것입니다.

다음 다이어그램은 혼합 시나리오에 대한 예제 토폴로지를 보여줍니다.

그림 6.5. Multi-supplier 및 cascading 복제



6.3. 복제 전략 정의

제공하려는 서비스에 따라 복제 전략을 결정할 수 있습니다. 다음은 구현할 수 있는 일반적인 복제 전략입니다.

- 고가용성이 주요 문제인 경우 단일 사이트에 여러 디렉터리 서버가 있는 데이터 센터를 생성합니다. 단일 공급 업체 복제는 읽기-failover를 제공하는 반면 다중 제공 복제는 쓰기 장애 조치

(write-failover)를 제공합니다.

자세한 내용은 고가용성에 대한 복제 사용을 참조하십시오.

- 로컬 가용성이 주요 문제인 경우 복제를 사용하여 전 세계 로컬 사무실의 디렉터리 서버에 데이터를 지리적으로 배포합니다. 회사 본부와 같은 단일 위치에 모든 정보의 주요 사본을 유지하거나 각 로컬 사이트에서 관련 디렉토리 부분을 관리할 수 있습니다.

  자세한 내용은 로컬 가용성에 복제 사용을 참조하십시오.

- Directory Server에서 네트워크 혼잡을 관리하고 방지하는 요청 부하를 조정하려면 로드 밸런싱에 복제 구성을 사용합니다.

  자세한 내용은 로드 밸런싱에 복제 사용을 참조하십시오.

- 다른 위치 또는 회사의 섹션에 여러 소비자를 사용하거나 일부 서버가 안전하지 않은 경우 소수 복제를 사용하여 중요한 정보를 손상시키지 않고 데이터 무결성을 유지 관리하기 위해 중요한 정보를 제외하거나 변경되지 않습니다.

  자세한 내용은 Fractional replication 을 참조하십시오.

- 네트워크가 여러 사이트에서 여러 Directory 서버를 포함하는 광범위한 지역에 걸쳐 확장되고 다중 공급 업체 복제에 의해 연결된 로컬 데이터 공급 업체의 경우 광범위한 영역 네트워크에 복제 구성을 사용합니다.

  자세한 내용은 광범위한 네트워크에서 복제를 참조하십시오.

복제 전략을 확인하려면 네트워크, 사용자, 애플리케이션 및 디렉터리 서비스를 사용하는 방법에 대한 설문 조사를 수행하여 시작합니다.

6.3.1. 복제 설문 조사 수행

복제 전략을 정의하는 데 도움이 되는 네트워크 품질 및 사용법에 대한 정보를 수집합니다.

- 서로 다른 빌딩 또는 원격 사이트를 연결하는 LAN 및 WAN의 품질과 사용 가능한 대역폭의 양입니다.

- 사용자의 물리적 위치, 각 사이트에 있는 사용자 수 및 디렉터리 서비스 사용 방법을 이해하기 위한 사용 패턴입니다.

  인적 자원 데이터베이스 또는 금융 정보를 관리하는 사이트는 일반적으로 전화 서적 용도로만 디렉토리를 사용하는 엔지니어링 직원이 포함된 사이트보다 디렉토리에 더 많은 부하를 생성합니다.

- 디렉터리에 액세스하는 애플리케이션 수 및 읽기,검색 및 비교 작업의 상대 백분율로 작업을 작성합니다.

  메시징 서버에서 디렉터리를 사용하는 경우 처리하는 각 이메일 메시지에 대해 수행하는 작업 수를 확인합니다. 디렉터리를 사용하는 기타 제품은 일반적으로 인증 애플리케이션 또는 메타 디렉터리 애플리케이션과 같은 제품입니다. 각 애플리케이션에 대해 디렉터리에서 수행되는 작업의 유형 및 빈도를 결정합니다.

- 디렉터리에 저장된 항목의 수 및 크기입니다.

6.3.2. 복제 리소스 요구 사항

복제에는 리소스가 필요합니다. 복제 전략을 정의할 때 다음 리소스 요구 사항을 고려하십시오.

디스크 사용량

공급자 서버에서 **Directory Server**는 각 업데이트 작업 후에 변경 로그를 작성합니다. 따라서 많은 업데이트 작업을 수신하는 공급업체 서버는 디스크 사용량이 높습니다.

서버 스레드

각 복제 계약에는 전용 스레드가 생성되고 **CPU** 로드는 복제 처리량에 따라 달라집니다.

파일 설명자

서버는 변경 로그에 하나의 파일 설명자를 사용하고 복제 계약에 대해 하나의 파일 설명자를 사용합니다.

6.3.3. 다중 제공 복제에 필요한 디스크 공간 관리

다중 제공 토폴로지에서 공급업체는 디렉터리 편집의 변경 로그, 업데이트된 항목에 대한 상태 정보, 삭제된 항목에 대한 **tombstone** 항목을 포함하여 복제에 필요한 추가 로그를 유지 관리합니다. 이러한 로

그 파일은 매우 커질 수 있으므로 디스크 공간을 불필요하게 사용하지 않도록 이러한 파일을 정기적으로 정리해야 합니다.

각 서버에서 다음 특성을 사용하여 복제 환경에서 복제 로그 유지 관리를 구성할 수 있습니다.

- nsslapd-changelogmaxage 속성은 변경 로그의 최대 항목 수를 설정합니다. 항목이 최대 사용 기간 값보다 오래된 경우 디렉터리 서버는 해당 항목을 삭제합니다. 최대 항목 수를 설정하면 변경 로그가 무기한 증가되지 않습니다.

- nsslapd-changelogmaxentries 속성은 변경 로그에 포함될 수 있는 최대 항목 수를 설정합니다. **nsslapd-changelogmaxentries** 값은 전체 디렉터리 정보 세트를 포함할 수 있을 만큼 충분히 커야 합니다. 그렇지 않으면 다중 제공 복제가 문제가 있을 수 있습니다.

- nsDS5ReplicaPurgeDelay 는 최대 **tombstone (deleted)** 항목 및 변경 로그의 상태 정보를 설정합니다. **tombstone** 또는 **state information** 항목이 해당 기간보다 오래된 경우 **Directory Server**는 해당 항목을 삭제합니다. **nsDS5ReplicaPurgeDelay** 값은 **tombstone** 및 **state** 정보 항목에만 적용되지만 **nsslapd-changelogmaxage** 는 디렉터리 수정을 포함하여 **changelog**의 모든 항목에 적용됩니다.

- nsDS5ReplicaTombstonePurgeInterval 속성은 서버가 제거 작업을 실행하는 빈도를 설정하여 변경 로그에서 **tombstone** 및 **state** 항목을 정리합니다. 최대 사용 기간이 가장 긴 복제 업데이트 일정보다 길어야 합니다. 그렇지 않으면 복제본을 업데이트할 때 다중 제공 복제에 문제가 있을 수 있습니다.

6.3.4. 고가용성에 복제 사용

단일 서버가 실패할 때 디렉터리를 사용할 수 없는 경우 복제를 사용하여 디렉터리를 사용할 수 없습니다. 최소한 로컬 디렉터리 트리를 하나 이상의 백업 서버에 복제합니다. 내결함성을 위해 복제하는 빈도는 요구 사항에 따라 다릅니다. 그러나 이 결정은 디렉터리에서 사용하는 하드웨어 및 네트워크의 품질에 따라 결정됩니다. 신뢰할 수 없는 하드웨어에는 더 많은 백업 서버가 필요합니다.

중요

복제 및 백업의 용도가 다르기 때문에 복제를 일반 데이터 백업 정책 대신 사용하지 마십시오. 디렉터리 데이터를 백업하는 방법에 대한 자세한 내용은 Red Hat Directory Server 백업 및 복원을 참조하십시오.

다음 전략을 선택하여 디렉터리를 사용할 수 없도록 할 수 있습니다.

- 모든 디렉터리 클라이언트에 대해 쓰기 장애 조치를 보장하려면 다중 제공 복제를 사용합니다.

- read-failover를 보장하려면 단일 공급 업체 복제 를 사용하십시오.

LDAP 클라이언트 애플리케이션은 일반적으로 하나의 LDAP 서버만 검색하도록 구성됩니다. 다른 DNS 호스트 이름에 있는 LDAP 서버를 통해 회전할 사용자 지정 클라이언트 애플리케이션이 없는 경우 Directory Server의 단일 DNS 호스트 이름을 확인하도록 LDAP 클라이언트 애플리케이션만 구성할 수 있습니다. 따라서 DNS 라운드 로빈 또는 네트워크 정렬을 사용하여 백업 디렉터리 서버에 대한 장애 조치를 제공해야 할 수 있습니다.

6.3.5. 로컬 가용성에 복제 사용

네트워크의 품질 및 데이터가 미션 크리티컬한지 여부에 따라 로컬 가용성에 복제를 사용해야 할 수도 있습니다.

다음과 같은 이유로 로컬 가용성에 복제를 사용하십시오.

- 데이터의 로컬 주요 사본이 필요합니다.

  대규모 다국적 기업은 특정 국가의 직원에게만 관심있는 디렉토리 정보를 유지해야 할 수도 있습니다. 또한 데이터의 로컬 주요 사본을 보유하는 것은 부서 또는 조직 수준에서 데이터를 제어하도록 지시하는 모든 기업에 중요합니다.

- 신뢰할 수 없거나 간헐적으로 사용 가능한 네트워크 연결이 있습니다.

  국제 네트워크에는 간헐적인 네트워크 연결을 유발하는 신뢰할 수 없는 WAN이 있습니다.

- Directory Server 성능에 영향을 미치는 주기적이고 매우 많은 네트워크 로드가 있습니다.

  노후된 네트워크를 사용하는 기업은 정상적인 업무 시간 동안 네트워크 로드가 많이 발생할 수 있습니다.

- 공급업체의 네트워크 로드 및 워크로드를 줄이고자 합니다.

네트워크가 안정적이고 사용 가능한 경우에도 네트워크 비용을 줄일 수 있습니다.

6.3.6. 로드 밸런싱에 복제 사용

디렉터리 데이터를 복제하는 주요 이유 중 하나는 네트워크의 워크로드 균형을 유지하고 디렉터리 성능을 개선하는 것입니다.

디렉터리 항목은 일반적으로 **1KB** 크기이므로 모든 디렉터리 검색은 네트워크 로드에 약 **1KB**를 추가합니다. 디렉터리 사용자가 하루에 **10**개의 디렉토리 검색을 수행하는 경우 모든 디렉터리에 대한 증가된 네트워크 로드는 매일 약 **10KB**입니다. 느리거나 많이 로드되거나 신뢰할 수 없는 **WAN**이 있는 경우 디렉터리 트리를 로컬 서버로 복제해야 할 수 있습니다.

그러나 로컬에서 사용 가능한 데이터가 복제로 인한 증가된 네트워크 로드 비용의 가치가 있는지 확인합니다. 전체 디렉터리 트리를 원격 사이트에 복제하는 경우 사용자의 검색으로 인한 트래픽과 비교하여 네트워크에 더 큰 로드를 추가할 수 있습니다. 특히 디렉터리가 자주 변경되지만 원격 사이트에서 하루에 몇 개의 디렉토리 검색을 수행하는 사용자는 몇 명뿐입니다.

다음 표에서는 디렉터리 복제의 부하 영향을 **1** 만 개의 항목으로, 여기서 **100,000** 개의 항목이 매일 변경되고, 매일 **10**개의 검색을 수행하는 **100**명의 직원 중 작은 원격 사이트를 갖는 부하 영향을 비교하고, 매일 **10**개의 검색을 수행하는 **100**명의 직원 중 부하 영향을 비교합니다.

표 6.1. 네트워크에서 복제 및 원격 검색의 영향

로드 유형	액세스/일	평균 항목 크기	load
복제	100,000	1KB	100MB/day
원격 검색	1,000	1KB	1MB/일

네트워크를 과부하하지 않고 로컬 사이트에 데이터를 사용할 수 있도록 하는 것 사이의 중단은 예약된 복제를 사용하는 것입니다. 데이터 일관성 및 복제 스케줄에 대한 자세한 내용은 데이터 일관성을 참조하십시오.

추가 리소스

- 네트워크 로드 밸런싱의 예
- 성능 향상을 위한 로드 밸런싱 예

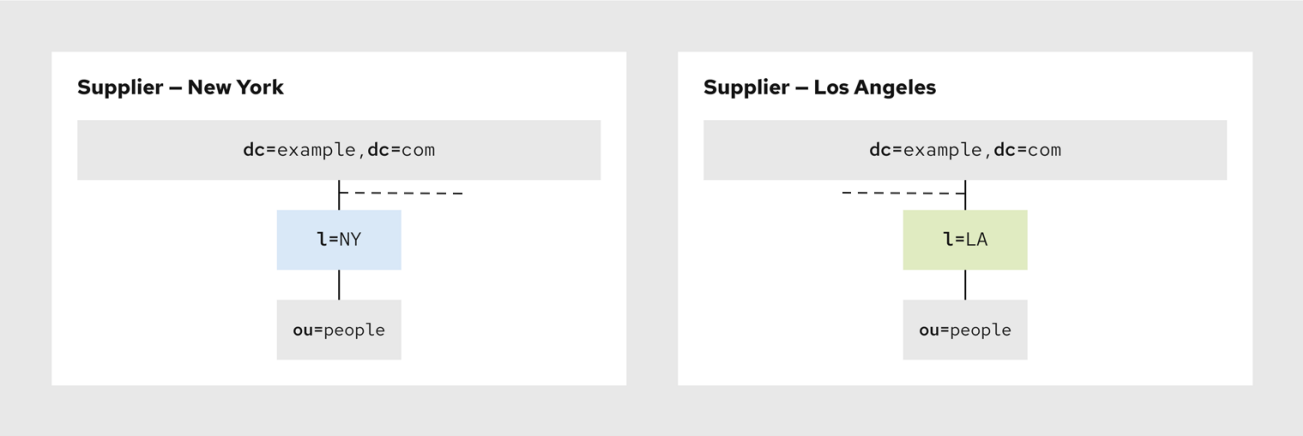
- 소규모 사이트의 복제 전략 예
- 대규모 사이트에 대한 복제 전략 예

6.3.6.1. 네트워크 로드 밸런싱의 예

이 예에서는 뉴질랜드 (NY) 및 로스 앤젤레스 (LA)에 사무실이 있고 각 사무실이 별도의 하위 트리를 관리하는 기업에 대해 설명합니다.

다음 다이어그램에서는 기업이 하위 트리를 관리하는 방법을 보여줍니다.

그림 6.6. enterprise Cryostat 및 LA 하위 트리



647_RHDS_0624

각 사무실에는 고속 네트워크가 포함되어 있지만 두 도시 간의 연결은 신뢰할 수 없습니다. 네트워크 로드의 균형을 조정하려면 다음 전략을 사용합니다.

- 각 사무실에서 로컬 관리 데이터의 공급자 서버로 전환하려면 하나의 서버를 선택합니다.

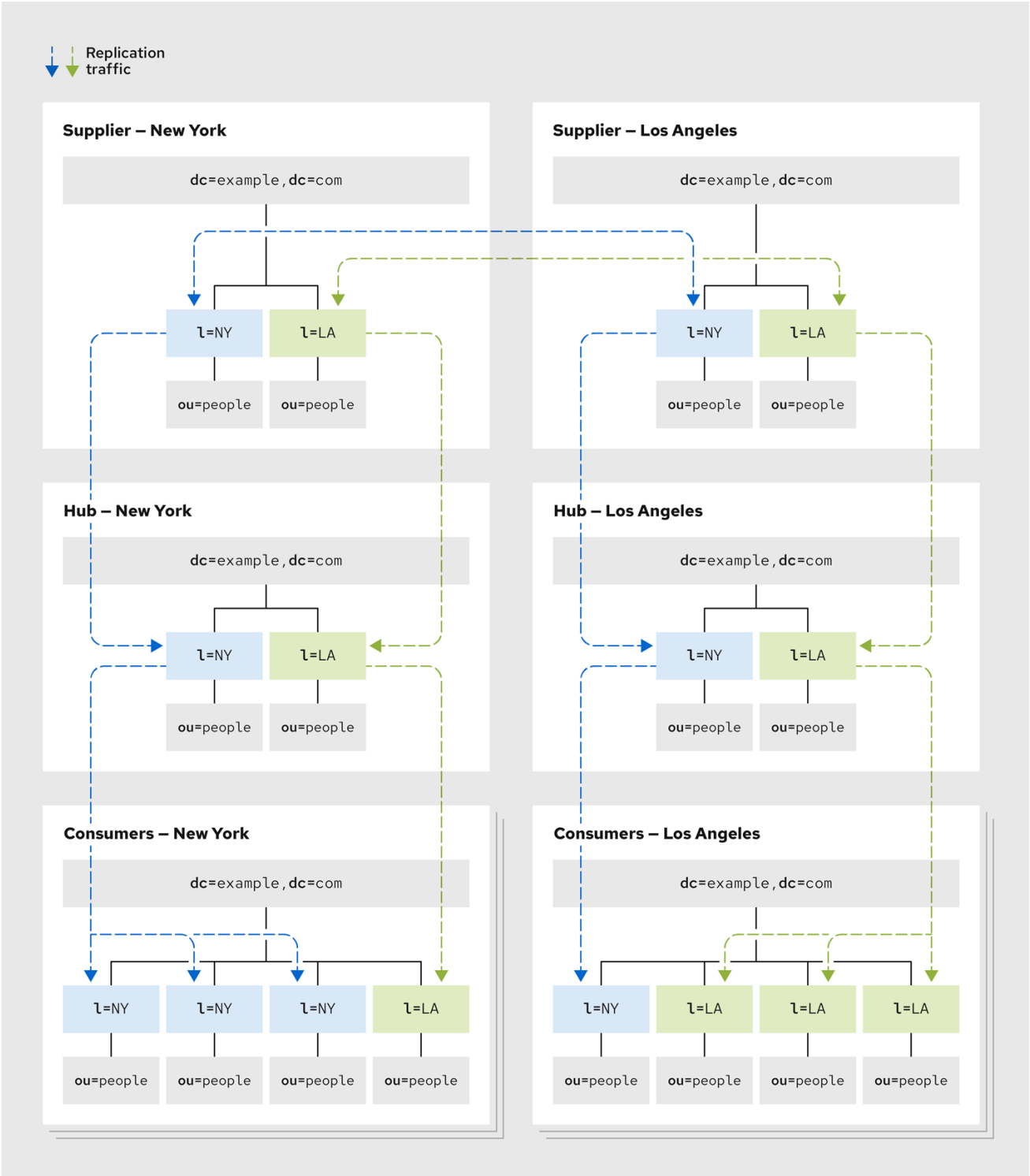
  해당 공급 업체에서 원격 사무실의 해당 공급자 서버로 로컬 관리 데이터를 복제합니다. 각 위치에 데이터의 주요 사본이 있는 경우 사용자는 신뢰할 수 없는 연결을 통해 업데이트 및 검색 작업을 수행하지 않습니다. 결과적으로 성능이 최적화됩니다.

- 디렉터리 데이터의 가용성을 보장하기 위해 각 공급자 서버(원격 사무실에서 제공된 데이터 포함)의 디렉터리 트리를 하나 이상의 로컬 디렉터리 서버로 복제합니다.

- 추가 로드 밸런싱을 제공하기 위해 로컬 데이터를 검색하는 데 전용 소비자 수가 증가하여 각 위치에서 계단식 복제를 구성합니다.

Cryostat 사무실은 **LA** 특정 검색보다 더 많은 특정 검색을 생성합니다. 이 예제에서는 **3** 명의 데이터 소비자와 **1 LA** 소비자가있는 사무실을 보여줍니다. **LA** 사무실에는 **3**개의 **LA** 데이터 소비자와 **1**개의 **data consumer**가 있습니다.

그림 6.7. 엔터프라이즈 로드 밸런싱 예



추가 리소스

- cascading-replication
- 다중 제공 복제

6.3.6.2. 성능 향상을 위한 로드 밸런싱 예

이 예제에서는 다음과 같은 특성을 가진 엔터프라이즈를 설명합니다.

- 디렉터리에는 1,000,000명의 사용자를 지원하는 Cryostat,000개의 항목이 포함되어 있습니다.
- 각 사용자는 하루에 10개의 디렉토리 검색을 수행합니다.
- 메시징 서버는 하루에 25,000,000개의 메일 메시지를 처리하고 모든 메일 메시지에 대해 5개의 디렉터리 검색을 수행합니다.
- 사용자는 4개의 시간대에 분산됩니다.

이는 총 135,000,000개의 디렉토리 검색과 동일합니다.

1,000,000명의 사용자 x 10 검색 = 10,000,000 사용자 검색

25,000,000개의 메일 5 검색 = 하루 125,000,000개의 메일 검색

매일 10,000,000 + 125,000,000 = 135,000,000개의 모든 검색

8시간 동안의 업무 시간 및 사용자가 4개의 시간대에 걸쳐 퍼지면서 4개의 시간대에 걸쳐 최대 사용량은 12시간으로 연장됩니다. 따라서 디렉터리 서버는 12시간 동안 135,000,000개의 디렉터리 검색을 지원해야 합니다. 이는 초당 3,125개의 검색에 해당합니다(135,000,000 / (60*60*12)).

Directory Server를 실행하는 하드웨어가 초당 500 읽기를 지원하는 경우 이 부하를 지원하기 위해 최소 6개 이상의 Directory 서버를 사용해야 합니다. 수백만 개의 디렉터리 사용자를 보유한 엔터프라이즈

의 경우 로컬 가용성에 사용할 **Directory Server**를 더 추가합니다.

이러한 시나리오에서는 다음 복제 전략을 사용할 수 있습니다.

- 모든 쓰기 트래픽을 처리할 수 있도록 다중 제공 구성에 두 개의 **Directory Server**를 배치합니다.

  이 구성은 모든 디렉터리 데이터에 대해 단일 제어 지점을 원하는 것으로 가정합니다.

- 공급자 서버를 사용하여 하나 이상의 허브에 복제합니다.

  소비자의 읽기,검색 및 비교하여 공급업체가 쓰기 요청만 처리할 수 있도록 합니다. 허브에 대한 자세한 내용은 Cascading-replication 을 참조하십시오.

- 허브를 사용하여 엔터프라이즈 전체의 로컬 사이트에 복제할 수 있습니다.

  로컬 사이트를 복제하면 서버와 네트워크의 부하를 분산하고 디렉터리 데이터의 고가용성을 보장할 수 있습니다.

- 각 사이트에서 읽기 작업의 경우 최소한 한 번 이상 복제하여 고가용성을 보장합니다.

  DNS 정렬을 사용하여 로컬 사용자가 항상 디렉터리 검색에 사용할 수 있는 로컬 디렉터리 서버를 찾도록 합니다.

6.3.6.3. 소규모 사이트의 복제 전략 예

예제 **enterprise**에는 다음과 같은 특성이 있습니다.

- 전체 기업은 하나의 건물에 포함되어 있습니다.
- 이 건물은 매우 빠른 (초당 **100Mb**) 및 간단한 네트워크입니다.
- 네트워크는 매우 안정적이 고 서버 하드웨어 및 **OS** 플랫폼이 신뢰할 수 있습니다.

