



Red Hat Enterprise Linux 7

Linux ドメイン ID、認証、およびポリシーガイド

Linux 環境での Red Hat Identity Management の使用

Red Hat Enterprise Linux 7 Linux ドメイン ID、認証、およびポリシーガイド

Linux 環境での Red Hat Identity Management の使用

Aneta Šteflová Petrová
Red Hat Customer Content Services
aneta@redhat.com

Filip Hanzelka
Red Hat Customer Content Services
fhanzelk@redhat.com

Lucie Maňásková
Red Hat Customer Content Services
lmanasko@redhat.com

Marc Muehlfeld
Red Hat Customer Content Services

Tomáš Čapek
Red Hat Customer Content Services

Ella Deon Ballard
Red Hat Customer Content Services

法律上の通知

Copyright © 2017 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

キーワード

1. FreeIPA. 2. ID 管理. 3. IdM. 4. IPA.

概要

ユーザーとマシンの両方にとって、ID とポリシーの管理はほとんどの企業環境における中核的な機能です。Identity Management は、ID ドメインを作成する方法を提供し、このドメインにより、マシンはドメインへの登録と、シングルサインオンおよび認証サービスに必要な ID 情報に即座にアクセスすることができるようになります。また、承認およびアクセスを管理するポリシー設定も可能になります。Red Hat Enterprise Linux Identity Management に関する他の機能およびサービスについての資料は、本ガイドのほかに以下のガイドがあります。システムレベルの認証ガイド は、ローカルシステム上で認証設定に使用可能な異なるアプリケーションやサービスについて説明しています。これには、authconfig ユーティリティーや System Security Services Daemon (SSSD) サービス、プラグ可能な認証モジュール (PAM) フレームワーク、Kerberos、certmonger ユーティリティー、アプリケーション用のシングルサインオン (SSO) などがあります。Windows 統合ガイドでは、Identity Management を使って Linux ドメインと

Microsoft Windows Active Directory (AD) を統合する方法について説明しています。また、直接および間接的 AD 統合の側面、SSSD を使って Common Internet File System (CIFS) にアクセスする方法、realmd システムなどについて説明しています。

目次

パート I. RED HAT IDENTITY MANAGEMENT の概要	8
第1章 RED HAT IDENTITY MANAGEMENT について	9
1.1. RED HAT IDENTITY MANAGEMENT のゴール	9
1.2. IDENTITY MANAGEMENT ドメイン	11
パート II. IDENTITY MANAGEMENT のインストール	16
第2章 IDENTITY MANAGEMENT サーバーのインストールとアンインストール	17
2.1. サーバーインストールの前提条件	17
2.2. IDM サーバーのインストールに必要なパッケージ	24
2.3. IDM サーバーのインストール: はじめに	25
2.4. IDM サーバーのアンインストール	38
2.5. サーバーの名前変更	38
第3章 IDENTITY MANAGEMENT クライアントのインストールおよびアンインストール	40
3.1. クライアントインストールの前提条件	40
3.2. クライアントのインストールに必要なパッケージ	41
3.3. クライアントのインストール	41
3.4. キックスタートを使用した IDM クライアントの設定	45
3.5. クライアントのインストール後の検討事項	46
3.6. 新規クライアントのテスト	47
3.7. クライアントのアンインストール	47
3.8. クライアントの IDM ドメインへの再登録	47
3.9. クライアントマシンの名前変更	49
第4章 IDENTITY MANAGEMENT のレプリカのインストールとアンインストール	51
4.1. IDM レプリカの説明	51
4.2. レプリカに関するデプロイメントの考慮事項	51
4.3. レプリカのインストールの前提条件	55
4.4. レプリカのインストールに必要なパッケージ	55
4.5. レプリカの作成: 概要	55
4.6. 新規レプリカのテスト	62
4.7. レプリカのアンインストール	62
パート III. サーバーの管理	63
第5章 IDM サーバーおよびサービスの基本的な管理	64
5.1. IDM サーバーの起動と停止	64
5.2. KERBEROS を使用した IDM へのログイン	64
5.3. IDM コマンドラインユーティリティ	66
5.4. IDM WEB UI	70
第6章 レプリケーショントポロジーの管理	76
6.1. レプリカ合意、トポロジーサフィックス、およびトポロジーセグメント	76
6.2. WEB UI: トポロジーグラフを使用したレプリカトポロジーの管理	78
6.3. コマンドライン: IPA TOPOLOGY* コマンドを使用したトポロジーの管理	83
6.4. トポロジーからサーバーを削除する	85
6.5. サーバーロールの管理	87
第7章 ドメインレベルの表示と引き上げ	92
7.1. 現行ドメインレベルの表示	92
7.2. ドメインレベルの引き上げ	93