- 단일 서버는 부하를 쉽게 처리할 수 있습니다.

이러한 조건을 사용하면 유지 관리 또는 하드웨어 업그레이드를 위해 기본 서버를 종료할 때 가용성을 보장하기 위해 한 번 이상 복제해야 합니다. 또한 **Directory Server** 중 하나를 사용할 수 없게 되는 경우 **LDAP** 연결 성능을 개선하기 위해 **DNS** 라운드 로빈을 설정합니다.

6.3.6.4. 대규모 사이트에 대한 복제 전략 예

소규모 사이트에 대한 **Example** 복제 전략 의 예는 다음과 같은 특징이 있습니다.

- 이 회사는 두 개의 별도의 빌딩, **Building A** 및 **Building B**에 포함되어 있습니다.
- 건물 간의 연결은 정상 영업 시간 동안 느리고 매우 바쁘다.
- 각 빌딩에는 매우 빠른 (초당 **100Mb**) 및 간단한 네트워크(**Lightly used network**)가 있습니다.
- 각 빌드 내의 네트워크는 매우 안정적이며 서버 하드웨어 및 **OS** 플랫폼은 신뢰할 수 있습니다.
- 단일 서버는 하나의 빌드 내에서 부하를 쉽게 처리할 수 있습니다.

이러한 조건을 사용하면 복제 전략에 다음 단계가 포함됩니다.

- 두 건물 중 하나에서 단일 서버를 선택하여 디렉터리 데이터의 주요 복사본을 포함합니다.

  디렉터리 데이터의 주요 복사본을 담당하는 가장 많은 수의 사용자가 포함된 빌딩에 서버를 배치합니다(예: 빌드 **A**).

- 디렉터리 데이터의 고가용성을 위해 빌드 **A** 내에서 한 번 이상 복제하십시오.

  다중 제공 복제 구성을 사용하여 쓰기 장애 조치(**failover**)를 확인합니다.

- 두 번째 **Building B**에 두 개의 복제본을 생성합니다.

- 공급자와 소비자 서버 간에 일관성이 거의 필요하지 않은 경우 사용량이 부족한 시간 동안만 복제를 예약하십시오.

6.3.7. 부분 복제

부분 복제를 사용하면 **Directory Server**가 공급 업체에서 소비자 또는 다른 공급자로 복제하지 않는 속성 집합을 선택할 수 있습니다. 따라서 데이터베이스에 포함된 모든 정보를 복제하지 않고 데이터베이스를 복제할 수 있습니다.

부분 복제는 복제 계약에 따라 활성화 및 구성됩니다. **Directory Server**는 특성을 모든 항목에 동일하게 제외합니다. 제외된 속성은 항상 소비자에 대한 값이 없습니다. 따라서 소비자 서버에 대해 검색을 수행하는 클라이언트는 검색 필터가 이러한 특성을 명시적으로 지정하는 경우에도 제외된 속성을 볼 수 없습니다.

다음과 같은 경우 소수 복제를 사용합니다.

- 소비자 서버는 느린 네트워크를 사용하여 연결됩니다. **jpeg Cryostat**와 같이 거의 변경되지 않은 속성 또는 더 큰 속성을 제외하면 네트워크 트래픽이 줄어듭니다.

- 소비자 서버는 공용 인터넷과 같은 신뢰할 수 없는 네트워크에 배치됩니다. 전화 번호와 같은 민감한 속성을 제외하면 서버 액세스 제어 조치가 손상되거나 공격자가 시스템이 손상되는 경우에도 중요한 속성에 액세스할 수 없는 추가 수준의 보호를 제공합니다.

6.3.8. 광역 네트워크에서 복제

광역 네트워크(**wide area networks**)는 일반적으로 더 높은 대기 시간, 대역폭 지연 제품, 로컬 영역 네트워크보다 빠른 속도를 갖습니다. **Directory Server**는 공급 업체와 소비자가 광역 네트워크를 사용하여 연결할 때 효율적인 복제를 지원합니다.

이전에는 공급자가 하나의 업데이트 작업만 보낸 다음 소비자의 응답을 기다리기 때문에 **Directory Server**에서 사용하는 복제 프로토콜이 대기 시간에 민감했습니다. 이로 인해 대기 시간이 길어진 처리량이 줄어들었습니다.

현재 공급자는 응답을 기다리지 않고 많은 업데이트 및 항목을 소비자에게 전송하고 복제 처리량은 로컬 영역 네트워크의 처리량과 유사합니다.

WAN을 사용할 때 다음과 같은 성능 및 보안 문제를 고려하십시오.

- TLS(Transport Layer Security) 프로토콜을 사용하여 인터넷과 같은 공용 네트워크에서 수행되는 복제를 보호합니다.
- 네트워크에 대해 T1 또는 더 빠른 인터넷 연결을 사용합니다.
- WAN을 통해 복제에 대한 계약을 생성할 때 서버 간의 지속적인 동기화를 방지합니다. 복제 트래픽은 대역폭의 많은 부분을 소비하고 전체 네트워크 및 인터넷 연결을 느리게 할 수 있습니다.

추가 리소스

- 다중 제공 복제 환경에서 대기 시간 개선

6.4. 다른 DIRECTORY SERVER 기능과 함께 복제 사용

복제 전략을 더 잘 설계하려면 복제와 다른 Directory Server 기능 간의 상호 작용에 대해 알아보십시오.

6.4.1. 복제 및 액세스 제어

디렉터리는ACI(Access Control instructions)를 항목의 속성으로 저장하고 디렉터리 서버는 다른 디렉터리 콘텐츠와 함께 이러한 ACI를 복제합니다. 예를 들어 특정 호스트의 디렉터리에 대한 액세스를 제한하려면 ACI의 호스트별 설정만 사용합니다. 그렇지 않으면 ACI가 다른 서버에 복제되면 Directory Server가 ACI를 로컬로 평가하므로 모든 서버에서 디렉터리에 대한 액세스가 거부됩니다.

디렉터리에 대한 액세스 제어 설계에 대한 자세한 내용은 액세스 제어 설계를 참조하십시오.

6.4.2. 복제 및 디렉터리 서버 플러그인

복제는 Directory Server에서 제공되는 대부분의 플러그인에서 작동합니다. 그러나 다음 플러그인에는 다중 제공 환경에서 제한 사항 및 예외가 있습니다.

- 특성 고유성 플러그인

  특성 고유성 플러그인은 로컬 서버의 항목에만 추가된 속성 값의 고유성을 검증합니다. 예를 들어, 회사에서는 사용자 항목에 대해 **mail** 속성이 고유해야 합니다. 서로 다른 두 명의 사용자가 동시에 두 개의 다른 공급자 서버의 **mail** 속성에 대해 동일한 값으로 추가되면 이름이 충돌하지 않고 복제 충돌이 발생하지 않기 때문에 **Directory Server**는 이러한 사용자를 디렉터리에 추가합니다. 특성 고유성 플러그인은 복제된 변경 사항을 확인하지 않으며 결과적으로 **mail** 특성 값은 디렉터리에서 고유하지 않습니다.

- 참조 무결성 플러그인

  참조 무결성은 **multi-supplier** 세트의 하나의 공급자에서만 활성화된 경우 다중 공급 업체 복제에서 작동합니다. 이렇게 하면 참조 무결성 업데이트가 공급자 서버 중 하나에서만 발생하고 다른 서버로 전파됩니다.

- 자동 멤버십 및 **MemberOf** 플러그인

  이 두 플러그인이 복제 환경에서 제대로 작동하려면 각 서버에서 로컬로 업데이트를 수행하도록 플러그인을 구성합니다.

참고

기본적으로 플러그인은 비활성화되어 있으며 수동으로 활성화해야 합니다.

추가 리소스

- 서버 플러그인 기능 참조

6.4.3. 복제 및 데이터베이스 링크

체인을 사용하여 디렉터리에 항목을 배포할 때 데이터베이스 링크가 포함된 서버는 실제 데이터가 포함된 원격 서버를 나타냅니다. 이 환경에서는 데이터베이스 링크를 복제할 수 없습니다. 그러나 원격 서버의 실제 데이터가 포함된 데이터베이스를 복제할 수 있습니다.

6.4.4. 스키마 복제

복제된 환경에서는 복제에 참여하는 모든 서버에서 스키마를 일관되게 유지해야 합니다. 스키마 일관성을 보장하기 위해 단일 공급자 서버에서만 스키마를 수정합니다.

서버 간 복제를 구성한 경우 기본적으로 스키마 복제가 수행됩니다.

표준 스키마

Directory Server는 표준 스키마 복제에 다음 시나리오를 사용합니다.

1. 소비자 서버로 데이터를 푸시하기 전에 공급자 서버는 해당 스키마 버전이 소비자 서버에 보관된 스키마 버전과 동일한지 확인합니다.

2. 공급자와 소비자의 스키마 항목이 모두 동일하면 복제 작업이 진행됩니다.

3. 공급자 스키마 버전이 소비자 스키마 버전보다 최신이면 공급자 서버는 데이터 복제를 진행하기 전에 해당 스키마를 소비자에게 복제합니다.

4. 공급자 스키마 버전이 소비자 스키마 버전보다 오래된 경우 복제가 실패하거나 소비자의 스키마가 새 데이터를 지원할 수 없기 때문에 복제 중에 오류가 반환될 수 있습니다. 따라서 소비자 서버의 스키마를 업데이트하지 마십시오. 복제 토폴로지의 공급자 서버에서만 스키마를 유지해야 합니다.

Directory Server는 **dsconf** 명령, 웹 콘솔, **LDAP** 수정 작업 또는 **99user.ldif** 파일을 사용하여 수행한 스키마에 대한 변경 사항을 복제합니다.

두 공급업체 서버에서 스키마를 변경하면 소비자는 두 공급업체의 데이터를 수신하며 각각 다른 스키마를 사용합니다. 소비자는 최신 스키마 버전이 있는 공급업체의 수정을 적용합니다. 이러한 상황에서 소비자의 스키마는 항상 공급자 중 하나와 다릅니다. 이를 방지하려면 항상 하나의 공급 업체에서만 스키마 변경을 수행해야 합니다.

스키마를 복제하기 위해 특수 복제 계약을 만들 필요가 없습니다. 그러나 동일한 디렉터리 서버는 공급자 및 소비자 복제본을 보유할 수 있습니다. 따라서 항상 스키마의 공급자 역할을 하는 서버를 확인한 다음 스키마 정보에 대한 소비자 역할을 하는 복제 환경과 복제 환경의 다른 모든 서버 간에 복제 계약을 설정합니다.

표준 스키마 파일에 대한 자세한 내용은 표준 스키마 를 참조하십시오.

사용자 정의 스키마

표준 **99user.ldif** 파일을 사용자 지정 스키마로 사용하는 경우 **Directory Server**는 모든 사용자에게 사용자 지정 스키마만 복제합니다. 웹 콘솔 또는 **dsconf** 명령을 통해 변경한 경우에도 **Directory Server**는 다른 사용자 지정 스키마 파일 또는 이러한 파일에 대한 변경 사항을 복제하지 않습니다.

다른 사용자 지정 파일을 사용하는 경우 공급업체를 변경한 후 이러한 파일을 토폴로지의 모든 서버에 수동으로 복사해야 합니다.

추가 리소스

- 스키마 사용자 정의
- **Directory Server**가 복제 환경의 스키마 업데이트를 관리하는 방법.

7장. 보안 디렉터리 설계

design-rhds

Red Hat Directory Server가 데이터를 보호하는 방법은 이전의 모든 설계 영역에 영향을 미칩니다. 모든 보안 설계는 디렉터리의 데이터를 보호하고 사용자와 애플리케이션의 보안 및 개인 정보 보호 요구 사항을 충족해야 합니다.

보안 요구 사항 및 이러한 요구 사항을 충족하기 위해 디렉토리를 설계하는 방법을 알아보십시오.

7.1. 보안 위협 정보

이 디렉터리는 잠재적인 보안 위협의 위험이 있을 수 있습니다. 가장 일반적인 위협을 이해하면 전체 보안 설계를 설명하는 데 도움이 됩니다. 디렉터리 보안에 대한 위협은 다음 세 가지 범주로 분류됩니다.

- 무단 액세스
- 무단 변조
- 서비스 거부

7.1.1. 무단 액세스

무단 액세스로부터 디렉토리를 보호하는 것은 쉽지 않은 것처럼 보일 수 있지만 보안 솔루션을 구현하는 것이 처음 나타나는 것보다 더 복잡할 수 있습니다. 디렉터리 정보 전달 경로에는 권한이 없는 클라이언트가 데이터에 액세스할 수 있는 여러 잠재적인 액세스 지점이 있습니다.

다음 시나리오에서는 권한이 없는 클라이언트가 디렉터리 데이터에 액세스하는 방법에 대한 몇 가지 예를 설명합니다.

- 권한이 없는 클라이언트는 다른 클라이언트 자격 증명을 사용하여 데이터에 액세스할 수 있습니다. 특히 디렉터리가 보호되지 않은 암호를 사용하는 경우 특히 그러합니다. 권한이 없는 클라이언트는 합법적인 클라이언트와 디렉터리 서버 간에 교환된 정보를 도청할 수도 있습니다.
-

무단 액세스는 회사 내부에서 발생하거나 회사 외부에서 엑스트라넷 또는 인터넷에 연결된 경우 발생할 수 있습니다.

Directory Server에서 제공하는 인증 방법, 암호 정책 및 액세스 제어 메커니즘은 무단 액세스를 방지하는 효율적인 방법을 제공합니다.

추가 리소스

- 적절한 인증 방법 선택
- 암호 정책 설계
- 액세스 제어 설계

7.1.2. 무단 변조

침입자가 디렉터리에 액세스하거나 Directory Server와 클라이언트 애플리케이션 간의 통신을 가로채는 경우 디렉터리 데이터를 수정하거나 변조할 가능성이 있습니다. 클라이언트가 데이터를 신뢰하지 않거나 디렉터리 자체에서 수정 사항을 신뢰할 수 없는 경우 디렉터리 서비스는 쓸모 없습니다.

예를 들어 디렉터리가 변조를 감지할 수 없는 경우 공격자는 클라이언트 요청을 서버로 변경하거나 이를 전달하지 않고 클라이언트에 서버 응답을 변경할 수 있습니다. TLS 및 유사한 기술은 연결 종료 시 정보에 서명하여 이 문제를 해결할 수 있습니다.

추가 리소스

- Directory Server에서 TLS를 사용하는 방법에 대한 자세한 내용은 서버 연결 보안을참조하십시오.

7.1.3. 서비스 거부

서비스 거부 공격에서 공격자는 디렉터리가 클라이언트에 서비스를 제공하지 못하도록 하는 것입니다. 예를 들어 공격자는 모든 시스템 리소스를 사용할 수 있으므로 다른 사용자가 이러한 리소스를 사용하지 못하도록 할 수 있습니다. Directory Server는 특정 바인딩 DN에 할당된 리소스에 대한 제한을 설정하여 서비스 거부 공격을 방지할 수 있습니다. 사용자 바인딩 DN을 기반으로 리소스 제한 설정에 대한 자세한 내용은 사용자 관리 및 인증 가이드를 참조하십시오.

7.2. 보안 요구 사항 분석

환경과 사용자를 분석하여 특정 보안 요구 사항을 파악합니다. 보안 디렉터리 설계 장의 사이트 설문 조사에서는 디렉터리에 있는 개별 데이터를 읽고 쓸 수 있는 사람에 대한 몇 가지 기본 결정을 명확히 합니다. 이 정보는 보안 설계의 기초가 됩니다.

비즈니스 지원에 디렉터리 서비스를 사용하는 방법은 보안 구현 방법을 정의합니다. 인트라넷을 제공하는 디렉터리에는 인터넷에 열려 있는 엑스트라넷 또는 전자 상거래 애플리케이션을 지원하는 디렉터리와 동일한 보안 조치가 필요하지 않습니다.

디렉터리가 인트라넷만 제공하는 경우 정보에 필요한 액세스 수준을 고려하십시오.

- 사용자와 애플리케이션에 작업을 수행하는 데 필요한 정보에 대한 액세스 권한을 제공하는 방법.
- 일반 액세스로부터 직원 또는 비즈니스에 대한 민감한 데이터를 보호하는 방법

디렉터리가 엑스트라넷을 제공하거나 인터넷을 통해 전자 상거래 애플리케이션을 지원하는 경우 다음과 같은 추가 지점을 고려하십시오.

- 고객에게 개인 정보 보호 보장을 제공하는 방법
- 정보의 무결성을 보장하는 방법

7.2.1. 액세스 권한 확인

데이터 분석은 사용자, 그룹, 파트너, 고객 및 애플리케이션이 디렉터리 서비스에 액세스해야 하는 정보를 식별합니다. 액세스 권한은 다음 두 가지 방법 중 하나로 부여할 수 있습니다.

- 중요한 데이터를 보호하면서 가능한 한 많은 권한을 모든 사용자에게 부여합니다.

  오픈 방법을 사용하려면 비즈니스에 민감하거나 중요한 데이터를 정확하게 결정해야 합니다.

- 각 범주의 사용자에게 작업을 수행하는 데 필요한 최소 액세스 권한을 부여합니다.

  제한적인 방법을 사용하려면 조직의 내부 및 외부에서 각 범주의 정보 요구 사항을 자세히 파악해야 합니다.

액세스 권한을 결정하는 데 사용되는 방법과 관계없이 조직의 사용자 범주와 각각에 부여된 액세스 권한을 나열하는 간단한 테이블을 만듭니다. 디렉터리에 보관된 중요한 데이터를 나열하고 각 데이터 조각에 대해 해당 데이터를 보호하는 단계를 나열하는 테이블을 만드는 것이 좋습니다.

추가 리소스

- 사용자 ID를 확인하는 방법에 대한 자세한 내용은 적절한 인증 방법 선택을 참조하십시오.
- 디렉터리 정보에 대한 액세스 제한에 대한 자세한 내용은 액세스 제어 설계 섹션을 참조하십시오.

7.2.2. 데이터 개인 정보 및 무결성 보장

이 디렉토리를 사용하여 엑스트라넷을 통해 비즈니스 파트너와의 교환을 지원하거나 인터넷상의 고객과 함께 전자 상거래 애플리케이션을 지원하는 경우, 교환된 데이터의 개인 정보 보호 및 무결성을 확인하십시오.

데이터 개인 정보 및 무결성을 보장하기 위해 다음 방법을 사용하십시오.

- 데이터 전송 암호화.
- 인증서를 사용하여 데이터 전송에 서명합니다.

추가 리소스

- Directory Server에서 제공하는 암호화 방법에 대한 자세한 내용은 Password Storage Schemes섹션을 참조하십시오.
- 서명 데이터에 대한 자세한 내용은 서버 연결 보안 섹션을 참조하십시오.

- Directory Server 데이터베이스에서 중요한 정보를 암호화하는 방법에 대한 자세한 내용은 데이터베이스 암호화 섹션을 참조하십시오.

7.2.3. 정기적인 감사 수행

추가 보안 조치로 로그 파일 및 SNMP 에이전트가 기록하는 정보를 검사하여 전체 보안 정책의 효율성을 확인하기 위해 정기적인 감사를 수행합니다.

추가 리소스

- 디렉터리 서버 모니터링에 대한 자세한 내용은 서버 및 데이터베이스 활동 모니터링을 참조하십시오.
- 로그 파일에 대한 자세한 내용은 로그 파일 참조를참조하십시오.

7.2.4. 보안 요구 사항 분석 예

예에서는 imaginary Cryostat 회사 example.com 이 보안 요구 사항을 분석하는 방법을 보여줍니다. example.com 은 웹 호스팅 및 인터넷 액세스를 제공합니다. example.com 활동 중 일부는 클라이언트 회사의 디렉터리를 호스팅하는 것입니다. 또한 여러 개별 구독자에게 인터넷 액세스를 제공합니다. 따라서 example.com 의 디렉터리에는 다음 세 가지 주요 카테고리가 있습니다.

- example.com 내부 정보
- 기업 고객에게 속한 정보
- 개별 구독자와 관련된 정보

example.com 에는 다음과 같은 액세스 제어가 필요합니다.

- 호스트 회사의 디렉터리 관리자(예: example_a 및 example_b)에 대한 액세스 권한을 자체 디렉터리 정보에 제공합니다.

- 호스팅 회사 디렉터리 정보에 대한 액세스 제어 정책을 구현합니다.
- 홈에서 인터넷 액세스를 위해 **example.com** 을 사용하는 모든 개별 클라이언트에 대한 표준 액세스 제어 정책을 구현합니다.
- **example.com** 회사 디렉토리에 대한 액세스를 모든 외부인에게 거부합니다.
- 전 세계에 구독자의 **example.com** 디렉토리에 대한 읽기 액세스 권한을 부여합니다.

7.3. 보안 방법 개요

Directory Server는 특정 요구에 맞는 전체 보안 정책을 설계하는 여러 가지 방법을 제공합니다. 보안 정책은 권한이 없는 사용자가 민감한 정보를 수정하거나 검색하지 못하도록 할 만큼 강력해야 하지만 쉽게 관리할 수 있을 만큼 단순해야 합니다. 복잡한 보안 정책으로 인해 사용자가 액세스하는 데 필요한 정보에 액세스하지 못하거나 더 나빠서 사용자가 액세스할 수 없는 디렉토리 정보를 수정하거나 검색할 수 있는 오류가 발생할 수 있습니다.

표 7.1. Directory Server에서 사용 가능한 보안 방법

보안 방법	설명
인증	다른 당사자의 ID를 확인합니다. 예를 들어 클라이언트는 LDAP 바인딩 작업 중에 Directory Server에 암호를 제공합니다.
암호 정책	이 암호를 유효하게 간주하기 위해 암호가 충족되어야 하는 기준을 정의합니다. 예를 들면 age, length, syntax 입니다.
Encryption	정보의 개인 정보를 보호합니다. 데이터가 암호화되면 수신자만 데이터를 이해할 수 있습니다.
액세스 제어	다른 디렉터리 사용자에게 부여된 액세스 권한을 조정하고 필요한 인증 정보 또는 바인딩 속성을 지정하는 방법을 제공합니다.
계정 비활성화	Directory Server가 모든 인증 시도를 자동으로 거부하도록 사용자 계정, 계정 그룹 또는 전체 도메인을 비활성화합니다.

보안 방법	설명
보안 연결	TLS, StartTLS 또는 SASL을 사용한 연결을 암호화하여 정보의 무결성을 유지 관리합니다. 전송 중에 정보가 암호화되면 수신자는 전송 중에 수정되지 않았음을 확인할 수 있습니다. 최소 보안 강도 요인을 설정하여 보안 연결이 필요할 수 있습니다.
감사	디렉터리 보안이 손상되었는지 확인합니다. 간단한 감사 방법 중 하나는 디렉터리가 유지 관리하는 로그 파일을 검토하는 것입니다.
SELinux	Red Hat Directory Server 시스템에 대한 보안 정책을 사용하여 Directory Server 파일 및 프로세스에 대한 액세스를 제한 및 제어합니다.

보안 설계의 보안을 유지 관리하기 위한 여러 툴을 결합하고 복제 및 데이터 배포와 같은 디렉터리 서비스의 다른 기능을 통합하여 보안 설계를 지원합니다.

7.4. 적절한 인증 방법 선택

보안 정책에 대한 기본 결정은 사용자가 디렉터리에 액세스하는 방법입니다. 익명 사용자가 디렉터리에 액세스할 수 있습니까, 아니면 사용자 이름과 암호(authenticate)를 사용하여 디렉터리에 로그인하는 데 필요한 모든 사용자가 있습니까?

Directory Server에서 제공하는 인증 방법에 대해 알아봅니다. 디렉터리는 사람이든 LDAP 인식 애플리케이션이든 관계없이 모든 사용자에게 동일한 인증 메커니즘을 사용합니다.

7.4.1. 익명 및 인증되지 않은 액세스

익명 액세스를 통해 디렉터리에 대한 가장 쉬운 형태의 액세스 권한을 제공합니다. 익명 액세스를 사용하면 디렉터리에 연결하는 모든 사용자가 데이터에 액세스할 수 있습니다.

익명 액세스를 구성하면 사용자가 검색을 수행하는 경우에만 어떤 종류의 검색을 수행하는지 추적할 수 없습니다. 특정 사용자 또는 사용자 그룹이 일부 종류의 디렉토리 데이터에 액세스하지 못하도록 차단할 수 있지만, 익명 액세스가 해당 데이터에 대해 허용되는 경우 해당 사용자는 디렉터리에 직접 바인딩하여 해당 데이터에 계속 액세스할 수 있습니다.

익명 액세스를 제한할 수 있습니다. 일반적으로 디렉터리 관리자는 쓰기, 추가, 삭제 또는 자체 쓰기 권한이 아닌 읽기, 검색 및 비교 권한에 대한 익명 액세스만 허용합니다. 관리자는 이름, 전화 번호 및 이메일

주소와 같은 일반 정보가 포함된 속성 하위 집합에 대한 액세스를 제한하는 경우가 많습니다. 정부 식별 번호와 같은 보다 민감한 데이터에 대한 익명 액세스를 허용해서는 안 됩니다 (예: 미국 사회 보안 번호, 가정 전화 번호 및 주소, 급여 정보).

디렉터리 데이터에 액세스하는 사용자에 대한 규칙을 강화해야 하는 경우 익명 액세스를 완전히 비활성화할 수 있습니다.

인증되지 않은 바인딩 은 사용자가 사용자 이름으로 바인딩하려고 하지만 사용자 암호 속성이 없는 경우입니다. 예를 들면 다음과 같습니다.

```
ldapsearch -x -D "cn=jsmith,ou=people,dc=example,dc=com" -b "dc=example,dc=com" "
(cn=joe)"
```

사용자가 암호를 제공하지 않는 경우 **Directory Server**는 익명 액세스 권한을 부여합니다. 인증되지 않은 바인딩에서는 바인딩 **DN**이 기존 항목일 필요가 없습니다.

익명 바인딩과 마찬가지로 인증되지 않은 바인딩을 비활성화하여 데이터베이스에 대한 액세스를 제한하여 보안을 강화할 수 있습니다. 또한 인증되지 않은 바인딩을 비활성화하여 클라이언트에 대한 자동 바인딩 실패를 방지할 수 있습니다. 일부 애플리케이션은 실제로 암호가 전달되지 않고 인증되지 않은 바인딩과 연결할 때 바인딩 성공 메시지를 수신했기 때문에 디렉터리에 성공적으로 인증되었다고 생각할 수 있습니다.

7.4.2. 간단한 바인딩 및 보안 바인딩

익명 액세스가 허용되지 않는 경우 사용자는 디렉터리 콘텐츠에 액세스하기 전에 디렉터리에 대한 인증을 받아야 합니다. 간단한 암호 인증을 사용하여 클라이언트는 재사용 가능한 암호를 전송하여 서버에 인증합니다.

예를 들어 클라이언트는 고유 이름과 인증 정보 집합을 제공하는 **bind** 작업을 사용하여 디렉터리에 인증합니다. 서버는 디렉터리에서 클라이언트 **DN**에 해당하는 항목을 찾고 클라이언트에서 제공한 암호가 해당 항목과 저장된 값과 일치하는지 확인합니다. 이 경우 서버는 클라이언트를 인증합니다. 그렇지 않으면 인증 작업이 실패하고 클라이언트에 오류 메시지가 표시됩니다.

바인딩 **DN**은 종종 사람의 입력에 해당합니다. 그러나 일부 디렉터리 관리자는 사람이 아닌 조직 항목으로 바인딩하는 것을 선호합니다. 디렉터리에는 **userPassword** 특성을 허용하는 오브젝트 클래스를 바인딩하는 데 사용하는 항목이 필요합니다. 이렇게 하면 디렉터리가 바인딩 **DN** 및 암호를 인식합니다.

대부분의 **LDAP** 클라이언트는 사용자가 이해하기 어려운 **DN** 문자의 긴 문자열을 찾을 수 있기 때문에 사용자의 바인딩 **DN**을 숨깁니다. 클라이언트에서 사용자의 바인딩 **DN**을 숨기려고 하면 다음 바인딩 알

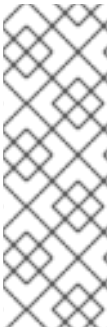
고리즘을 사용합니다.

1. 사용자는 사용자 ID와 같은 고유 식별자를 입력합니다. 예를 들면 fchen 입니다.

2. LDAP 클라이언트 애플리케이션에서 해당 ID의 디렉터리를 검색하고 관련 고유 이름을 반환합니다. 예를 들면 uid=fchen,ou=people,dc=example,dc=com 입니다.

3. LDAP 클라이언트 애플리케이션은 검색된 고유 이름과 사용자가 제공하는 암호를 사용하여 디렉터리에 바인딩됩니다.

간단한 암호 인증은 사용자를 인증하는 쉬운 방법을 제공하지만 추가 보안 방법이 필요합니다. 조직 인트라넷에 대한 사용을 제한하는 것이 좋습니다. 엑스트라넷을 통해 비즈니스 파트너 간 연결 또는 인터넷 상의 고객과의 전송을 위해서는 안전한(암호화) 연결이 필요한 것이 가장 좋습니다.



참고

간단한 암호 인증의 단점은 암호가 일반 텍스트로 전송된다는 것입니다. 권한이 없는 사용자가 수신 대기 중인 경우 해당 사용자가 권한 있는 사용자를 가장할 수 있으므로 디렉터리의 보안이 손상될 수 있습니다. nsslapd-require-secure-binds 구성 속성에는 TLS 또는 Start TLS를 사용하여 보안 연결을 통해 간단한 암호 인증이 필요합니다. 이렇게 하면 일반 텍스트 암호를 효과적으로 암호화하므로 악의적인 행위자가 스니핑할 수 없습니다.

nsslapd-require-secure-binds 구성 속성을 사용하여 TLS 또는 Start TLS를 사용하여 보안 연결을 설정합니다. SASL 인증 또는 인증서 기반 인증도 가능합니다. Directory Server 및 클라이언트 애플리케이션이 서로 보안 연결을 설정하는 경우 클라이언트는 일반 텍스트로 암호를 전송하지 않고 추가 수준의 보호로 간단한 바인딩을 수행합니다.

추가 리소스

- 서버 연결 보안

- nsslapd-require-secure-binds 구성 특성 설명.

7.4.3. 인증서 기반 인증

다른 형식의 디렉터리 인증에는 디지털 인증서를 사용하여 디렉터리에 바인딩해야 합니다. 디렉터리에 처음 액세스할 때 사용자에게 암호를 입력하라는 메시지가 표시됩니다. 그러나 디렉터리에 저장된 암

호와 일치하지 않고 암호는 사용자 인증서 데이터베이스를 엽니다.

사용자가 올바른 암호를 제공하는 경우 디렉터리 클라이언트 애플리케이션은 인증서 데이터베이스에서 인증 정보를 가져옵니다. 그런 다음 클라이언트 애플리케이션 및 디렉터리는 이 정보를 사용하여 사용자 인증서를 디렉터리 **DN**에 매핑하여 사용자를 식별합니다. 디렉터리는 이 인증 프로세스 중에 식별된 디렉터리 **DN**을 기반으로 액세스를 허용하거나 거부합니다.

추가 리소스

- Red Hat Directory Server 보안

7.4.4. 프록시 인증

디렉터리에 대한 액세스를 요청하는 사용자가 자체 **DN**과 바인딩되지 않지만 프록시 **DN**과 바인딩되므로 *프록시* 인증은 특별한 형식의 인증입니다.

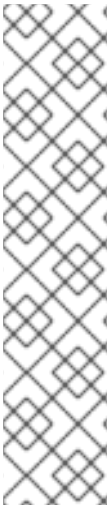
프록시 **DN**은 사용자가 요청하는 작업을 수행할 수 있는 적절한 권한이 있는 엔티티입니다. 사용자 또는 애플리케이션에서 프록시 권한을 수신하는 경우 **Directory Manager DN**을 제외하고 모든 **DN**을 프록시 **DN**으로 지정할 수 있습니다.

프록시 권한의 주요 이점 중 하나는 **LDAP** 애플리케이션에서 디렉터리 서버에 대한 요청을 수행하는 여러 사용자를 서비스하는 단일 스레드를 사용할 수 있다는 것입니다. 각 사용자에 대해 바인딩하고 인증하는 대신 클라이언트 애플리케이션은 프록시 **DN**을 사용하여 **Directory Server**에 바인딩합니다.

프록시 **DN**은 클라이언트 애플리케이션이 제출하는 **LDAP** 작업에 지정됩니다. 예를 들면 다음과 같습니다.

```
ldapmodify -D "cn=Directory Manager" -W -H ldap://server.example.com -X
"dn:cn=joe,dc=example,dc=com" -f mods.ldif
```

이 명령을 사용하면 관리자 항목 **cn=Directory Manager** 에서 **cn=joe** 사용자 권한을 수신하여 **mods.ldif** 파일에 수정 사항을 적용합니다. 관리자는 이러한 변경을 위해 사용자 암호를 제공할 필요가 없습니다.



참고

프록시 메커니즘은 매우 강력하며 신중하게 사용해야 합니다. 프록시 권한은 **ACL**(액세스 제어 목록) 범위 내에 부여되며 사용자에게 프록시 권한을 부여할 때 이 사용자는 대상 아래의 모든 사용자에 대해 프록시할 수 있습니다. 특정 사용자에게만 프록시 권한을 제한할 수 없습니다.

예를 들어 항목에 **dc=example,dc=com** 트리에 대한 프록시 권한이 있는 경우 이 항목은 모든 작업을 수행할 수 있습니다. 따라서 디렉터리의 가능한 가장 낮은 수준에서 프록시 액세스 제어 명령(**ACI**)을 설정해야 합니다.

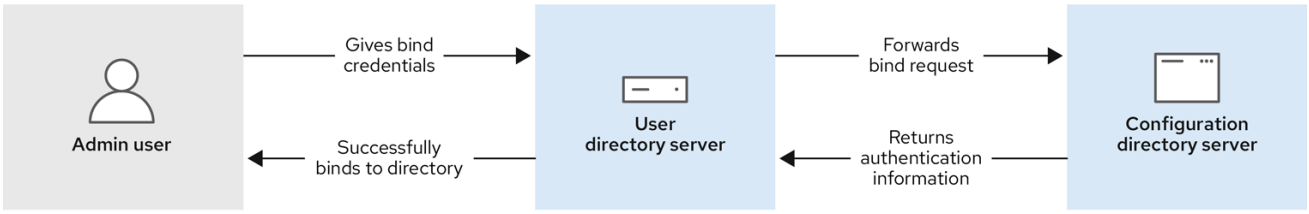
추가 리소스

- 액세스 제어 관리

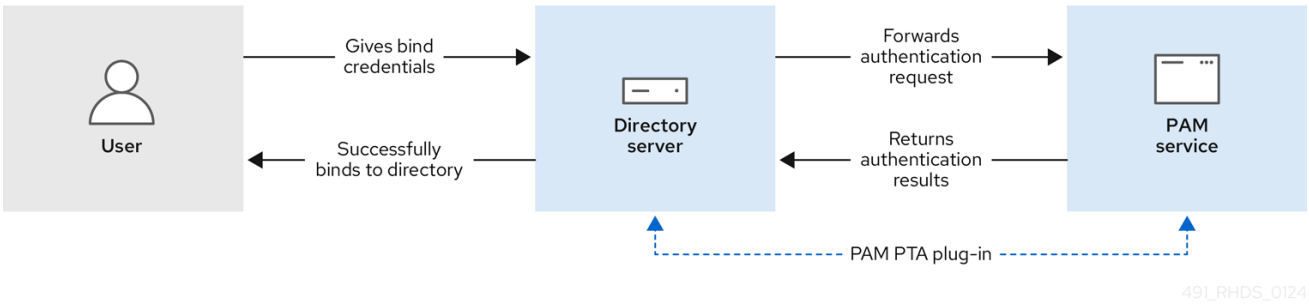
7.4.5. Pass-through 인증 (PTA)

PTA(Pass-through Authentication) 는 **Directory Server**가 한 서버에서 다른 서버로 인증 요청을 전달하는 경우입니다.

예를 들어 **Directory Server**가 다른 디렉터리 인스턴스에 대한 모든 구성 정보를 저장할 때 **Directory Server**는 사용자 디렉터리 서버에 대한 패스스루 인증을 사용하여 구성 디렉터리 서버에 연결합니다. **PTA** 플러그인은 **Directory Server-to-Directory Server pass-through** 인증을 처리합니다.



많은 시스템에는 이미 **PAM**(**Pluggable Authentication Modules**)과 같은 **Unix** 및 **Linux** 사용자를 위한 인증 메커니즘이 있습니다. **LDAP** 클라이언트에 기존 인증 저장소를 사용하도록 **Directory Server**에 지시하도록 **PAM** 모듈을 구성할 수 있습니다. 디렉터리 서버는 **PAM** 서비스와 상호 작용하여 **PAM** 통과 인증 플러그인을 사용하여 **LDAP** 클라이언트를 인증합니다.



PAM 통과 인증을 사용하여 사용자가 **Directory Server**에 바인딩하려고 하면 디렉터리 서버는 자격 증명을 **PAM** 서비스에 전달합니다. 인증 정보가 **PAM** 서비스의 정보와 일치하는 경우 사용자는 모든 **Directory Server** 액세스 제어 제한 및 계정 설정을 사용하여 디렉터리 서버에 성공적으로 바인딩할 수 있습니다.

참고

PAM을 사용하도록 **Directory Server**를 구성할 수 있지만 인증에 **Directory Server**를 사용하도록 **PAM**을 구성할 수는 없습니다.

SSSD(System Security Services Daemon)를 사용하여 **PAM** 서비스를 구성할 수 있습니다. **PAM** 패스스루 인증 플러그인은 기본적으로 **/etc/pam.d/system-auth** 와 같이 **SSSD**에서 사용하는 **PAM** 파일을 가리키기만 하면 됩니다. **SSSD**는 **Active Directory, Red Hat Directory Server** 또는 **OpenLDAP**와 같은 기타 디렉토리 또는 로컬 시스템 설정을 포함하여 다양한 **ID** 공급자를 사용할 수 있습니다.

7.4.6. 암호 없는 인증

인증 시도는 먼저 사용자 계정을 인증할 수 있는지 평가합니다. 계정은 다음 기준에 속해야 합니다.

- 활성화되어야 합니다.
- 잠길 수 없습니다.
- 적용 가능한 암호 정책에 따라 유효한 암호가 있어야 합니다.

경우에 따라 사용자가 **Active Directory Server**에 바인딩하거나 바인딩할 수 없을 때 사용자 계정 인증을 수행해야 하는 경우가 있습니다. 예를 들어 시스템은 **PAM**을 사용하여 시스템 계정을 관리하고, **LDAP** 디렉터리를 **ID** 저장소로 사용하도록 **PAM**을 구성할 수 있습니다. 그러나 시스템은 **SSH** 키 또는

RSA 토큰과 같은 암호 없는 자격 증명을 사용하며 해당 인증 정보를 전달하여 Directory Server에 인증할 수 없습니다.

Red Hat Directory Server는 LDAP 검색에 *대한 계정 지원 확장 제어* 확장을 지원합니다. 이 확장은 계정 상태를 제공하는 반환된 각 항목에 대한 추가 행과 해당 계정의 암호 정책에 대한 몇 가지 정보를 반환합니다. 그러면 클라이언트 또는 애플리케이션에서 해당 사용자 계정에 대한 디렉터리 서버 외부의 인증 시도를 평가하는 데 해당 상태를 사용할 수 있습니다. 기본적으로 이 제어는 인증 작업을 수행할 필요 없이 사용자가 인증할 수 있는지 여부를 나타냅니다.

또한 PAM과 같은 시스템 수준 서비스와 함께 이 확장을 사용하여 Directory Server를 사용하여 ID를 저장하고 계정 상태를 제어하는 암호 없는 로그인을 허용할 수 있습니다.

참고

기본적으로 디렉터리 관리자만 계정 가용성 확장 제어를 사용할 수 있습니다. 다른 사용자가 컨트롤을 사용할 수 있도록 하려면 지원되는 제어 항목인 **oid=1.3.6.1.4.1.42.2.27.9.5.8,cn=features,cn=config** 에서 적절한 **ACI**를 설정합니다.

추가 리소스

- 암호 없는 액세스를 위한 계정 가용성 확인

7.5. 계정 잠금 정책 설계

계정 잠금 정책은 디렉터리에 대한 무단 액세스 또는 손상된 액세스를 방지하여 디렉터리 데이터와 사용자 암호를 모두 보호할 수 있습니다. Directory Server를 잠그거나 *비활성화* 한 후에는 해당 사용자가 디렉터리에 바인딩할 수 없으며 모든 인증 작업이 실패합니다.

nsAccountLock 작동 속성을 사용하여 계정 비활성화를 구현합니다. 항목에 **true** 값이 있는 **nsAccountLock** 속성이 포함된 경우 서버는 해당 계정의 바인딩 시도를 거부합니다.

Directory Server는 자동 기준에 따라 계정 잠금 정책을 정의할 수 있습니다.

- Directory Server는 계정 잠금 정책을 암호 정책과 연결할 수 있습니다. 사용자가 지정된 횟수 후에 적절한 인증 정보를 사용하여 로그인하지 못하면 관리자가 수동으로 잠금 해제할 때까지 Directory Server에서 계정을 잠급니다.

  이러한 정책은 사용자 암호를 반복적으로 추측하여 디렉토리에 침입하려고 시도하는 악의적

인 행위자로부터 보호합니다.

- **Directory Server**는 전달된 일정 시간 후에 계정을 잠글 수 있습니다. 이 정책을 사용하여 계정이 생성된 시간을 기반으로 시간 제한 액세스 권한이 있는 임시 사용자(예: 인턴, 학회 작업자)에 대한 액세스를 제어할 수 있습니다. 또는 마지막 로그인 시간 이후 계정이 일정 시간 동안 비활성 상태인 경우 사용자 계정을 비활성화하는 계정 정책을 만들 수 있습니다.

  계정 정책 플러그인을 사용하여 시간 기반 계정 잠금 정책을 구현하고 디렉터리에 대한 글로벌 설정을 설정합니다. 다른 만료 시간 및 유형에 대해 여러 계정 정책 하위 항목을 생성한 다음 서비스 클래스를 통해 이러한 정책을 항목에 적용할 수 있습니다.

추가 리소스

- 암호 정책 설계

7.6. 암호 정책 설계

암호 정책은 지정된 시스템에서 암호를 사용하는 방법을 관리하는 규칙 집합입니다. **Directory Server** 암호 정책은 암호가 기간, 길이 및 사용자가 암호를 재사용할 수 있는지 여부와 같이 유효한 것으로 간주해야 하는 기준을 지정합니다.

7.6.1. 암호 정책이 작동하는 방식

Directory Server는 세분화된 암호 정책을 지원합니다. 즉 디렉터리 서버는 디렉터리 트리의 모든 지점에서 암호 정책을 정의합니다. **Directory Server**는 다음 수준에서 암호 정책을 정의합니다.

전체 디렉터리

이러한 정책을 *글로벌* 암호 정책이라고 합니다. 이 정책을 구성하고 활성화하면 **Directory Manager** 항목 및 로컬 암호 정책이 활성화된 해당 사용자 항목을 제외하고 디렉터리 내의 모든 사용자에게 이를 적용합니다.

이 정책 유형은 모든 디렉터리 사용자에 대해 공통 단일 암호 정책을 정의할 수 있습니다.

디렉터리의 특정 하위 트리

이러한 정책을 하위 *트리 수준* 또는 *로컬* 암호 정책이라고 합니다. 이 정책을 구성하고 활성화하면 **Directory Server**는 지정된 하위 트리 아래에 있는 모든 사용자에게 적용합니다.

이 정책 유형은 모든 호스팅 회사에 대해 단일 정책을 적용하지 않고 호스팅되는 각 회사에 대해 서로 다른 암호 정책을 지원하는 호스팅 환경에서 유용합니다.

디렉터리의 특정 사용자

이러한 정책을 *사용자 수준* 또는 *로컬* 암호 정책이라고 합니다. 이 정책을 구성하고 활성화하면 **Directory Server**는 지정된 사용자에게만 적용됩니다.

이 정책 유형은 다른 디렉터리 사용자에 대해 다른 암호 정책을 정의할 수 있습니다. 예를 들어 일부 사용자가 암호를 매일 변경하고 일부 사용자는 월간 변경하며 다른 모든 사용자는 6개월마다 암호를 변경하도록 지정합니다.

기본적으로 **Directory Server**에는 글로벌 암호 정책과 관련된 항목 및 속성이 포함되어 있습니다. 즉, 모든 사용자에게 동일한 정책이 적용됩니다. 하위 트리 또는 사용자에 대한 암호 정책을 설정하려면 하위 트리 또는 사용자 수준에서 추가 항목을 추가하고 **cn=config** 항목의 **nsslapd-pwpolicy-local** 속성을 활성화합니다. 이 속성은 스위치 역할을 하며, / **off**의 암호 정책을 세밀하게 설정합니다.

명령줄 또는 웹 콘솔을 사용하여 암호 정책을 변경할 수 있습니다. 명령줄에서 **dsconf pwpolicy** 명령은 글로벌 정책을 변경하고 **dsconf localpwp** 명령은 로컬 정책을 변경합니다. 암호 정책 구성 섹션에서 암호 정책을 설정하는 절차를 확인할 수 있습니다.

암호 정책 확인 프로세스

디렉터리에 추가하는 암호 정책 항목은 **Directory Server**에서 적용해야 하는 암호 정책의 유형(글로벌 또는 로컬)을 결정합니다.

사용자가 디렉터리에 바인딩하려고 하면 **Directory Server**에서 사용자 항목에 대해 로컬 정책이 정의되고 활성화되어 있는지 여부를 결정합니다. **Directory Server**는 다음 순서로 정책 설정을 확인합니다.

1. **Directory Server**는 세분화된 암호 정책이 활성화되어 있는지 여부를 결정합니다. 서버는 **cn=config** 항목에서 **nsslapd-pwpolicy-local** 속성의 값(켜 짐또는)을 확인합니다. 값이 **off** 로 설정된 경우 서버는 하위 트리 및 사용자 수준에 정의된 정책을 무시하고 글로벌 암호 정책을 적용합니다.

2. **Directory Server**는 하위 트리 또는 사용자에 대해 로컬 정책이 정의되었는지 여부를 결정합니다. 서버는 해당 사용자 항목에서 **pwdPolicysubentry** 속성을 확인합니다.

   a. 속성이 있는 경우 서버는 사용자에 대해 구성된 로컬 암호 정책을 적용합니다. 항목에 특성이 있지만 값이 비어 있거나 유효하지 않은 경우(예: 존재하지 않는 항목을 가리킵니다) 서버에서 오류 메시지를 기록합니다.

b.
사용자 항목에 **pwdPolicysubentry** 속성이 없는 경우 서버는 상단에 도달할 때까지 상위 항목, **grandparent** 항목 및 기타 상위 수준 항목을 확인합니다.

c.
상위 수준 항목에서 **pwdPolicysubentry** 속성을 찾을 수 없는 경우 서버는 글로벌 정책을 적용합니다.

3.
서버는 사용자 제공 암호를 사용자 디렉터리 항목에 지정된 값과 비교하여 일치하는지 확인합니다. 또한 서버는 암호 정책에서 정의하는 규칙을 사용하여 사용자가 디렉터리에 바인딩하기 전에 암호가 유효한지 확인합니다.

요청을 바인딩하는 것 외에도 **userPassword** 속성이 요청에 있는 경우 추가 및 수정 작업 중에 암호 정책 확인이 수행됩니다.

userPassword 값을 수정하면 다음 두 가지 암호 정책 설정이 있는지 확인합니다.

- 암호 최소 사용 기간 정책이 활성화됩니다. 최소 사용 기간 요구 사항이 충족되지 않으면 서버에서 **constraintViolation** 오류를 반환합니다. 암호 업데이트 작업이 실패합니다.

- 암호 기록 정책이 활성화됩니다. **userPassword** 속성의 새 값이 암호 기록에 있거나 현재 암호와 동일한 경우 서버에서 제약 조건**Violation** 오류를 반환합니다. 암호 업데이트 작업이 실패합니다.

userPassword 값을 추가하고 수정하면 모두 암호 구문에 대해 설정된 암호 정책을 확인합니다.

- 암호 최소 길이 정책이 활성화됩니다. **userPassword** 속성의 새 값이 필요한 최소 길이보다 작으면 서버는 **constraintViolation** 오류를 반환합니다. 암호 업데이트 작업이 실패합니다.

- 암호 구문 검사 정책이 활성화됩니다. **userPassword** 의 새 값이 항목의 다른 속성과 동일한 경우 서버는 제약 조건**Violation** 오류를 반환합니다. 암호 업데이트 작업이 실패합니다.

7.6.2. 암호 정책 속성

서버의 암호 정책을 생성하는 데 사용할 수 있는 특성에 대해 알아봅니다. **Directory Server**는 **cn=config** 항목에 암호 정책 속성을 저장하고 **dsconf** 유틸리티를 사용하여 이러한 설정을 변경할 수 있

습니다.

최대 실패 수

이 설정은 암호 정책에서 암호 기반 계정 잠금을 활성화합니다. 사용자가 특정 횟수에 로그인하여 실패하는 경우 관리자가 잠금을 해제할 때까지 **Directory Server**는 해당 계정을 잠급니다. **passwordMaxFailure** 구성 매개변수를 사용하여 최대 실패 수를 설정합니다.

Directory Server에는 로그인 시도를 계산하고 로그인 시도가 제한에 도달할 때 계정을 잠그는 두 가지 방법이 있습니다.

- 번호가 적을 때 **Directory Server**가 계정을 잠급니다 (n)
- **Directory Server**는 개수가 초과된 경우에만 계정을 잠급니다(n+1).

예를 들어 실패 제한이 세 번 시도한 경우 세 번째 실패 시도 (n) 또는 네 번째 실패 시도 (n+1)에서 계정을 잠글 수 있습니다. n+1 동작은 **LDAP** 서버의 이전 동작이므로 기존 동작으로 간주됩니다. 최신 **LDAP** 클라이언트는 더 엄격한 하드 제한을 기대합니다. 기본적으로 **Directory Server**는 **strict** 제한(n)을 사용하지만 **passwordLegacyPolicy** 구성 매개변수에서 레거시 동작을 변경할 수 있습니다.

재설정 후 암호 변경

Directory Server 암호 정책은 사용자가 처음 로그인 후 또는 관리자가 암호를 재설정한 후 암호를 변경해야 하는지 여부를 지정할 수 있습니다. 관리자가 설정한 기본 암호는 일반적으로 사용자 **initials**, 사용자 **ID** 또는 회사 이름과 같은 회사 규칙을 따릅니다. 이 규칙이 발견되면 일반적으로 악의적인 행위자가 시스템에 침입하려는 시도에서 사용하는 첫 번째 값입니다. 따라서 관리자가 암호를 재설정한 후 사용자가 암호를 변경해야 합니다.

암호 정책에 대해 이 설정을 구성하는 경우 사용자 정의 암호가 비활성화된 경우에도 암호를 변경해야 합니다. 암호 정책에 필요하지 않거나 사용자가 암호 변경을 허용하지 않는 경우 관리자가 할당한 암호는 명확한 규칙을 따르지 않아야 하며 검색하기 어려울 수 있습니다.

기본 설정에서는 사용자가 암호를 재설정한 후 변경할 필요가 없습니다.

사용자 정의 암호

사용자가 자신의 암호를 변경할 수 있도록 허용하거나 허용하지 않도록 암호 정책을 설정할 수 있습니다. 좋은 암호는 강력한 암호 정책의 키입니다. 좋은 암호는 사전 단어, 가사나 자식의 이름, 달란, 사용자 **ID** 또는 쉽게 찾을 수 있는 사용자에 대한 기타 정보와 같은 간단한 단어를 사용해서는 안 됩니다(또는 디렉토리 자체에 저장). 좋은 암호에는 문자, 숫자 및 특수 문자의 조합이 포함되어야 합니다. 그러나 편의를

위해 사용자가 쉽게 기억할 수 있는 암호를 사용하는 경우가 많습니다. 결과적으로 일부 기업은 강력한 암호 기준을 충족하고 사용자가 암호를 변경할 수 없는 사용자의 암호를 설정하도록 선택합니다.

사용자에 대한 관리자의 암호 설정에는 다음과 같은 단점이 있습니다.

- 관리자 시간이 많이 필요합니다.
- 관리자 지정 암호는 일반적으로 변경하기가 더 어렵기 때문에 사용자가 암호를 아래로 작성할 가능성이 더 많아 검색 위험이 증가합니다.

기본적으로 사용자 정의 암호는 허용됩니다.

암호 만료

암호 정책을 사용하면 사용자가 동일한 암호를 무기한 사용하거나 지정된 시간 후에 암호가 만료되도록 지정할 수 있습니다. 일반적으로 암호가 더 오래 사용할수록 검색될 가능성이 높아집니다. 그러나 암호가 너무 자주 만료되는 경우 사용자가 암호를 기억하는 데 어려움이 있을 수 있으며 암호를 적어 둘 수 있습니다. 일반적인 정책은 **30**일에서 **90**일마다 암호가 만료되는 것입니다. 암호 만료가 비활성화된 경우에도 서버는 암호 만료 사양을 기억합니다. 암호 만료가 다시 활성화된 경우 암호가 마지막으로 비활성화되기 전에 설정된 기간 동안만 유효합니다. 예를 들어 **90**일마다 만료되도록 암호를 구성한 다음 암호 만료를 비활성화하고 다시 활성화하면 기본 암호 만료 기간은 **90**일로 유지됩니다.

기본적으로 사용자 암호는 만료되지 않습니다.

만료 경고

암호 만료 기간을 설정하는 경우 암호가 만료되기 전에 사용자에게 경고를 보내는 것이 좋습니다.

사용자가 서버에 바인딩되면 **Directory Server**가 경고를 표시합니다. 암호 만료가 활성화된 경우 기본적으로 디렉터리 서버는 사용자 암호가 만료되기 직전에 **LDAP** 메시지를 사용하여 사용자에게 경고를 보냅니다. 사용자 클라이언트 애플리케이션에서 이 기능을 지원해야 합니다.

암호 만료 경고에 대한 유효한 범위는 **1**일에서 **2455**일 사이입니다.

참고

암호는 **Directory Server**가 만료 경고를 보낼 때까지 *만료되지 않습니다*.

유예 로그인 제한

만료된 암호의 *유예 기간은* 사용자가 암호가 만료된 경우에도 계속 시스템에 로그인할 수 있음을 의미합니다. 일부 사용자가 만료된 암호를 사용하여 로그인할 수 있도록 하려면 암호가 만료된 후 사용자에게 허용되는 유예 로그인 시도 횟수를 지정합니다.

기본적으로 **Directory Server**는 유예 로그인을 허용하지 않습니다.

암호 구문 확인

암호 구문 검사에서는 암호를 사용하여 특정 기준을 충족하거나 초과하도록 암호 문자열 규칙을 적용합니다. 모든 암호 구문 검사는 하위 트리 또는 사용자별로 전역적으로 적용할 수 있습니다. **passwordCheckSyntax** 속성은 암호 구문 검사를 관리합니다.

기본 암호 구문에는 최소 **8**자의 암호 길이가 필요하며 암호에는 간단한 단어가 사용되지 않습니다. 간단한 단어는 사용자 항목의 **uid,cn,sn,givenName,ou** 또는 **mailattributes** 에 저장된 모든 값입니다.

또한 다른 암호 구문 적용 양식을 사용하여 암호 구문에 대한 다양한 선택적 카테고리를 제공할 수 있습니다.

- 암호에 필요한 최소 문자 수(**passwordMinLength**).
- 최소 숫자 문자 수(**0**에서 **9** 사이의 숫자)입니다(암호**MinDigits**).
- 대문자 및 소문자(**passwordMinAlphas**) 최소 **ASCII** 알파벳 문자 수입니다.
- 최소 대문자 **ASCII** 알파벳 문자 수(**passwordMinUppers**).
- 최소 소문자 **ASCII** 알파벳 문자 수(**passwordMinLowers**).

- !@#$ (암호MinSpecials)와 같은 최소 특수 ASCII 문자 수입니다.
- 최소 8비트 문자 수(passwordMin8bit).
- aaabbb (passwordMaxRepeats)와 같이 동일한 문자를 즉시 반복할 수 있는 최대 횟수입니다.
- 암호가 요구하는 최소 문자 수입니다. 범주는 대문자 또는 소문자, 특수 문자, 숫자 또는 8비트 문자(passwordMinCategories)일 수 있습니다.
- Directory Server는 CrackLib 사전(passwordDictCheck)에 대해 암호를 확인합니다.
- Directory Server는 암호에 palindrome (passwordPalindrome)이 포함되어 있는지 확인합니다.
- Directory Server에서는 동일한 카테고리(passwordMaxClassChars)에서 더 연속 문자가 있는 암호를 설정하지 않습니다.
- Directory Server에서는 특정 문자열(암호BadWords)이 포함된 암호를설정하지 않습니다.
- Directory Server는 관리자 정의 속성(passwordUserAttributes)에 설정된 문자열이 포함된 암호를 설정하지 않습니다.

구문 카테고리가 많을수록 암호가 강화됩니다.

기본적으로 암호 구문 검사는 비활성화되어 있습니다.

암호 길이

암호 정책에는 사용자 암호에 대한 최소 길이가 필요할 수 있습니다. 일반적으로 더 짧은 암호는 해독하기가 더 쉽습니다. 권장되는 최소 암호 길이는 8자입니다. 크래킹하기가 어려울 만큼 길지만 사용자가 암호를 적어 두지 않고도 암호를 기억할 수 있을 만큼 짧습니다. 이 속성에 유효한 값 범위는 2에서 512자 사이입니다.

기본적으로 서버에는 최소 암호 길이가 없습니다.

암호 최소 기간

암호 정책을 사용하면 사용자가 지정된 시간 동안 암호를 변경하지 못하도록 할 수 있습니다. **passwordHistory** 속성과 함께 **passwordMinAge** 속성을 설정하면 사용자가 이전 암호를 재사용할 수 없습니다. 예를 들어 암호 최소 사용 기간(**passwordMinAge**) 속성이 **2**일이면 사용자는 단일 세션 중에 암호를 반복적으로 변경할 수 없습니다. 이렇게 하면 이전 암호를 재사용할 수 있도록 암호 기록을 순환하지 못합니다.

passwordMinAge 속성에 유효한 값 범위는 **0**에서 **24855**일 사이입니다. 값**0(0)**은 사용자가 암호를 즉시 변경할 수 있음을 나타냅니다.

암호 내역

Directory Server는 암호 기록에 **2~24**개의 암호를 저장할 수 있습니다. 암호가 기록에 있는 경우 사용자는 이전 암호로 암호를 재설정할 수 없습니다. 이렇게 하면 사용자가 쉽게 기억할 수 있는 몇 개의 암호를 재사용할 수 없습니다. 또는 암호 기록을 비활성화하여 사용자가 암호를 재사용할 수 있습니다.

암호 기록이 꺼져 있어도 암호는 기록에 남아 있습니다. 암호 기록이 다시 켜지면 암호 기록을 비활성화하기 전에 사용자가 기록에 있는 암호를 재사용할 수 없습니다.

서버는 기본적으로 암호 기록을 유지하지 않습니다.

암호 스토리지 체계

암호 스토리지 스키마는 디렉터리 내에서 **Directory Server** 암호를 저장하는 데 사용되는 암호화 유형을 지정합니다. **Directory Server**는 다양한 암호 스토리지 체계를 지원합니다.

암호 기반 키 비활성화 기능 **2 (PBKDF2_SHA256, PBKDF2-SHA1, PBKDF2-SHA256, PBKDF2-SHA512)**

이는 가장 안전한 암호 스토리지 체계입니다. 기본 스토리지 스키마는 **PBKDF2-SHA512**입니다.

SSHA, SSHA-256, SSHA-384 및 **SSHA-512)**

권장 **SSHA** 스키마는 **SSHA-256** 또는 그 이상입니다.

CLEAR

즉, 암호화가 없으며 **SASL Digest-MD5**와 함께 사용할 수 있는 유일한 옵션이므로 **SASL**을 사용

하려면 **CLEAR** 암호 저장 스키마가 필요합니다. 암호는**ACI**(액세스 제어 정보) 지침을 사용하여 디렉터리 저장소를 보호할 수 있지만 일반 텍스트 암호를 디렉터리에 저장하는 것은 여전히 좋지 않습니다.

보안 해시 알고리즘(**SHA, SHA-256, SHA-384** 및 **SHA-512**)

SSHA보다 안전하지 않습니다.

UNIX CRYPT

이 알고리즘은 **UNIX** 암호와의 호환성을 제공합니다.

MD5

이 스토리지 스키마는 **SSHA**보다 덜 안전하지만 **MD5**가 필요한 레거시 애플리케이션에는 포함되어 있습니다.

Salted MD5

이 스토리지 스키마는 일반 **MD5** 해시보다 안전하지만 **SSHA**보다 안전하지는 않습니다. 이 스토리지 스키마는 새 암호와 함께 사용하기 위해 포함되지 않지만 **Salted MD5**를 지원하는 디렉터리에서 사용자 계정을 마이그레이션하는 데 도움이 됩니다.

암호 마지막 변경 시간

passwordTrackUpdateTime 구성 속성은 **Directory Server**가 항목에 대한 암호를 마지막으로 업데이트할 때 서버에 타임스탬프를 기록하도록 지시합니다. **Directory Server**는 암호 변경 시간을 사용자 항목에 **pwdUpdateTime** 으로 저장합니다. 이는 **modifyTimestamp** 또는 **last Cryostat** 운영 속성과 다릅니다.

기본적으로 서버는 암호를 마지막으로 변경한 시간을 저장하지 않습니다.

추가 리소스

- **cn=config** 항목 아래의 구성 속성

7.6.3. 복제된 환경에서 암호 정책 설계

Directory Server는 다음과 같이 복제된 환경에서 암호 및 계정 잠금 정책을 적용합니다.

- 암호 정책은 데이터 공급자에 적용됩니다.

- 계정 잠금은 복제 설정의 모든 서버에 적용됩니다.

Directory Server는 암호 사용, 계정 잠금 카운터 및 만료 경고 카운터와 같은 디렉터리에 암호 정책 정보를 복제합니다. 그러나 **Directory Server**는 암호 구문 및 암호 변경 기록과 같은 구성 정보를 복제하지 않습니다. **Directory Server**는 이 정보를 로컬에 저장합니다.

복제된 환경에서 암호 정책을 구성할 때 다음 사항을 고려하십시오.

- 모든 복제본은 무제한 암호 만료에 대한 경고를 발행합니다. **Directory Server**는 각 서버에서 이 정보를 로컬로 유지하므로 사용자가 여러 복제본에 바인딩하면 사용자가 동일한 경고를 여러 번 수신합니다. 또한 사용자가 암호를 변경하는 경우 복제본이 이 정보를 수신하는 데 시간이 걸릴 수 있습니다. 사용자가 암호를 변경한 후 즉시 다시 바인딩하면 복제본이 변경 사항을 등록할 때까지 바인딩이 실패할 수 있습니다.

- 공급자 및 복제본을 포함한 모든 서버에서 동일한 바인딩 동작이 발생해야 합니다. 각 서버에서 항상 동일한 암호 정책 구성 정보를 생성합니다.

- 다중 제공 환경에서 계정 잠금 카운터가 예상대로 작동하지 않을 수 있습니다.

7.7. 액세스 제어 설계

인증 체계를 결정한 후 해당 체계를 사용하여 디렉터리에 포함된 정보를 보호하는 방법을 결정합니다. 액세스 제어는 특정 클라이언트가 특정 정보에 액세스할 수 있도록 지정할 수 있지만 다른 클라이언트는 그렇지 않습니다.

액세스 제어를 정의하려면 하나 이상의 ***ACL(****액세스 제어 목록* **)**을 사용합니다. 디렉터리 **ACL**은 읽기, 쓰기, 검색 및 비교와 같은 권한을 허용하거나 거부하는 일련의 **ACI(** *액세스 제어 정보* **)**로 구성됩니다.

ACL을 사용하면 디렉터리 트리의 모든 수준에서 권한을 설정할 수 있습니다.

- 전체 디렉터리

- 디렉터리의 특정 하위 트리

- 디렉터리의 특정 항목
- 특정 항목 특성 세트
- 지정된 LDAP 검색 필터와 일치하는 모든 항목

또한 특정 사용자에 대한 권한, 특정 그룹에 속하는 모든 사용자에 대해 또는 디렉터리의 모든 사용자에 대해 권한을 설정할 수 있습니다. IP 주소(IPv4 또는 IPv6) 또는 DNS 이름과 같은 네트워크 위치에 대한 액세스를 정의할 수 있습니다.

7.7.1. ACI 형식 정보

보안 정책을 설계할 때 디렉터리에 ACI가 표시되는 방법과 설정할 수 있는 권한을 이해해야 합니다.

Directory ACI는 다음과 같은 일반적인 형식을 사용합니다.

> *target permission bind_rule*

ACI 변수에는 다음과 같은 설명이 있습니다.

대상

일반적으로 ACI가 대상으로 하는 하위 트리, 대상 특성 또는 둘 다 항목을 지정합니다. 대상은 ACI가 적용되는 디렉터리 요소를 식별합니다. ACI는 하나의 항목만 대상으로 지정할 수 있지만 여러 속성을 대상으로 지정할 수 있습니다. 또한 대상에 LDAP 검색 필터가 포함될 수 있습니다. 공통 특성 값이 포함된 널리 사용되는 항목에 대한 권한을 설정할 수 있습니다.

권한

ACI가 설정한 실제 권한을 식별합니다. 권한 변수는 ACI가 지정된 대상에 대한 읽기 또는 검색과 같은 특정 유형의 디렉터리 액세스를 허용하거나 거부합니다.

바인딩 규칙

권한이 적용되는 바인딩 DN 또는 네트워크 위치를 식별합니다. 바인딩 규칙은 LDAP 필터를 지정할 수도 있으며 해당 필터가 바인딩 클라이언트 애플리케이션에 대해 true로 평가되면 ACI가 클라이언트 애플리케이션에 적용됩니다.

따라서 디렉터리 오브젝트 대상의 경우 ACI는 바인딩 규칙이 true인 경우 권한을 허용하거나 거부합니다.

권한 및 바인딩 규칙은 쌍으로 설정되며 모든 대상에 여러 권한 바인딩 규칙 쌍이 있을 수 있습니다. 지정된 대상에 대해 여러 액세스 제어를 효과적으로 설정할 수 있습니다. 예를 들면 다음과 같습니다.

```
target (permission bind_rule)(permission bind_rule) ...
```

추가 리소스

- ACI 형식에 대한 전체 설명은 액세스 제어 관리를참조하십시오.

7.7.1.1. 대상

ACI는 항목의 디렉터리 항목 및 속성을 대상으로 지정할 수 있습니다.

디렉터리 항목을 대상으로 지정하면 해당 항목과 권한 범위에 있는 모든 하위 항목이 포함됩니다. ACI에 대한 대상 항목을 명시적으로 정의하지 않으면 ACI가 ACI 문이 포함된 디렉터리 항목을 대상으로 합니다.If you do not explicitly define a target entry for the ACI, then the ACI targets to the directory entry that contains the ACI statement. ACI는 하나의 항목 또는 단일 LDAP 검색 필터와 일치하는 항목만 대상으로 지정할 수 있습니다.

대상 속성은 속성 값의 하위 집합에만 사용 권한을 적용합니다. 일련의 특성을 대상으로 지정하는 경우 ACI 대상 또는 ACI가 명시적으로 대상으로 하지 않는 속성을 지정합니다. 대상의 속성을 제외하면 오브젝트 클래스 구조가 허용하는 몇 가지 특성을 제외한 모든 속성에 대한 권한이 설정됩니다.

추가 리소스

- 디렉터리 항목 대상으로 지정
- 대상 속성

7.7.1.2. 권한

권한은 액세스를 허용하거나 거부할 수 있습니다. 자세한 내용은 액세스 허용 또는 거부를참조하십시오.

권한은 디렉터리 서비스에서 수행되는 모든 작업이 될 수 있습니다.

권한	설명
읽기	사용자가 디렉터리 데이터를 읽을 수 있는지 여부를 나타냅니다.
쓰기	사용자가 디렉터리를 변경하거나 만들 수 있는지 여부를 나타냅니다. 또한 이 권한을 사용하면 사용자가 디렉터리 데이터를 삭제할 수 있지만 항목 자체는 삭제할 수 없습니다. 그러나 전체 항목을 삭제하려면 사용자에게 삭제 권한이 있어야 합니다.
검색	사용자가 디렉터리 데이터를 검색할 수 있는지 여부를 나타냅니다. 읽기 권한의 읽기 권한과는 다릅니다. 이는 사용자가 검색 작업의 일부로 반환되는 경우 디렉터리 데이터를 볼 수 있다는 점에서 읽기 권한과 다릅니다. 예를 들어 일반 이름(**cn**)을 검색하고 개인 실 번호를 읽을 수 있는 경우 Directory Server는 공통 이름 검색의 일부로 방 번호를 반환할 수 있습니다. 그러나 사용자는 검색 제목으로 방 번호를 사용할 수 없습니다. 이 조합을 사용하여 사용자가 특정방에 있는 사람을 검색하지 못하도록 합니다.
비교	사용자가 데이터를 비교할 수 있는지 여부를 나타냅니다. 비교 권한은 검색할 수 있는 기능을 의미하지만 Directory Server는 검색 결과로 실제 디렉터리 정보를 반환하지 않습니다. 대신 Directory Server는 비교된 값이 일치하는지 여부를 나타내는 간단한 부울 값을 반환합니다. 디렉터리 인증 중에 **userPassword** 속성 값과 일치하도록 비교 작업을 사용합니다.
자체 쓰기	그룹 관리에만 자체 쓰기 권한을 사용합니다. 이 권한을 사용하면 사용자가 그룹에 자신을 추가하거나 삭제할 수 있습니다.
add	사용자가 대상 항목 아래에 하위 항목을 만들 수 있는지 여부를 나타냅니다.
delete	사용자가 대상 항목을 삭제할 수 있는지 여부를 나타냅니다.
proxy	사용자가 Directory Manager를 제외한 다른 DN을 사용하여 이 DN의 권한이 있는 디렉터리에 액세스할 수 있음을 나타냅니다.

7.7.1.3. 바인딩 규칙

바인딩 규칙은 **ACI**가 적용되는 바인딩 **DN**(사용자)을 정의합니다. 또한 날짜 또는 **IP** 주소와 같은 바인딩 특성을 지정할 수도 있습니다.

또한 바인딩 규칙은 **ACI**가 사용자 자체 항목에만 적용되도록 쉽게 정의합니다. 사용자는 사용자가 다른 사용자 항목을 업데이트할 위험을 실행하지 않고도 자신의 항목을 업데이트할 수 있습니다.

바인딩 규칙은 **ACI**가 적용되는 경우 다음 상황을 나타냅니다.

- 바인딩 작업이 특정 **IP** 주소(**IPv4** 또는 **IPv6**) 또는 **DNS** 호스트 이름에서 도달하는 경우. 이를 사용하여 지정된 시스템 또는 네트워크 도메인에서 모든 디렉터리 업데이트를 강제로 수행할 수 있습니다.
- 사용자가 **anonymous**를 바인딩하는 경우 익명 바인딩에 대한 권한 설정은 디렉터리에 바인딩하는 모든 사용자에게 권한이 적용됨을 의미합니다.
- 디렉터리에 성공적으로 바인딩하는 모든 사용자의 경우 익명 액세스를 방지하는 동안 일반 액세스를 허용하는 데 사용할 수 있습니다.
- 사용자가 항목의 즉시 상위로 바인딩된 경우
- 사용자가 특정 **LDAP** 검색 기준을 충족하는 경우

Directory Server는 바인딩 규칙에 대해 다음 키워드를 제공합니다.

상위

바인딩 **DN**이 즉시 상위 항목인 경우 바인딩 규칙이 **true**입니다. 디렉터리 항목이 즉시 하위 항목을 관리할 수 있는 특정 권한을 부여할 수 있습니다.

자체

바인딩 **DN**이 액세스를 요청하는 항목과 동일한 경우 바인딩 규칙이 **true**입니다. 개인이 자신의 항목을 업데이트할 수 있도록 특정 권한을 부여할 수 있습니다.

All

바인딩 규칙은 디렉터리에 성공적으로 바인딩된 모든 사용자에게 적용됩니다.

모든 사람

바인딩 규칙은 모든 사람에게 적용됩니다. 익명 액세스를 허용하거나 거부하려면 이 키워드를 사용합니다.

7.7.2. 권한 설정

기본적으로 **Directory Server**는 **Directory Manager**를 제외하고 모든 사용자에 대한 액세스를 거부합니다. 결과적으로 사용자가 디렉터리에 액세스하려면 **ACI**를 설정해야 합니다.

7.7.2.1. 우선순위 규칙

사용자가 디렉터리 항목에 대한 모든 유형의 액세스를 시도하면 **Directory Server**는 디렉터리에 설정된 액세스 제어를 확인합니다. 액세스를 결정하기 위해 디렉터리 서버는 *우선순위 규칙을* 적용합니다. 이 규칙은 두 개의 충돌하는 권한이 있는 경우 액세스를 거부하는 권한은 항상 액세스 권한을 부여하는 권한보다 우선합니다.

예를 들어 디렉터리 서버가 디렉터리 루트 수준의 쓰기 권한을 거부하고 해당 권한이 디렉터리에 액세스하는 모든 사용자에게 적용되는 경우 쓰기 액세스를 허용할 수 있는 기타 권한에 관계없이 사용자는 디렉터리에 쓸 수 없습니다. 특정 사용자가 디렉터리에 대한 권한을 쓸 수 있도록 허용하려면 해당 사용자를 포함하지 않도록 원래의 **deny-for-write**의 범위를 설정해야 합니다. 그런 다음 사용자에 대한 추가 **allow-for-write** 권한을 설정해야 합니다.

7.7.2.2. 액세스 허용 또는 거부

디렉터리 트리에 대한 액세스를 허용하거나 거부할 수 있지만 액세스를 명시적으로 거부해야 합니다. 우선순위 규칙으로 인해 **Directory Server**에서 더 높은 수준의 디렉터리에서 액세스를 거부하는 규칙을 발견하면 액세스 권한을 부여할 수 있는 충돌하는 권한에 관계없이 더 낮은 수준에서 액세스를 거부합니다.

사용자 또는 클라이언트 애플리케이션의 가능한 최소 하위 집합만 포함하도록 허용 액세스 규칙의 범위를 제한합니다. 예를 들어 사용자가 디렉터리 항목의 모든 속성에 쓸 수 있도록 권한을 설정할 수 있지만 **Directory Administrators** 그룹의 멤버를 제외한 모든 사용자는 **uid** 속성에 쓰는 권한을 거부할 수 있습니다.

또는 다음과 같은 방법으로 쓰기 액세스를 허용하는 두 가지 액세스 규칙을 작성합니다.

- **uid** 특성을 제외한 모든 속성에 대한 쓰기 권한을 허용하는 하나의 규칙을 생성합니다. 이 규칙은 모든 사람에게 적용되어야 합니다.

- uid 속성에 대한 쓰기 권한을 허용하는 하나의 규칙을 생성합니다. 이 규칙은 Directory Administrators 그룹의 멤버에게만 적용되어야 합니다.

허용 권한만 제공하면 명시적 거부 권한을 설정할 필요가 없습니다.

7.7.2.3. 액세스를 거부해야 하는 경우

명시적 거부 권한을 설정할 필요는 없지만 다음과 같은 경우에는 유용합니다.

- 복잡한 ACL 분배가 포함된 대규모 디렉터리 트리가 있습니다.

  보안상의 이유로 Directory Server는 특정 사용자, 그룹 또는 물리적 위치에 대한 액세스를 갑자기 거부해야 할 수 있습니다. 기존 ACL을 신중하게 검사하여 허용 권한을 제한하는 방법을 이해하는 대신 분석을 수행할 시간이 있을 때까지 명시적 거부 권한을 일시적으로 설정합니다. ACL이 이러한 복잡성이 되면 거부 ACI는 향후 관리 오버헤드에만 비용을 추가합니다. 가능한 한 빨리 명시적 거부 권한을 방지하기 위해 ACL을 다시 작업한 다음 전체 액세스 제어 체계를 단순화합니다.

- 요일 또는 1시간에 따라 액세스 제어를 설정합니다.

  예를 들어 Directory Server는 일요일 오후 11시부터 모든 쓰기 작업을 거부할 수 있습니다. (2300) - 월요일부터 오전 1:00. (0100). 관리 관점에서 이 유형의 시간 기반 액세스를 명시적으로 제한하는 ACI를 관리하는 것이 모든 allow-for-write ACI에 대해 디렉토리를 통해 검색하고 이 시간 프레임에서 범위를 제한하는 것보다 더 쉽습니다.

- 디렉터리 관리 권한을 여러 사용자에게 위임할 때 권한을 제한합니다.

  사람 또는 사용자 그룹이 트리의 일부 측면을 수정하지 않고 디렉터리 트리의 일부를 관리할 수 있도록 허용하려면 명시적 거부 권한을 사용합니다.

  예를 들어 메일 관리자가 공통 이름(cn) 속성에 대한 쓰기 액세스를 허용하지 않도록 하려면 공통 이름 속성에 대한 쓰기 액세스를 명시적으로 거부하는 ACI를 설정합니다.

7.7.2.4. 액세스 제어 규칙 배치 위치

디렉터리의 모든 항목에 액세스 제어 규칙을 추가할 수 있습니다. 관리자는 종종 오브젝트 클래스 domainComponent,country,organization,organizationalUnit,inetOrgPerson 또는 group 을 사용하여 항목에 액세스 제어 규칙을 추가합니다. ACL 관리를 단순화하기 위해 규칙을 가능한 한 그룹으로 구성합

니다. 규칙은 대상 항목 및 해당 항목 모든 하위 항목에 적용됩니다. 따라서 디렉터리 또는 디렉터리 분기 지점의 루트 지점에 액세스 제어 규칙을 개인 등의 개별 리프 항목에 분산하지 않고 배치하는 것이 가장 좋습니다.

7.7.2.5. 필터링된 액세스 제어 규칙 사용

LDAP 검색 필터를 사용하여 정의된 기준 세트와 일치하는 디렉터리 항목에 대한 액세스를 설정할 수 있습니다. 예를 들어 **Marketing** 으로 설정된 **organizationalUnit** 속성이 포함된 모든 항목에 대해 읽기 액세스를 허용합니다.

필터링된 액세스 제어 규칙을 사용하면 사전 정의된 액세스 수준을 허용합니다. 예를 들어 디렉터리에는 홈 주소 및 전화 번호 정보가 포함되어 있습니다. 어떤 사람들은 이 정보를 공개하고 싶은 반면, 다른 사람들은 이 정보를 공개하고자 합니다.

다음과 같은 방법으로 액세스를 구성할 수 있습니다.

1. **publish HomeContactInfo** 라는 모든 사용자 디렉터리 항목에 속성을 추가합니다.

2. **publishhomeContactInfo** 속성이 **true**로 설정된 항목에 대해서만 **home Cryostat** 및 **homePostalAddress** 속성에 대한 읽기 액세스 권한을 부여하는 액세스 제어 규칙을 설정합니다. **LDAP** 검색 필터를 사용하여 이 규칙의 대상을 표현합니다.

3. 디렉터리 사용자가 자체 **publish HomeContactInfo** 속성 값을 **true** 또는 **false**로 변경할 수 있도록 허용합니다. 이러한 방식으로 디렉터리 사용자는 이 정보를 공개적으로 사용할 수 있는지 여부를 결정할 수 있습니다.

추가 리소스

LDAP 검색 필터

7.7.3. ACI 보기: 유효한 권한 얻기

GER(***Get effective rights*** **)**는 항목 내의 각 속성에 설정된 액세스 제어 권한을 반환하는 확장된 **ldapsearch** 명령입니다. 이 검색을 통해 **LDAP** 클라이언트는 서버 액세스 제어 구성으로 사용자가 수행할 수 있는 작업을 결정할 수 있습니다.

액세스 제어 정보는 두 개의 액세스 그룹, 즉 항목 권한 및 속성 권한으로 나뉩니다. 항목 권한은 해당

특정 항목으로 제한되는 수정 또는 삭제와 같은 권한입니다. 속성 권한은 디렉터리 전체의 해당 속성의 모든 인스턴스에 대한 액세스 권한입니다.

이러한 자세한 액세스 제어는 다음과 같은 상황에서 필요할 수 있습니다.

- GER 명령을 사용하여 디렉터리에 대한 액세스 제어 지침을 보다 효과적으로 구성할 수 있습니다. 다른 그룹에 비해 한 사용자 그룹이 보거나 편집할 수 있는 항목을 제한해야 하는 경우가 많습니다. 예를 들어 Cryostat 관리자 그룹의 멤버는 관리자 및 급여 와 같은 특성을 검색하고 읽을 수 있는 권한이 있을 수 있지만 HR 그룹 멤버만 수정하거나 삭제할 수 있는 권한이 있습니다. 사용자 또는 그룹에 대한 효과적인 권한을 확인하는 것이 관리자가 적절한 액세스 제어를 설정하는지 확인하는 한 가지 방법입니다.

- GER 명령을 사용하여 개인 항목에서 보거나 수정할 수 있는 속성을 확인할 수 있습니다. 예를 들어 사용자는 homePostalAddress 및 cn 과 같은 속성에 액세스할 수 있어야 하지만 manager 및 salary 속성에 대한 읽기 액세스 권한만 있을 수 있습니다.

추가 리소스

- Get Effective Rights 검색을 사용하여 항목에 대한 액세스 권한 확인

- Get Effective Rights 검색에 대한 일반적인 시나리오

7.7.4. ACI 사용: 일부 팁 및 팁

다음 팁은 디렉터리 보안 모델을 관리하고 디렉터리 성능 특성을 개선하는 데 도움이 될 수 있습니다.

- 디렉터리의 ACI 수를 최소화합니다.

  Directory Server는 50,000개 이상의 ACI를 평가할 수 있지만 많은 ACI 문을 관리하기가 어렵습니다. 많은 ACI를 사용하면 사용자 관리자가 특정 클라이언트에서 사용할 수 있는 디렉터리 오브젝트를 즉시 확인할 수 있습니다.

  Directory Server는 매크로를 사용하여 디렉터리의 ACI 수를 최소화합니다. 매크로를 사용하여 ACI 대상 또는 바인딩 규칙 또는 둘 다에서 DN 또는 해당 부분을 나타냅니다.

- 권한 균형을 허용 및 거부합니다.

기본 규칙은 액세스 권한이 특별히 부여되지 않은 모든 사용자에 대한 액세스를 거부하는 것이지만 하나의 **ACI**를 사용하여 트리의 루트에 가까운 액세스 권한과 리프 항목에 가까운 일부 거부 **ACI**를 사용하여 **ACI** 수를 줄이는 것이 좋습니다. 이 시나리오에서는 리프 항목에 가까운 여러 허용 **ACI**를 사용하지 않습니다.

- **ACI**에서 가장 작은 속성 세트를 식별합니다.

  속성 하위 집합에 대한 액세스를 허용하거나 거부할 때 가장 작은 목록이 허용되는 속성 세트 또는 거부되는 속성 집합인지 선택합니다. 그런 다음 가장 작은 목록만 관리해야 하므로 **ACI**를 설정합니다.

  예를 들어 **person** 오브젝트 클래스는 많은 수의 특성을 포함합니다. 사용자가 몇 개의 속성만 업데이트할 수 있도록 하려면 해당 속성에 대한 쓰기 액세스 권한을 허용하는 **ACI**를 작성합니다. 그러나 사용자가 몇 가지 속성을 제외한 모든 속성을 업데이트할 수 있도록 하려면 이름이 몇 가지 특성을 제외한 모든 항목에 대한 쓰기 액세스를 허용하는 **ACI**를 만듭니다.

- **LDAP** 검색 필터를 신중하게 사용하십시오.

  검색 필터는 액세스를 관리하는 오브젝트의 이름을 직접 지정하지 않습니다. 결과적으로 사용으로 예기치 않은 결과가 발생할 수 있습니다. 특히 디렉터리가 더 복잡해지면 더욱 복잡해집니다. **ACI**에서 검색 필터를 사용하기 전에 동일한 필터를 사용하여 **ldapsearch** 작업을 실행하여 결과를 지웁니다.

- 디렉터리 트리의 다른 부분에서 **ACI**를 복제하지 마십시오.

  겹치는 **ACI**를 보호합니다. 예를 들어 디렉터리 루트 지점에는 **commonName** 및 **givenName** 속성에 대한 그룹 쓰기 액세스 권한이 있고 **commonName** 속성에 대해 동일한 그룹 쓰기 액세스 권한을 허용하는 다른 **ACI**가 있는 경우 하나의 컨트롤만 그룹에 대한 쓰기 액세스 권한을 부여하도록 **ACI**를 업데이트하는 것이 좋습니다.

  디렉터리가 점점 복잡해지면 실수로 **ACI**가 중복될 위험이 빠르게 증가합니다. **ACI**가 겹치는 것을 방지하면 디렉터리에 포함된 총 **ACI** 수를 줄임으로써 보안 관리가 더 쉬워집니다.

- 이름 **ACI**.

  ACI 이름을 지정하는 것은 선택 사항이지만 각 **ACI**에 짧고 의미 있는 이름을 지정하는 것은 보안 모델을 관리하는 데 도움이 됩니다.

- 디렉토리 내에서 **ACI**를 가능한 한 밀접하게 함께 그룹화합니다.

  ACI 위치를 디렉터리 루트 지점 및 주요 디렉토리 분기 지점으로 제한하십시오. **ACI** 그룹화는 전체 **ACI** 목록을 관리하고 디렉터리에 있는 총 **ACI** 수를 최소로 유지하는 데 도움이 됩니다.

- 바인딩 **DN**이 **cn=Joe** 와 같지 않은 경우 쓰기 거부와 같은 이중 음수를 사용하지 마십시오.

  이 구문은 서버에 완벽하게 허용되지만 사람이 읽을 수 있는 것은 아닙니다.

추가 리소스

- 매크로 액세스 제어 지침 사용

7.7.5. 루트 DN(Directory Manager)에 ACI 적용

일반적으로 액세스 제어 규칙은 **Directory Manager** 사용자에게 적용되지 않습니다. 디렉터리 관리자는 일반 사용자 데이터베이스가 아닌 **dse.ldif** 파일에 정의되어 있으며 **ACI** 대상에는 해당 사용자가 포함되지 않습니다.

디렉터리 관리자는 유지 관리 작업을 수행하고 사고에 응답하기 위해 높은 수준의 액세스 권한이 필요합니다. 그러나 디렉터리 관리자에 특정 수준의 액세스 제어 권한을 부여하여 권한이 없는 액세스 또는 공격이 **root** 사용자로 수행되지 않도록 할 수 있습니다.

RootDN 액세스 제어 플러그인을 사용하여 **Directory Manager** 사용자와 관련된 특정 액세스 제어 규칙을 설정합니다.

- 특정 요일 및 특정 시간 범위에서 액세스를 허용하거나 거부하는 시간 기반 액세스 제어입니다.
- 정의된 **IP** 주소, 서브넷 및 도메인의 액세스를 허용하거나 거부하는 **IP** 주소 규칙입니다.
- 특정 호스트, 도메인 및 하위 도메인의 액세스를 허용하거나 거부하는 호스트 액세스 규칙입니다.

Directory Manager에 대해 하나의 액세스 제어 규칙만 설정할 수 있습니다. 플러그인 항목에 있으며

전체 디렉터리에 적용됩니다.

중요

Directory Manager 계정에 적절한 수준의 액세스 권한이 있는지 확인합니다. 이 관리 사용자는 시간외로 유지 관리 작업을 수행하거나 오류에 응답해야 할 수 있습니다. 이 경우 너무 제한적인 시간 또는 일 규칙을 설정하면 **Directory Manager** 사용자가 디렉터리를 효과적으로 관리하지 못할 수 있습니다.

추가 리소스

- **Directory Manager** 계정에서 액세스 제어 설정

7.8. 데이터베이스 암호화

데이터베이스는 정보를 일반 텍스트로 저장합니다. 결과적으로 액세스 제어 조치가 정부 식별 번호 또는 암호와 같은 매우 민감한 정보를 충분히 보호하지 못할 수 있습니다. 파일 시스템을 통해 직접 또는 폐기된 디스크 드라이브 또는 아카이브 미디어에 액세스하여 서버 영구 스토리지 파일에 액세스할 수 있습니다.

데이터베이스 암호화를 사용하면 개별 특성이 데이터베이스에 저장되므로 암호화할 수 있습니다. 구성된 경우 인덱스 데이터도 특정 속성의 모든 인스턴스가 암호화되며 **TLS**와 같은 보안 채널을 사용하여 액세스할 수 있습니다.

추가 리소스

- 데이터베이스 암호화 사용에 대한 자세한 내용은 특성 암호화 관리 장을 참조하십시오.

7.9. 서버 연결 보안

식별된 사용자에 대한 인증 체계와 디렉터리에서 정보를 보호하기 위한 액세스 제어 체계를 설계한 후 다음 단계는 서버와 클라이언트 애플리케이션 간에 전달할 때 정보의 무결성을 보호하는 방법을 설계하는 것입니다.

서버 간 연결 및 서버 간 연결 모두에서 디렉터리 서버는 다양한 보안 연결 유형을 지원합니다.

TLS(Transport Layer Security)

Directory Server는 **TLS**를 통해 **LDAP**를 사용하여 네트워크를 통해 보안 통신을 제공할 수 있습

니다. 특정 연결에 대해 선택한 암호화 방법은 클라이언트 애플리케이션과 디렉터리 서버 간의 협상 결과입니다.

TLS 시작

Directory Server는 또한 암호화되지 않은 일반 LDAP 포트를 통해 TLS(Transport Layer Security) 연결을 시작하는 방법인 Start TLS를 지원합니다.

SASL(Simple Authentication and Security Layer)

SASL은 클라이언트 및 서버 애플리케이션 모두에서 활성화되는 메커니즘에 따라 서버에 사용자를 인증하기 위해 다양한 메커니즘을 구성하는 데 사용할 수 있는 보안 프레임워크입니다. 또한 SASL은 클라이언트와 서버 간에 암호화된 세션을 설정할 수 있습니다. Directory Server는 GSS-API와 함께 SASL을 사용하여 Kerberos 로그인을 활성화하고 복제, 체인 및 패스스루 인증을 포함한 거의 모든 서버 간 연결을 활성화합니다. Directory Server는 Windows 동기화와 함께 SASL을 사용할 수 없습니다.

보안 연결은 복제와 같은 민감한 정보를 처리하는 모든 작업에 권장되며 Windows 암호 동기화와 같은 일부 작업에 필요합니다. Directory Server는 TLS 연결, SASL 및 비보안 연결을 동시에 지원할 수 있습니다.

Directory Server는 SASL 인증 및 TLS 연결을 동시에 지원할 수 있습니다. 예를 들어 서버에 TLS 연결이 필요하고 복제 연결에 대한 SASL 인증도 지원하도록 Directory Server 인스턴스를 구성했습니다. 즉, 네트워크 환경에서 TLS 또는 SASL 사용 여부를 선택할 필요가 없습니다.

또한 서버 연결에 대한 최소 수준의 보안을 설정할 수 있습니다. 보안 강도 요인은 키 강도로 보안 연결이 얼마나 강력한지 측정합니다. 암호 변경과 같은 특정 작업이 필요한 ACI를 설정할 수 있습니다. 연결이 특정 강도 이상인 경우에만 발생합니다. 또한 기본적으로 표준 연결을 비활성화하고 모든 연결에 대해 TLS, Start TLS 또는 SASL이 필요한 최소 SSF를 설정할 수 있습니다. Directory Server는 TLS 및 SASL을 동시에 지원하며 서버는 사용 가능한 모든 연결 유형의 SSF를 계산하고 가장 강력한 연결을 선택합니다.

추가 리소스

- TLS, TLS 및 SASL 사용에 대한 자세한 내용은 Red Hat Directory Server 보안을참조하십시오.

7.10. SELINUX 정책 사용

SELinux는 시스템의 애플리케이션, 프로세스 및 파일에 대한 액세스 제어를 정의하는 보안 정책 컬렉션입니다. 보안 정책은 SELinux에 무단 액세스 및 변조를 방지하기 위해 액세스할 수 있거나 액세스할 수 없는 규칙을 SELinux에 알리는 일련의 규칙입니다.

SELinux는 서버의 파일, 디렉터리, 포트, 프로세스, 사용자 및 기타 오브젝트를 분류합니다. SELinux는 각 개체를 적절한 보안 컨텍스트에 배치하여 역할, 사용자 및 보안 수준을 통해 개체가 서버에서 작동하는 방법을 정의합니다. SELinux는 오브젝트에 대한 이러한 역할을 도메인에 그룹화하고, SELinux 규칙은 한 도메인의 오브젝트가 다른 도메인의 오브젝트와 상호 작용하는 방법을 정의합니다.

Directory Server에는 다음과 같은 도메인이 있습니다.

- 디렉터리 서버용 dirsrv_t
- dirsrv_snmp_t for SNMP
- LDAP 포트용 ldap_port_t

이러한 도메인은 Directory Server의 모든 프로세스, 파일, 디렉토리, 포트, 소켓 및 사용자에 대한 보안 컨텍스트를 제공합니다.

- SELinux는 특정 보안 컨텍스트를 사용하여 각 인스턴스의 파일 및 디렉터리에 레이블을 지정합니다. Directory Server에서 사용하는 대부분의 기본 디렉터리에는 수에 관계없이 모든 로컬 인스턴스의 하위 디렉터리가 있으므로 SELinux는 새 인스턴스에 단일 정책을 쉽게 적용할 수 있습니다.
- SELinux는 특정 보안 컨텍스트를 사용하여 각 인스턴스의 포트에 레이블을 지정합니다.
- SELinux는 적절한 도메인 내의 모든 Directory Server 프로세스를 제한합니다.
- 각 도메인에는 도메인에 대해 권한이 부여된 작업을 정의하는 특정 규칙이 있습니다.
- SELinux 정책에서 지정하지 않는 경우 SELinux는 인스턴스에 대한 액세스를 거부합니다.

SELinux에는 다음과 같은 세 가지 수준의 적용 수준이 있습니다.

비활성화됨

SELinux 없음

허용

SELinux 프로세스 규칙은 처리되지만 강제 적용하지는 않습니다.

강제

SELinux는 모든 규칙을 엄격하게 적용합니다.

Red Hat Directory Server는 엄격한 SELinux 강제 모드에서 정상적으로 실행할 수 있는 SELinux 정책을 정의했습니다. Directory Server는 정상적인 작업을 위해 그리고 하나는 가져오기(ldif2db 모드)와 같은 데이터베이스 작업에 대해 다른 모드로 실행할 수 있습니다. Directory Server의 SELinux 정책은 일반 모드에만 적용됩니다.

기본적으로 **Directory Server는 SELinux 정책과 함께 일반 모드에서 실행됩니다.**

추가 리소스

- SELinux의 작동 방식

8장. 디렉터리 설계 예

디렉터리 서비스의 설계는 엔터프라이즈의 크기와 특성에 따라 다릅니다. 다음 예제는 실제 디렉터리 서비스 배포 계획을 개발하기 위한 시작점입니다.

8.1. 로컬 엔터프라이즈 설계 예

ExampleCom은 소규모 부품 제조업체이며 **500** 명의 직원이 있습니다. **ExampleCom**은 사용하는 디렉터리 지원 애플리케이션을 지원하기 위해 **Red Hat Directory Server**를 배포하기로 결정했습니다.

8.1.1. 로컬 엔터프라이즈의 데이터 설계

디렉터리가 저장할 데이터 유형을 결정하기 위해 **ExampleCom**은 사이트 설문 조사를 수행하는 배포 팀을 만듭니다. 배포 팀은 다음 주요 포인트를 결정합니다.

- 메시징 서버, 웹 서버, 일정 서버, 인적 리소스 애플리케이션 및 흰색 페이지 애플리케이션은 이 디렉터리를 사용합니다.
- 메시징 서버는 **uid,mailServerName, mailAddress** 와 같은 속성에 대한 정확한 검색을 수행합니다. 데이터베이스 성능을 개선하기 위해 **ExampleCom**은 이러한 특성에 대한 인덱스를 유지 관리합니다.

  인덱스 사용에 대한 자세한 내용은 인덱스를 사용하여 데이터베이스 성능을 향상합니다.**For more information about using indexes, see Using indexes to improve database performance.**
- 흰색 페이지 애플리케이션은 사용자 이름과 전화 번호를 검색합니다. 따라서 디렉토리는 많은 수의 하위 문자열, 와일드카드 및 유사 항목 검색을 처리하여 많은 결과를 반환해야 합니다. **ExampleCom** 회사는 다음 인덱스를 유지 관리하기로 결정했습니다.
  - **cn ,sn** 및 **givenName** 속성에 대한 ,동일성,대략적인 및 하위 문자열 인덱스입니다.
  - **telephone Number** 특성에 대한 ,같음 및 하위 문자열 인덱스입니다.
- 이 디렉터리는 조직에 배포된 **LDAP** 서버 기반 인트라넷을 지원하기 위해 사용자 및 그룹 정보를 유지해야 합니다. 디렉터리 관리자 그룹은 대부분의 **ExampleCom** 사용자 및 그룹 정보를

관리합니다. 그러나 **ExampleCom**은 별도의 메일 관리자 그룹이 이메일 정보를 관리하고자 합니다.

- 디렉터리는 **S/MIME** 이메일과 같은 **PKI(**공개 키 인프라**)** 애플리케이션을 지원하기 위해 사용자 공개 키 인증서를 저장해야 합니다.

8.1.2. 로컬 엔터프라이즈의 스키마 설계

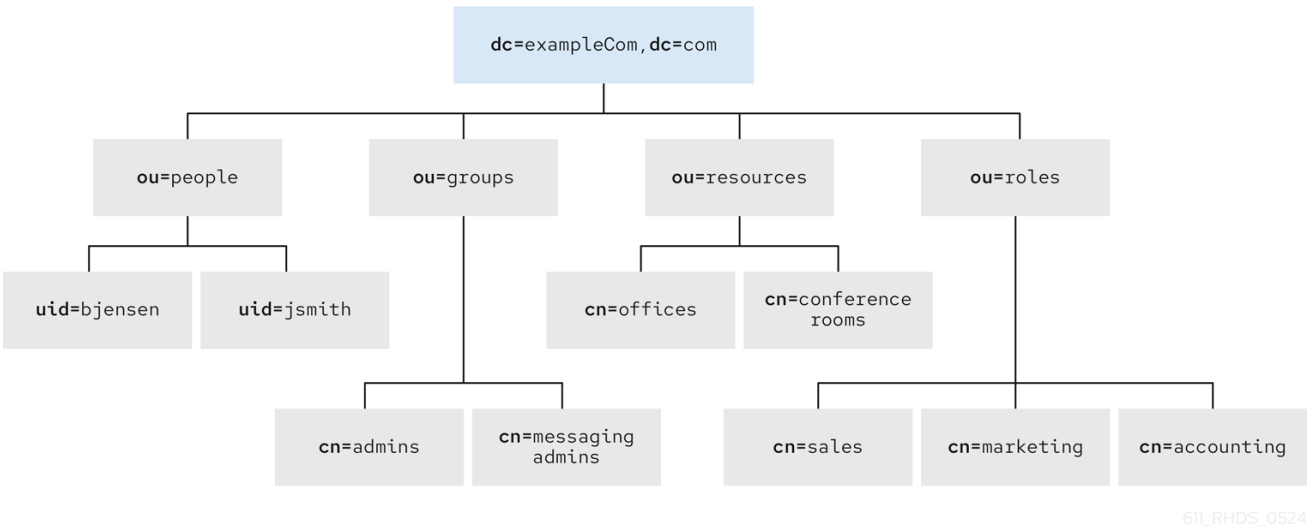
ExampleCom 디렉터리에서 지원하는 애플리케이션에는 **userCertificate** 및 **uid (userID)** 속성이 필요합니다. 따라서 **ExampleCom** 배포 팀은 **inetOrgPerson** 오브젝트 클래스를 사용하여 두 특성을 모두 허용하므로 디렉터리의 항목을 나타냅니다.

또한 **ExampleCom**은 예제**Person** 오브젝트 클래스를 생성하여 직원을 나타내는 기본 디렉터리 스키마를 사용자 지정하려고 합니다. 이 개체 클래스는 **inetOrgPerson** 개체 클래스에서 파생됩니다. **examplePerson** 은 하나의 **exampleID** 속성을 허용합니다. 이 속성에는 각 직원에 할당된 특수 직원 번호가 포함됩니다. 나중에 **ExampleCom**은 **examplePerson** 오브젝트 클래스에 새 속성을 추가할 수 있습니다.

8.1.3. 로컬 엔터프라이즈의 디렉터리 트리 설계

준비된 데이터 및 스키마 설계에 따라 **ExampleCom**은 다음 디렉터리 트리를 생성합니다.

그림 **8.1.** 예제**Com**의 디렉터리 트리



- 디렉터리 트리의 루트는 회사 인터넷 도메인 이름인 **dc=example,dc=com** 입니다.

- 디렉터리 트리에는 4개의 분기 지점이 있습니다.
  - ou=people
  - ou=groups
  - ou=resources
  - ou=roles
- 모든 ExampleCom 사용자 항목은 ou=people 분기 아래에 생성됩니다.

  people 항목은 모두 사람 ,organizational Person ,inetOrgPerson, examplePerson 개체 클래스의 멤버입니다. uid 속성은 각 항목에 대해 고유 이름(DN)을 고유하게 식별합니다. 예를 들어, 회사는 Babs Jensen (uid=bjensen) 및 Emily Stanton (uid=estanton)에 대한 항목이 포함되어 있습니다.
- ExampleCom의 각 부서에 대해 영업,마케팅 및 회계 역할이 생성됩니다.

  각 person 항목에는 해당 사람이 속한 부서를 식별하는 역할 특성이 포함되어 있습니다. 이제 회사는 이러한 역할을 기반으로 액세스 제어 지침(ACI)을 생성할 수 있습니다.

  역할에 대한 자세한 내용은 다음을 참조하십시오. 4.3.2절. “Directory Server의 역할 정보”
- 다음 그룹 분기는 ou=groups 분기 아래에 생성됩니다.
  - cn=administrators 그룹에는 디렉터리 콘텐츠를 관리하는 디렉터리 관리자에 대한 항목이 포함되어 있습니다.
  - cn=messaging admins 그룹에는 메일 계정만 관리하는 메일 관리자의 항목이 포함되어 있습니다. 이 그룹은 메시징 서버에서 사용하는 관리자 그룹에 해당합니다.

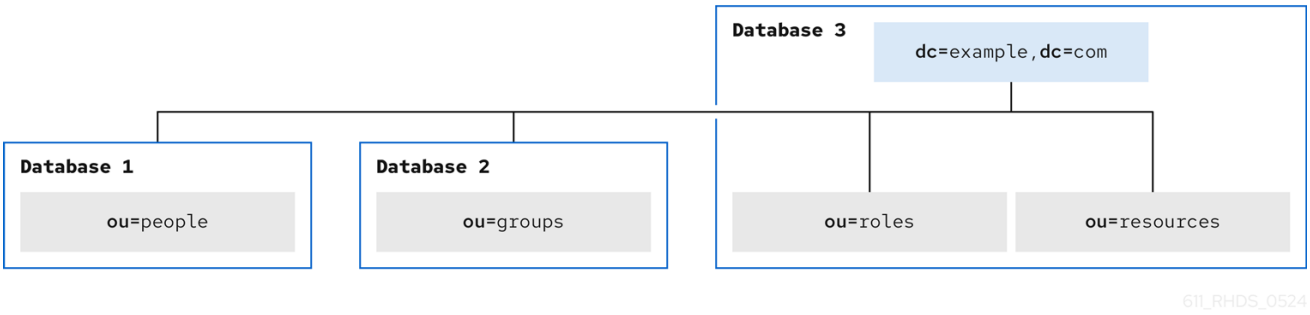
- ou=resources 분기 아래의 다음 분기가 생성됩니다.
  - ou=conference room은 회의룸 을 위한 방입니다.
  - 사무실의 ou= Cryo stats 분기입니다.
- 항목이 관리 그룹에 속하는지 여부에 따라 mailquota 속성 값을 제공하는 서비스 클래스(CoS)가 생성됩니다. 이 CoS는 관리자에게 100GB의 메일 할당량을 제공하는 반면 일반적인 ExampleCom 직원은 5GB의 메일 할당량을 갖습니다.

8.1.4. 로컬 엔터프라이즈의 토폴로지 설계

ExampleCom 배포 팀은 디렉터리 데이터베이스 및 서버 토폴로지를 설계하기 시작합니다.

ExamleCom은 다음과 같은 데이터베이스 토폴로지를 설계합니다.

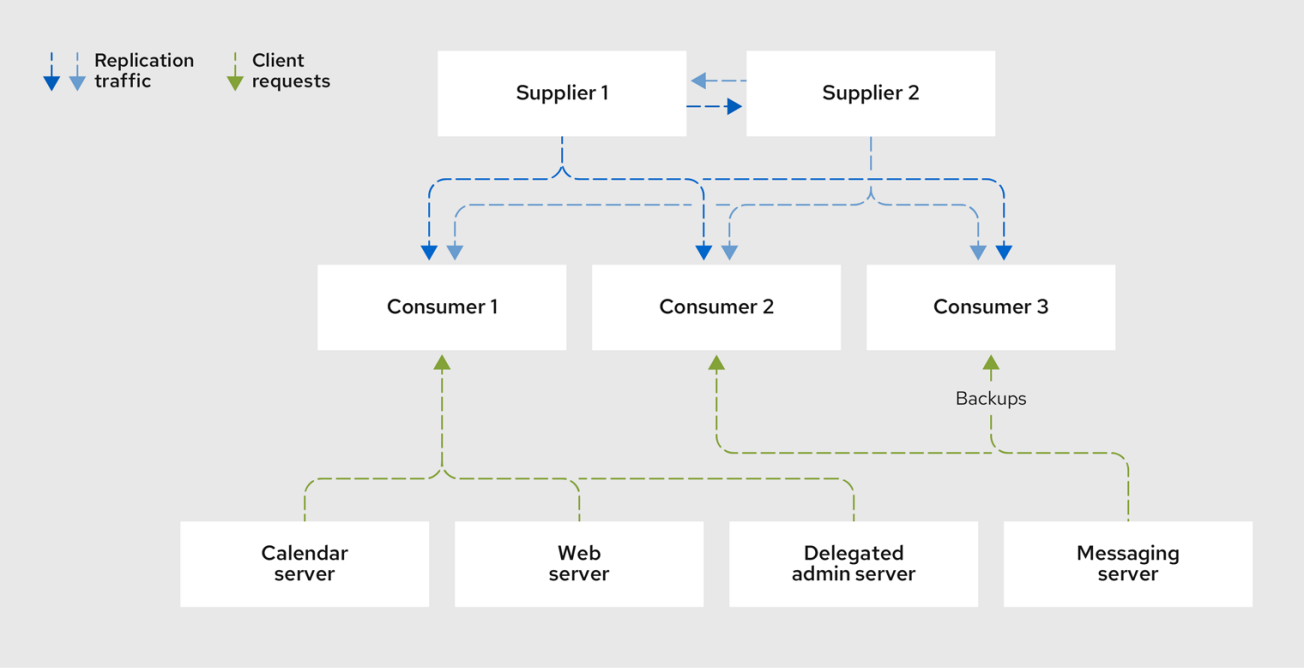
그림 8.2. 로컬 엔터프라이즈 데이터베이스 토폴로지



- 데이터베이스 1 은 ou=people 분기를 저장합니다.
- 데이터베이스 2 는 ou=groups 분기를 저장합니다.
- 데이터베이스 3 은 ou=resources 및 ou=roles 분기와 dc=example,dc=com 루트 접미사를 저장합니다.

ExamleCom은 다음 서버 토폴로지를 설계합니다.

그림 8.3. 로컬 엔터프라이즈 서버 토폴로지



ExampleCom은 두 개의 공급자 서버와 세 개의 소비자 서버가 있는 서버 토폴로지를 사용하기로 결정했습니다. 두 공급자 각각은 **Directory Server** 배포에서 세 명의 소비자를 모두 업데이트합니다.

소비자는 하나의 메시징 서버 및 다른 서버에 데이터를 제공합니다. 호환되는 서버의 요청 수정은 적절한 소비자 서버로 라우팅됩니다. 소비자 서버는 스마트 추천을 사용하여 요청이 수정되는 데이터의 주요 사본에 대한 책임이 있는 공급자 서버로 라우팅합니다.

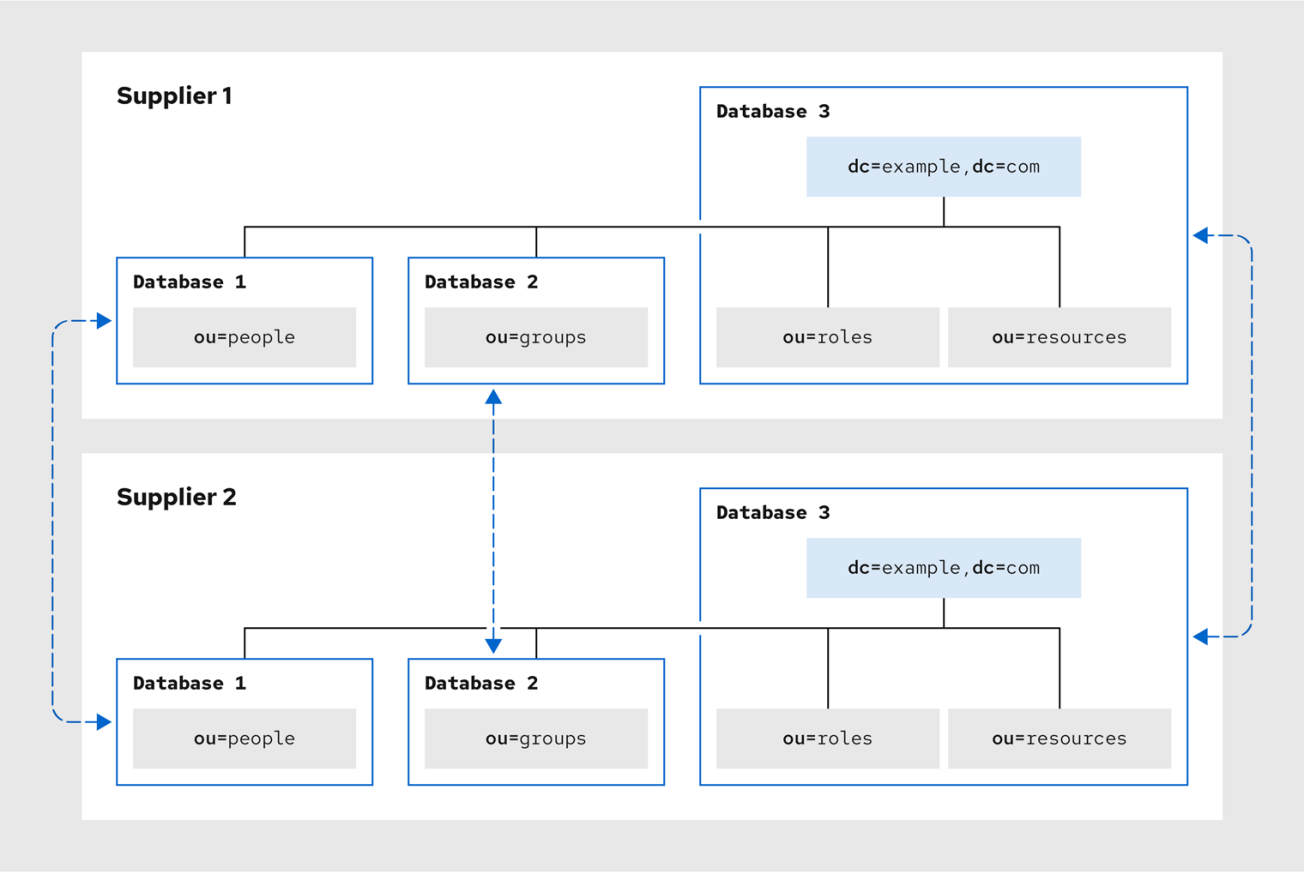
8.1.5. 로컬 엔터프라이즈의 복제 설계

ExampleCom은 디렉터리 데이터의 고가용성을 보장하기 위해 다중 제공 복제 설계를 사용하기로 결정했습니다. 다중 제공 복제에 대한 자세한 내용은 다음을 참조하십시오.

Multi-supplier 아키텍처

ExampleCom은 다중 제공 복제 아키텍처에서 두 개의 공급자 서버를 사용합니다. 공급업체는 디렉터리 데이터가 일관되게 유지되도록 서로를 업데이트합니다. 다음 다이어그램은 **ExampleCom**의 공급업체 제공 아키텍처를 보여줍니다.

그림 8.4. ExampleCom 다중 공급 아키텍처

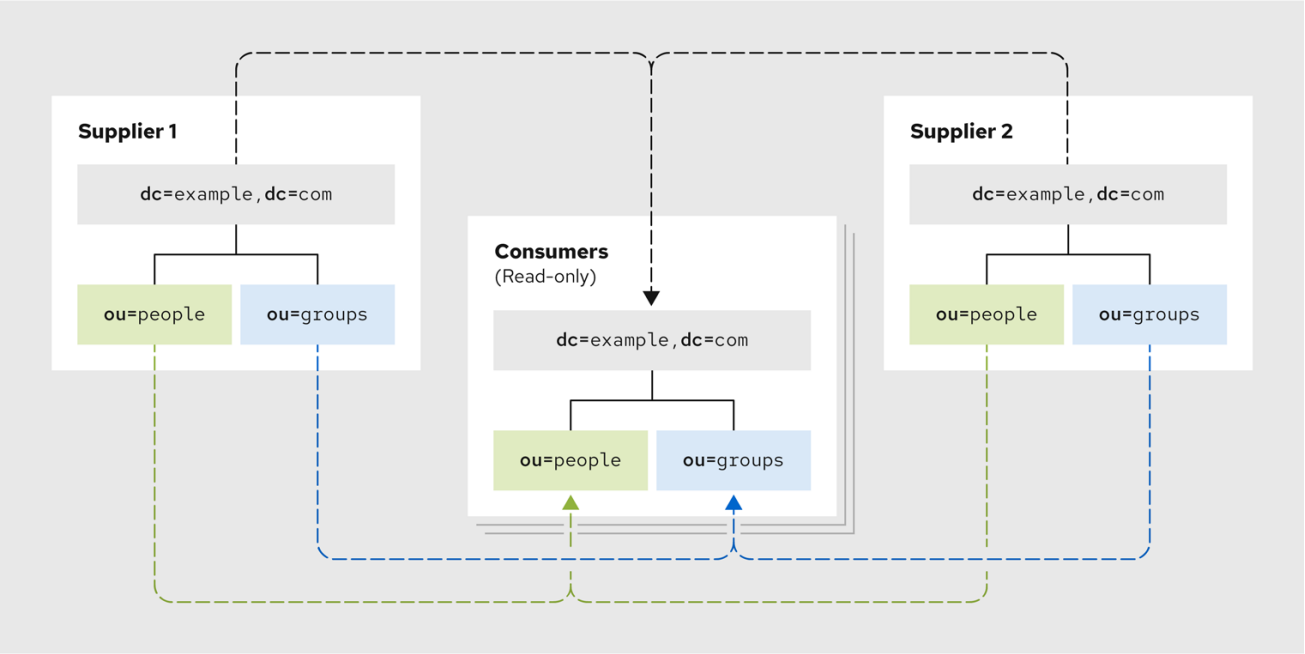


611_RHDS_0524

vendor-consumer 아키텍처

다음 다이어그램에서는 디렉터리의 **ExampleCom** 배포에서 공급업체 서버가 각 소비자에게 복제하는 방법을 설명합니다.

그림 8.5. 예Com 공급자 소비자 아키텍처



두 공급자 서버 모두 세 개의 소비자 서버를 업데이트합니다. 이렇게 하면 공급자 서버 중 하나가 실패하면 소비자가 영향을 받지 않습니다.

8.1.6. 로컬 엔터프라이즈 보안 설계

디렉터리 데이터를 보호하기 위해 ExampleCom은 다음 액세스 제어 명령(ACI)을 생성합니다.

- 직원이 항목을 수정할 수 있는 ACI입니다. 사용자는 uid,manager 및 department 속성을 제외한 모든 속성을 수정할 수 있습니다.
- 직원 및 직원 관리자만 직원 데이터 개인 정보를 보호하기 위해 직원 홈 주소와 전화 번호를 볼 수 있는 ACI입니다.
- 두 관리자 그룹에 적절한 디렉터리 권한을 부여하는 디렉터리 트리의 루트에 있는 ACI입니다.
  - 디렉터리 관리자 그룹에는 디렉터리에 대한 전체 액세스 권한이 필요합니다.
  - 메시징 관리자 그룹에는 mailRecipient 및 mailGroup 개체 클래스와 mail 특성을 포함

하여 이러한 오브젝트 클래스에서 허용하는 속성에 대한 쓰기 및 삭제 권한이 필요합니다. ExampleCom은 또한 메시징 관리자 그룹에 메일 그룹을 생성하기 위해 그룹 하위 디렉터리에 대한 쓰기,삭제 및 추가 권한을 부여합니다.

- 읽기,검색 및 액세스에 대한 익명 액세스를 허용하는 디렉터리 트리의 루트에 있는 일반 ACI입니다. 또한 이 ACI는 익명의 사용자가 암호 정보에 대한 액세스를 거부합니다.

- 모든 급여 정보에 대한 회계 역할 액세스 권한을 부여하는 ACI입니다.

또한 ExampleCom은 다음과 같은 보안 조치를 결정합니다.

- 서비스 거부 공격 및 부적절한 사용으로부터 서버를 보호하기 위해 ExampleCom은 클라이언트가 바인딩하는 데 사용하는 DN에 따라 리소스 제한을 설정합니다.

  - 익명 사용자는 검색 요청에 대한 응답으로 한 번에 100 개의 항목을 수신할 수 있습니다.

  - 메시지 관리자는 1,000개의 항목을 수신할 수 있습니다.

  - 디렉터리 관리자는 무제한의 항목을 수신할 수 있습니다.

- ExampleCom은 암호가 8자 이상이어야 하고 90일 후에 만료되는 암호 정책을 생성합니다.

  암호 정책에 대한 자세한 내용은 암호 정책 설정을 참조하십시오.

8.1.7. 로컬 엔터프라이즈의 운영 결정

회사는 디렉터리의 일상적인 운영에 대해 다음과 같은 결정을 내립니다.

- 매일 매일 데이터베이스를 백업하십시오.

- SNMP를 사용하여 서버 상태를 모니터링합니다.

- 액세스 및 오류 로그를 자동으로 순환합니다.
- 오류 로그를 모니터링하여 서버가 예상대로 수행 중인지 확인합니다.
- 액세스 로그를 모니터링하여 인덱싱할 수 있는 검색을 나타냅니다.

추가 리소스

- 로그 파일 참조.

8.2. 다국적 엔터프라이즈 설계 예

이전에는 로컬 엔터프라이즈 설계 예제 의 소규모 회사인 **ExampleCom**은 미국, 유럽 및 아시아의 세 가지 지역에 분산된 대규모 조직으로 성장했습니다. 이 회사는 현재 **20,000**명 이상의 직원이 있으며, 모든 직원이 **ExampleCom** 사무실이 있는 국가에 거주하고 있습니다.

ExampleCom은 회사 전체 **LDAP** 디렉토리를 시작하여 내부 통신을 개선하여 웹 애플리케이션을 보다 쉽게 개발 및 배포하고 보안 및 개인 정보를 개선하기로 결정했습니다.

국제 회사의 디렉터리 트리를 설계할 때 **ExampleCom**은 다음 질문에 대한 솔루션을 찾아야 합니다.

- 디렉토리 항목을 논리적으로 수집하는 방법은 무엇입니까?
- 데이터 관리를 어떻게 지원할 수 있습니까?
- 글로벌 규모에서 복제를 지원하는 방법은 무엇입니까?

또한 **ExampleCom**은 공급업체 및 거래 파트너가 외부 클라이언트에 회사 인트라넷의 확장으로 이 엑스트라넷을 사용하고 구현할 수 있는 엑스트라넷을 만들고자 합니다.

8.2.1. 다국적 기업의 데이터 설계

ExampleCom International은 설문 조사를 수행하기 위한 배포 팀을 생성합니다. 배포 팀은 사이트 설문 조사에서 다음 주요 포인트를 결정합니다.

- 메시징 서버는 대부분의 **ExampleCom** 사이트에 대한 이메일 라우팅, 전달 및 읽기 서비스를 제공하는 데 사용됩니다. 엔터프라이즈 서버는 문서 게시 서비스를 제공합니다. 모든 서버는 **Red Hat Directory Server 12**에서 실행됩니다.

- **ExampleCom International**은 관리자가 데이터를 로컬에서 관리할 수 있도록 허용해야 합니다. 예를 들어, 유럽 사이트는 디렉터리의 유럽 지점과 이 분기 데이터의 주요 사본을 관리하는 역할을 담당합니다.

- **ExampleCom** 국제 사무실의 지리적 배포로 사용자 및 애플리케이션은 하루 **24**시간 디렉터리에 액세스해야 합니다.

- 특정 데이터 요소의 데이터 값은 여러 언어로 되어 있어야 합니다.

참고

모든 데이터는 **UTF-8** 문자 세트를 사용합니다. 다른 문자 세트는 **LDAP** 표준을 위반합니다.

또한 엑스트라넷의 데이터 설계는 다음 조건이 충족되어야 합니다.

- 일부 공급업체는 회사 계약 관리를 위해 **ExampleCom International** 디렉터리에 로그인해야 합니다. 부분 공급업체는 인증에 사용되는 데이터 요소(예: 이름 및 사용자 암호)에 의존합니다.

- 거래 파트너는 이 디렉토리를 사용하여 이메일 주소 및 전화번호와 같은 파트너 네트워크의 연락처 정보를 조회할 것입니다.

8.2.2. 다국적 기업의 스키마 설계

ExampleCom International은 원래 스키마 설계를 사용하고 엑스트라넷을 지원하기 위해 두 가지 새로운 오브젝트 클래스를 추가합니다.

-

exampleSupplier 오브젝트 클래스는 **exampleSupplierID** 특성을 허용합니다. 이 속성에는 **ExampleCom International**에서 각 기타 부품 공급 업체에 할당하는 고유 **ID**가 포함되어 있습니다.

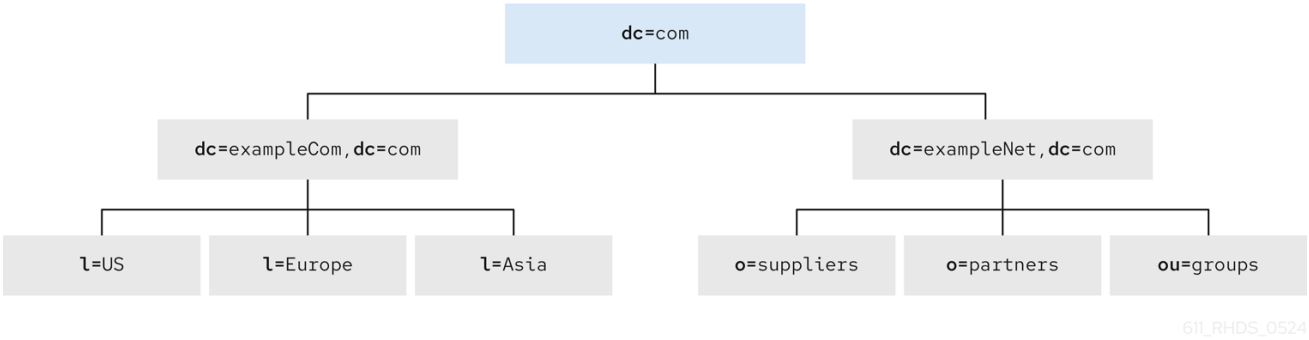
- **examplePartner** 오브젝트 클래스는 **examplePartnerID** 특성을 허용합니다. 이 속성에는 **ExampleCom International**에서 각 거래 파트너에게 할당하는 고유 **ID**가 포함되어 있습니다.

기본 디렉터리 스키마 사용자 지정에 대한 자세한 내용은 스키마 사용자 지정을 참조하십시오.

8.2.3. 다국적 기업의 디렉터리 트리 설계

ExampleCom International은 다음 디렉터리 트리를 생성합니다.

그림 8.6. ExampleCom International의 기본 디렉터리 트리



dc=com 접미사는 디렉터리 트리의 루트입니다. 이 접미사 아래에 회사는 다음 분기를 생성합니다.

- **ExampleCom International**의 내부 데이터가 포함된 **dc=exampleCom,dc=com** 분기입니다.
- **extranet**에 대한 데이터를 포함하는 **dc=exampleNet,dc=com** 분기입니다.

dc=exampleCom,dc=com 에서 인트라넷의 디렉터리 트리에는 세 가지 주요 분기가 있습니다. 각 분기는 **ExampleCom International**에 사무실이 있는 지역 중 하나에 해당합니다. 이러한 분기는 **l** (로컬) 속성을 사용하여 식별됩니다.

dc=exampleNet,dc=com 브랜치 아래에 **ExampleCom International**은 다음과 같은 분기를 생성합니다.

- 회사가 협력하는 공급 업체의 **o=suppliers** 브랜치.

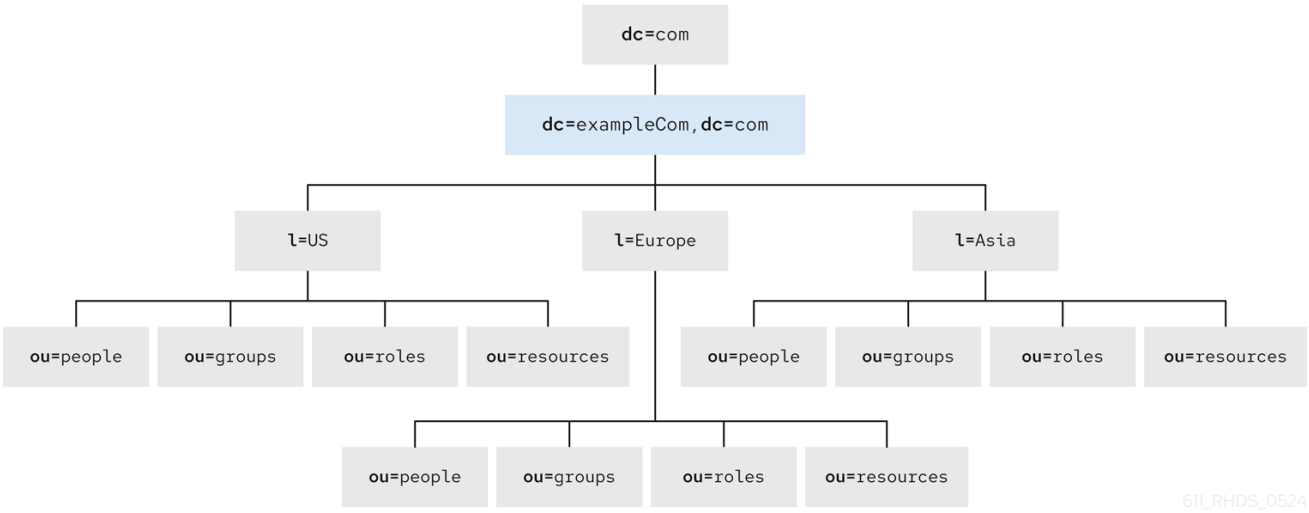
- 거래 파트너를 위한 **o=**파트너 분기입니다.

- **ou=groups** 분기에는 엑스트라넷 관리자 및 메일링 목록에 대한 항목이 포함된 **ou=groups** 분기는 파트너사가 기타 부품 제조에 대한 최신 정보를 구독합니다.

8.2.3.1. ExampleCom International의 인트라넷 설계

dc=exampleCom,dc=com 아래의 각 분기는 로컬 엔터프라이즈 예제의 디렉터리 트리 설계에서 ExampleCom의 원래 디렉터리 트리 설계를 반복합니다.

그림 8.7. 인트라넷의 디렉터리 트리 예



각 지역에 따라 **ExampleCom International**은 다음 분기 지점을 생성합니다.

- **ou=people**

- **ou=groups**

- **ou=roles**

- ou=resources

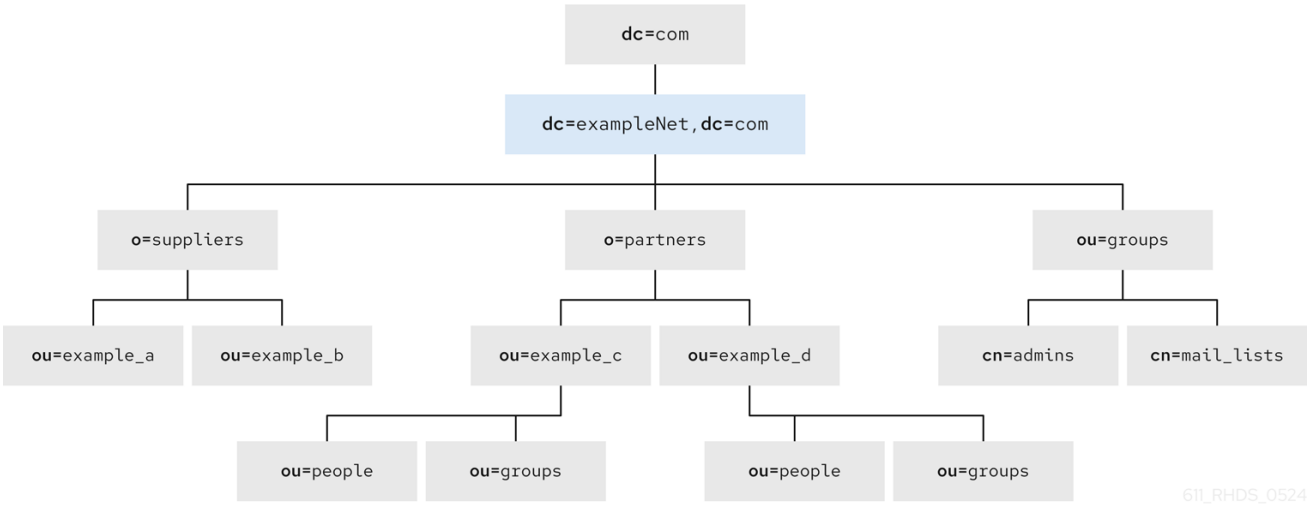
l=Asia locality의 항목은 다음과 같이 LDIF에 나타납니다.

```
dn: l=Asia,dc=exampleCom,dc=com
objectclass: top
objectclass: locality
l: Asia
description: includes all sites in Asia
```

8.2.3.2. ExampleCom International의 엑스트라넷 설계

다음 다이어그램은 ExampleCom extranet의 디렉터리 트리를 보여줍니다.

그림 8.8. 엑스트라넷의 디렉터리 트리 예



8.2.4. 다국적 기업의 토폴로지 설계

ExampleCom International 배포 팀은 디렉터리 데이터베이스 및 서버 토폴로지를 설계하기 시작합니다.

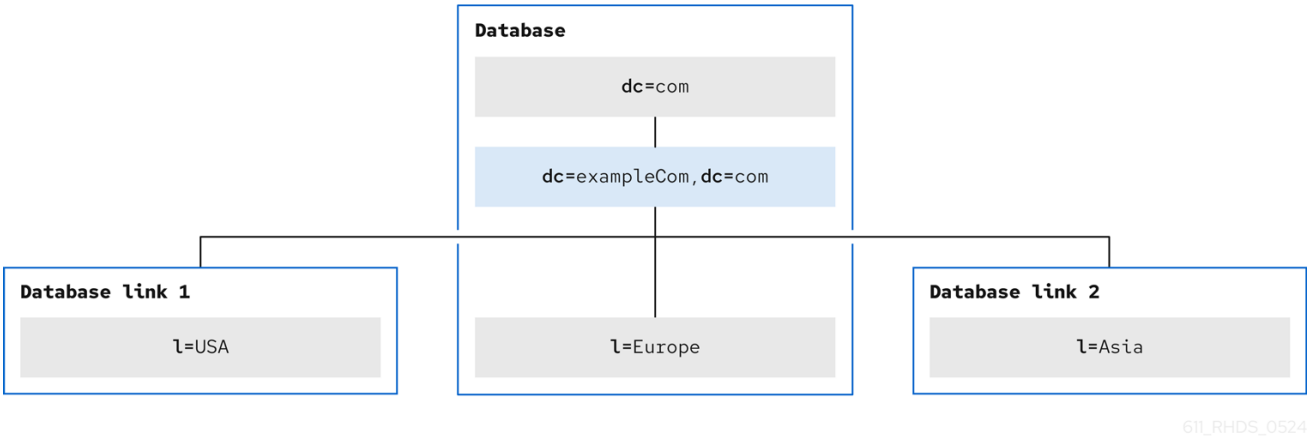
8.2.4.1. ExampleCom International의 데이터베이스 토폴로지

ExampleCom International은 모든 지역에 대해 동일한 토폴로지 설계를 사용합니다. 그러나 유럽 지역성은 다음 분기에 대한 데이터의 주요 사본을 저장합니다.

- dc=com root 항목
- dc=exampleCom,dc=com아래의 인트라넷
- dc=exampleNet,dc=com아래의 extranet

다음 다이어그램에서는 지역별 유럽의 데이터베이스 토폴로지를 보여줍니다.

그림 8.9. ExampleCom 유럽의 데이터베이스 토폴로지



l=Europe 데이터베이스는 dc=exampleCom,dc=com 및 dc=com 항목의 주요 사본을 저장합니다.

데이터베이스 링크 1 과 데이터베이스 링크 2 는 각 국가에 로컬로 저장된 데이터베이스를 가리킵니다. 예를 들어, 작업 요청에서는 l=USA 분기 아래에 있는 데이터를 수신하도록 작업 요청이 미국 내의 서버의 데이터베이스에 대한 데이터베이스 링크에 의해 연결됩니다. 데이터베이스 링크 및 체인에 대한 자세한 내용은 체인 사용을 참조하십시오.

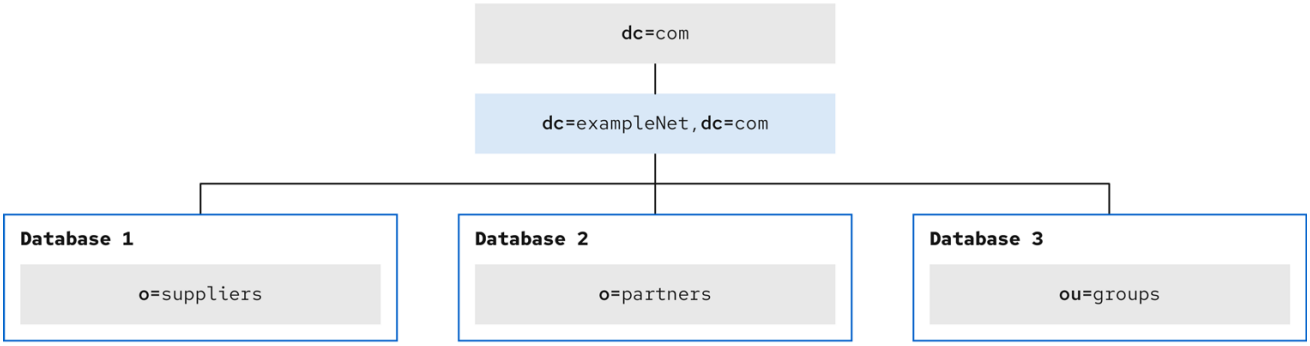
유럽 서버에는 엑스트라넷에 대한 데이터의 주요 사본이 포함되어 있습니다. 엑스트라넷 데이터는 다음과 같은 세 가지 데이터베이스에 저장됩니다.

- 데이터베이스 1 은 o=suppliers 분기의 주요 사본을 저장합니다.
- 데이터베이스 2 는 o=partners 분기의 주요 사본을 저장합니다.

- 데이터베이스 3 은 ou=groups 분기의 주요 사본을 저장합니다.

다음 다이어그램은 extranet의 데이터베이스 토폴로지를 보여줍니다.

그림 8.10. ExampleCom International Extranet의 데이터베이스 토폴로지



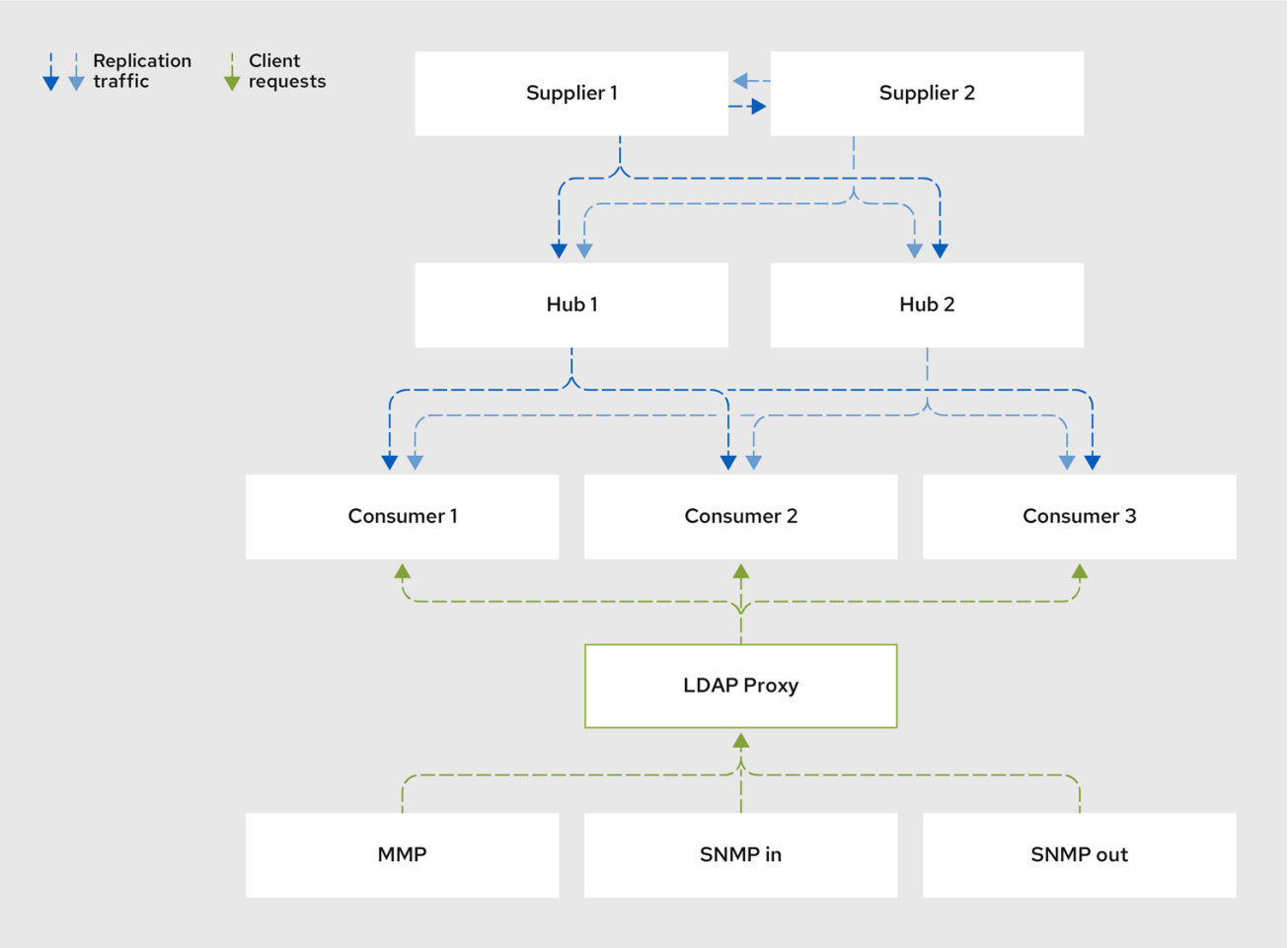
8.2.4.2. ExampleCom International의 서버 토폴로지

ExampleCom International은 다음과 같은 유형의 서버 토폴로지를 개발합니다.

- 기업 인트라넷의 토폴로지입니다. ExampleCom은 각 주요 지역에 대해 하나씩 세 개의 데이터 센터 즉 유럽, 미국 및 아시아를 사용하기로 결정했습니다. 각 데이터 센터에는 다음 서버가 포함되어 있습니다.
  - 두 개의 공급자 서버.
  - 두 개의 허브 서버.
  - 소비자 서버 세 개.
- 파트너 엑스트라넷의 토폴로지입니다.

다음 다이어그램에서는 ExampleCom 유럽 데이터 센터의 아키텍처를 보여줍니다.

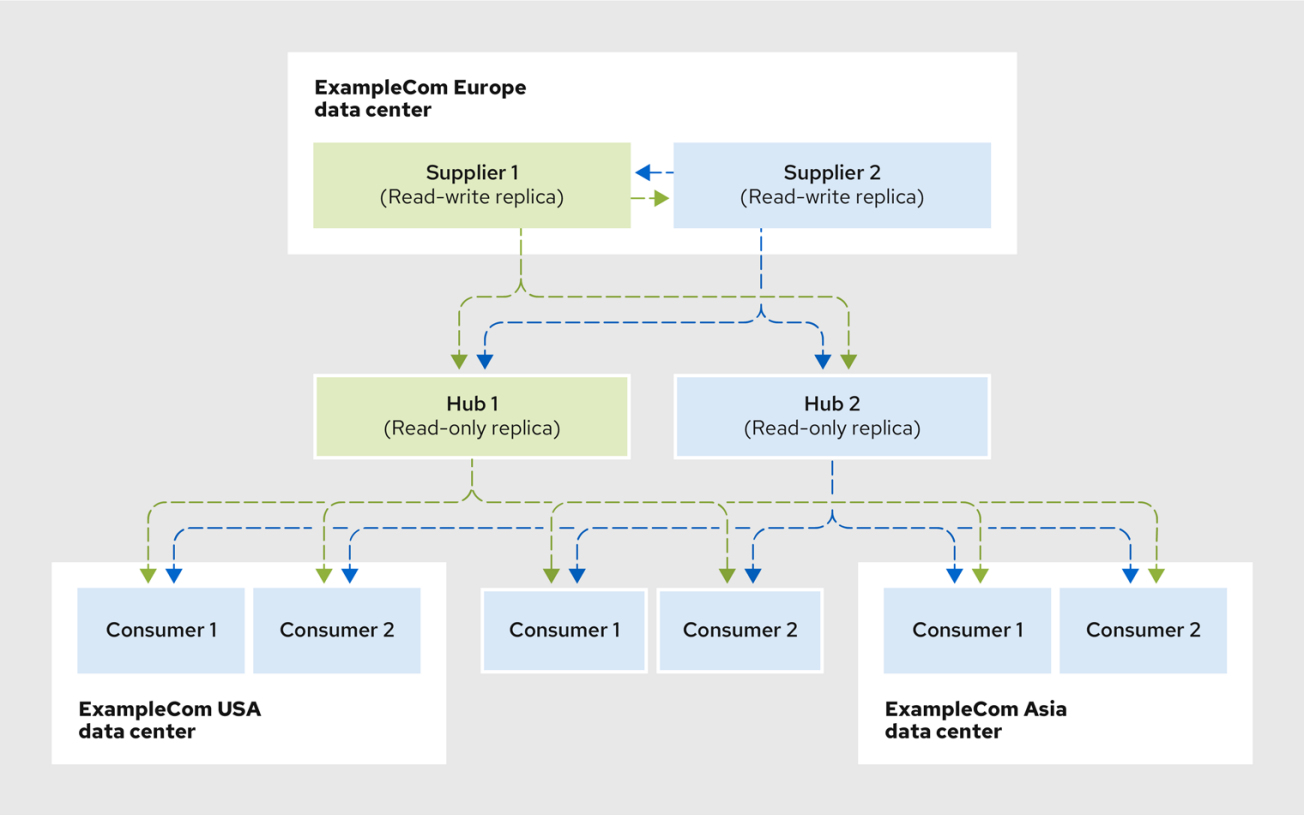
그림 8.11. ExampleCom Europe의 서버 토폴로지



유럽 데이터 센터에는 **ExampleCom** 엑스트라넷의 주요 사본이 포함되어 있습니다. 이 데이터는 미국 데이터 센터의 두 개의 소비자 서버와 아시아 데이터 센터의 소비자 서버로 복제됩니다. 전반적으로 **ExampleCom**은 엑스트라넷을 지원하기 위해 **10** 대의 서버가 필요합니다.

다음 다이어그램은 유럽 데이터 센터에 있는 **ExampleCom** 엑스트라넷의 서버 아키텍처를 보여줍니다.

그림 8.12. ExampleCom International extranet의 서버 토폴로지



허브 서버는 데이터를 각 데이터 센터의 두 소비자 서버(유럽, 미국 및 아시아)에 복제합니다.

8.2.5. 다국적 기업의 복제 설계

ExampleCom International에서는 디렉터리에 대한 복제를 설계할 때 다음 사항을 고려합니다.

- 데이터는 로컬로 관리됩니다.
- 네트워크 연결의 품질은 사이트마다 다릅니다.
- 데이터베이스 링크는 원격 서버의 데이터를 연결하는 데 사용됩니다.
- 데이터의 읽기 전용 복사본이 포함된 Hub 서버는 데이터를 소비자 서버에 복제하는 데 사용됩니다.

허브 서버는 메일 서버 또는 웹 서버와 같은 중요한 디렉터리 지원 애플리케이션 근처에 있습니다.

공급자 서버가 쓰기 작업에 집중하도록 하려면 허브 서버만 복제를 수행합니다.

나중에 **ExampleCom**이 확장되고 더 많은 소비자 서버를 추가해야 하는 경우 추가 소비자는 공급 업체 서버의 성능에 영향을 미치지 않습니다.

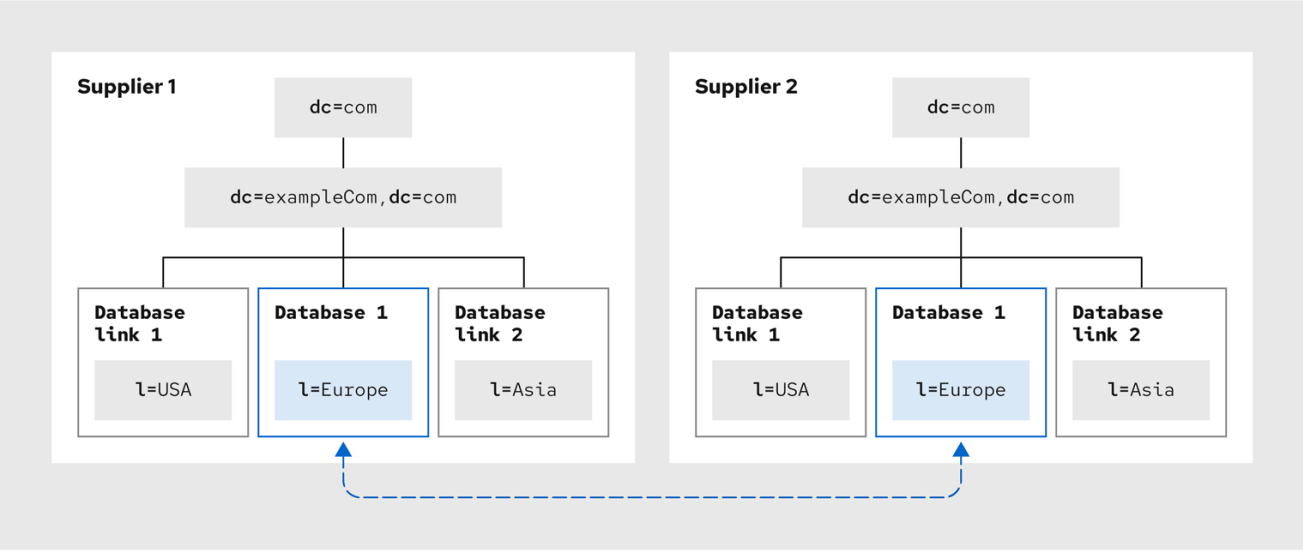
Multi-supplier 아키텍처

ExampleCom 인트라넷의 경우 각 지역성은 데이터의 주요 사본을 저장하고 데이터베이스 링크를 사용하여 다른 지역의 데이터에 체인합니다.

데이터의 주요 사본에 대해 각 지역성은 다중 제공 복제 아키텍처를 사용합니다.

다음 다이어그램은 **dc=exampleCom,dc=com** 및 **dc=com** 분기를 포함하는 지역 유럽의 다중 공급 업체 아키텍처를 보여줍니다.

그림 **8.13. ExampleCom** 유럽의 다중 공급 아키텍처



611_RHDS_0524

각 지역에는 해당 사이트에 대한 데이터의 주요 사본을 공유하는 두 개의 공급자가 포함되어 있습니다. 각 지역성은 데이터의 주요 복사본에 대한 책임이 있습니다.

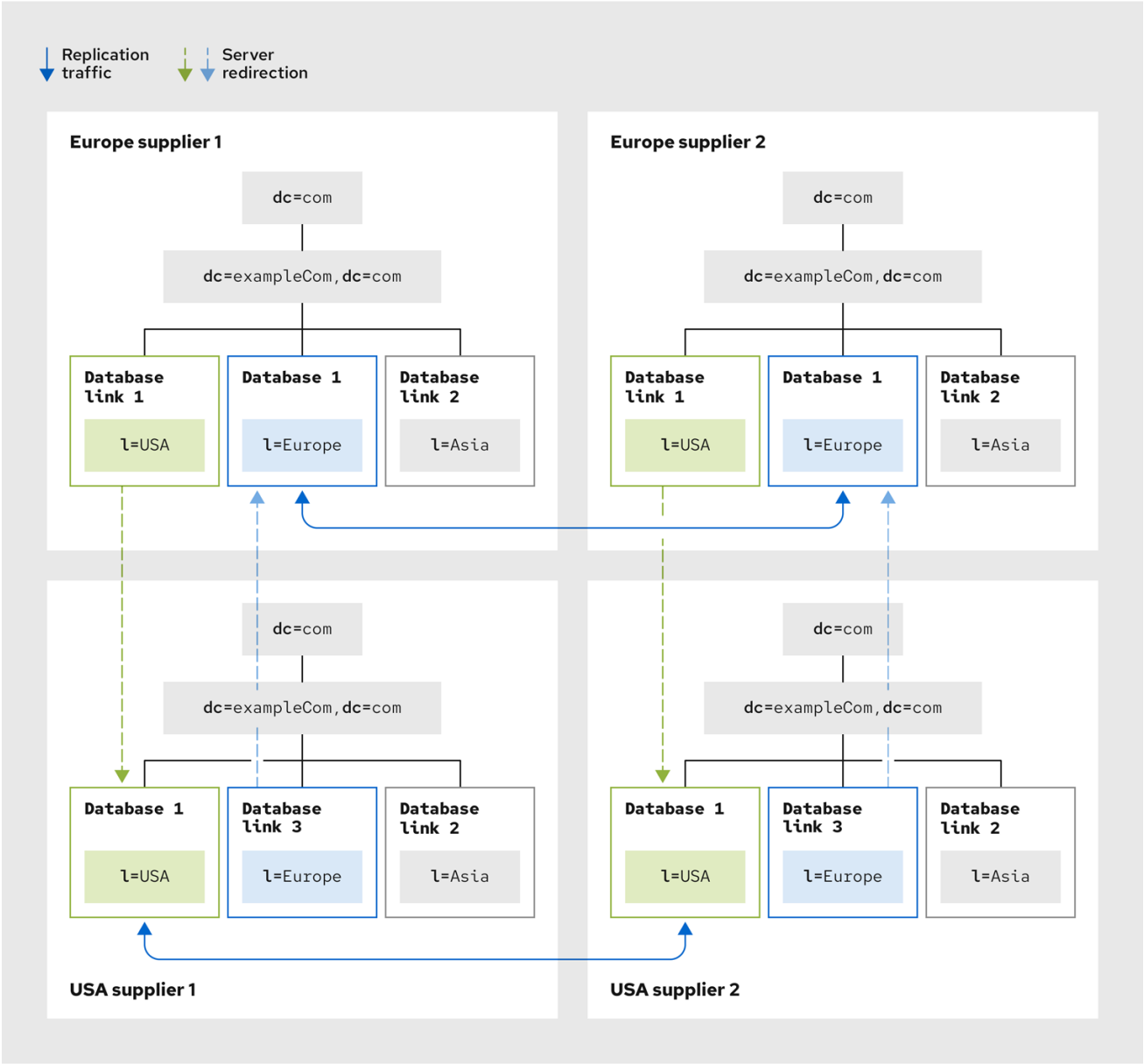
다중 공급 아키텍처를 사용하면 데이터 가용성을 보장하고 각 공급업체 서버에서 관리하는 워크로드

의 균형을 유지하는 데 도움이 됩니다.

전체 실패 위험을 줄이기 위해 **ExampleCom**은 각 사이트에서 여러 읽기-쓰기 공급자 디렉터리 서버를 사용합니다.

다음 다이어그램은 유럽의 두 공급자 서버와 미국에서 두 공급자 서버 간의 상호 작용을 보여줍니다.

그림 8.14. ExampleCom 유럽 및 ExampleCom USA의 멀티 공급 업체 아키텍처



ExampleCom USA와 **ExampleCom Asia**와 **ExampleCom Europe**과 **ExampleCom Asia**와 동일한 관계가 있습니다.

8.2.6. 다국적 기업의 보안 설계

ExampleCom International은 새로운 다국적 인트라넷을 지원하기 위해 다음과 같은 액세스 제어를 추가하는 이전 보안 설계를 사용합니다.

- **ExampleCom**은 각 국가에서보다 제한적인 **ACI**를 생성하는 인트라넷의 루트와 각 국가의 지부에 일반 **ACI**를 추가합니다.

- **ExampleCom**은 디렉터리의 **ACI** 수를 최소화하기 위해 매크로 **ACI**를 사용하기로 결정합니다.

  ExampleCom은 매크로를 사용하여 **ACI**의 대상 또는 바인딩 규칙 부분에서 **DN**을 나타냅니다. 디렉터리가 들어오는 **LDAP** 작업을 가져오면 **ACI** 매크로가 **LDAP** 작업이 대상으로 하는 리소스와 일치합니다. 일치 항목이 발생하면 **Directory Server**에서 매크로를 대상 리소스의 **DN** 값으로 대체합니다.

매크로 **ACI**에 대한 자세한 내용은 매크로 액세스 제어 지침 사용을 참조하십시오.

ExampleCom은 다음 액세스 제어를 추가하여 엑스트라넷을 지원합니다.

- **ExampleCom**은 모든 추가 작업에 대해 인증서 기반 인증을 사용하기로 결정했습니다. 엑스트라넷에 로그인할 때 사용자는 디지털 인증서가 필요합니다. 디렉터리는 인증서를 저장합니다. 따라서 사용자는 디렉터리에 저장된 공개 키를 검색하여 암호화된 이메일을 보낼 수 있습니다.

- **ExampleCom**은 엑스트라넷에 대한 익명 액세스를 금지하는 **ACI**를 생성합니다. 이렇게 하면 서비스 거부 공격으로부터 엑스트라넷이 보호됩니다.

- **ExampleCom**은 예제**Com** 호스팅 애플리케이션에서만 디렉터리 데이터를 업데이트하려고 합니다. 즉, 엑스트라를 사용하는 파트너 및 공급 업체는 **ExampleCom**에서 제공하는 툴만 사용할 수 있습니다. 엑스트라넷 사용자를 **ExampleCom** 선호 툴로 제한함으로써 **ExampleCom** 관리자는 감사 로그를 사용하여 디렉터리 사용을 추적하고 **ExampleCom International** 이외의 엑스트라넷 사용자가 도입할 수 있는 문제 유형을 제한할 수 있습니다.

9장. DIRECTORY SERVER RFC 지원

지원되는 주요 LDAP 관련 RFC 목록을 찾습니다. Directory Server에서 지원하는 RFC의 전체 목록은 아닙니다.

9.1. LDAPV3 기능

기술 사양 로드맵 (RFC 4510)

이는 추적 문서이며 요구 사항이 포함되어 있지 않습니다.

프로토콜 (RFC 4511)

다음과 같은 예외로 지원됩니다.

- RFC 4511 섹션 4.4.1 Disconnection: Directory Server가 이 경우 연결을 종료합니다.
- RFC 4511 섹션 4.5.1.3. SearchRequest.derefAliases: LDAP 별칭은 지원되지 않습니다.
- RFC 4511 섹션 4.13. IntermediateResponse 메시지

디렉터리 정보 모델 (RFC 4512)

다음과 같은 예외로 지원됩니다.

- RFC 4512 섹션 2.4.2. 구조 개체 클래스: Directory Server는 여러 구조 개체 클래스가 있는 항목을 지원합니다.
- RFC 4512 섹션 2.6. 별칭 항목
- RFC 4512 섹션 4.1.2. 특성 유형: 속성 유형 COLLECTIVE는 지원되지 않습니다.
- RFC 4512 섹션 4.1.4. 일치 규칙 사용

- RFC 4512 섹션 4.1.6. DIT 콘텐츠 규칙
- RFC 4512 섹션 4.1.7. DIT Cryostat 규칙 및 이름
- RFC 4512 섹션 5.1.1. altServer

RFC 4512는 LDAP 서버가 이전에 나열된 예외를 지원하지 않도록 합니다. 자세한 내용은 RFC 4512 섹션 7.1을 참조하십시오. 서버 지침.

인증 방법 및 보안 메커니즘 (RFC 4513)

지원됨.

문자열 Distinguished Names (RFC 4514)

지원됨.

문자열 검색 필터 표시 (RFC 4515)

지원됨.

균일 리소스 로케이터 (RFC 4516)

지원됨. 그러나 이 RFC는 주로 LDAP 클라이언트에 중점을 두고 있습니다.

구문 및 일치 규칙(RFC 4517)

지원됨. 예외:

- directoryStringFirstComponentMatch
- integerFirstComponentMatch
- objectIdentifierFirstComponentMatch
- objectIdentifierFirstComponentMatch

- **keywordMatch**
- **wordMatch**

국제화된 문자열 준비 (RFC 4518)

지원됨.

사용자 애플리케이션용 스키마 (RFC 4519)

지원됨.

entryUUID 운영 속성 (RFC 4530)

지원됨.

콘텐츠 동기화 작업 (RFC 4533)

지원됨.

9.2. 인증 방법

anonymous SASL Mechanism (RFC 4505)

지원되지 않습니다. RFC 4512 에는 **ANONYMOUS SASL** 메커니즘이 필요하지 않습니다. 그러나 **Directory Server**는 **LDAP** 익명 바인딩을 지원합니다.

외부 **SASL** 메커니즘 (RFC 4422)

지원됨.

일반 **SASL** 메커니즘 (RFC 4616)

지원되지 않습니다. RFC 4512 에는 **PLAIN SASL** 메커니즘이 필요하지 않습니다. 그러나 **Directory Server**는 **LDAP** 익명 바인딩을 지원합니다.

SecurID SASL Mechanism (RFC 2808)

지원되지 않습니다. 그러나 **Cyrus SASL** 플러그인이 있는 경우 **Directory Server**에서 이 플러그인을 사용할 수 있습니다.

Kerberos V5(GSSAPI) SASL Mechanism (RFC 4752)

지원됨.

CRAM-MD5 SASL Mechanism (RFC 2195)

지원됨.

digest-MD5 SASL Mechanism (RFC 2831)

지원됨.

일회성 암호 SASL 메커니즘(RFC 2444)

지원되지 않습니다. 그러나 **Cyrus SASL** 플러그인이 있는 경우 **Directory Server**에서 이 플러그인을 사용할 수 있습니다.

9.3. X.509 인증서 스키마 및 속성 지원

X.509 인증서에 대한 LDAP 스키마 정의(RFC 4523)

- 특성 유형 및 오브젝트 클래스: 지원.
- 구문: 지원되지 않습니다. **Directory Server**는 바이너리 및 옥텟 구문을 사용합니다.
- 일치 규칙: 지원되지 않습니다.