第8章 IDENTITY MANAGEMENT の更新と移行	94
8.1. IDENTITY MANAGEMENT の更新	94
8.2. RED HAT ENTERPRISE LINUX 6 からバージョン 7 への IDENTITY MANAGEMENT の移行	95
第9章 IDENTITY MANAGEMENT のバックアップと復元	103
9.1. 完全なサーバーバックアップおよびデータのためのバックアップ	103
9.2. バックアップの復元	107
第10章 IDM ユーザーのアクセス制御の定義	110
10.1. IDM エントリーのアクセス制御	110
10.2. セルフサービス設定の定義	111
10.3. ユーザーへのパーミッションの委任	115
10.4. ロールベースのアクセス制御の定義	117
パート IV. アイデンティティの管理	134
第11章 ユーザーアカウントの管理	135
11.1. ユーザーホームディレクトリーの設定	135
11.2. ユーザーのライフサイクル	136
11.3. ユーザーの編集	146
11.4. ユーザーアカウントの有効化、無効化	148
11.5. 管理者以外のユーザーによるユーザーエントリーの管理許可	149
11.6. ユーザーおよびグループへの外部プロビジョニングシステムの使用	152
第12章 ホストの管理	160
12.1. ホスト、サービス、およびマシン ID と認証	160
12.2. ホストエントリー設定のプロパティ	161
12.3. ホストエントリーの追加	162
12.4. ホストエントリーの無効化および再有効化	165
12.5. ホストの公開 SSH キーの管理	165
12.6. ホストの ETHERS 情報の設定	171
第13章 ユーザーおよびホストグループの管理	173
13.1. ユーザーおよびホストグループの IDM での機能	173
13.2. ユーザーまたはホストグループの追加および削除	176
13.3. ユーザーまたはホストグループメンバーの追加および削除	178
13.4. ユーザープライベートグループの無効化	180
13.5. ユーザーおよびユーザーグループの検索属性の設定	182
13.6. ユーザーおよびホストの自動グループメンバーシップの定義	182
第14章 一意の UID および GID 番号の割り当て	190
14.1. ID の範囲	190
14.2. インストール中の ID 範囲の割り当て	190
14.3. 現在割り当てられている ID 範囲の表示	191
14.4. レプリカ削除後の ID 範囲の自動拡張	191
14.5. 手動での ID 範囲の拡張および新規 ID 範囲の割り当て	191
14.6. ID の値が一意であることを確認する	193
14.7. 変更された UID および GID 番号の修復	193
第15章 ユーザーおよびグループスキーマ	194
15.1. デフォルトのユーザーおよびグループスキーマの変更	196
15.2. カスタムのオブジェクトクラスを新規ユーザーエントリーに適用する	196
15.3. カスタムのオブジェクトクラスを新規グループエントリーに適用する	199
15.4. デフォルトのユーザーおよびグループ属性の指定	200
第16章 サービスの管理	205

16.1. サービスエントリーおよび KEYTAB の追加と編集	205
16.2. クラスタサービスの設定	207
16.3. 複数サービスでの同一サービスプリンシパルの使用	208
16.4. 複数のサーバー向けの既存の KEYTAB 取得	208
16.5. サービスエントリーの無効化および再有効化	210
第17章 ユーザーアクセスのホストおよびサービスへの委任	211
17.1. サービス管理の委任	211
17.2. ホスト管理の委任	212
17.3. WEB UI を使ったホストまたはサービス管理の委任	212
17.4. 委任サービスへのアクセス	213
第18章 ID ビュー	215
SSSD パフォーマンスへのマイナス影響の可能性	215
その他のリソース	215
18.1. ID ビューで上書き可能な属性	215
18.2. ID ビューコマンドのヘルプ	216
18.3. ホストごとにユーザーアカウントで異なる属性値を定義する	216
第19章 IDM ユーザーのアクセス制御の定義	222
第20章 KERBEROS フラグとプリンシパルエイリアスの管理	223
20.1. サービスおよびホスト向けの KERBEROS フラグ	223
20.2. ユーザー、ホスト、およびサービス向け KERBEROS プリンシパルエイリアスの管理	225
第21章 NIS ドメインおよびネットグループとの統合	228
21.1. NIS と IDENTITY MANAGEMENT	228
21.2. IDENTITY MANAGEMENT で NIS を有効にする	230
21.3. NETGROUPS の作成	230
21.4. 自動マウントマップの NIS クライアントへの公開	234
21.5. NIS から IDM への移行	235
パート V. 認証メカニズムの管理	242
第22章 ユーザー認証	243
22.1. ユーザーパスワード	243
22.2. ワンタイムパスワード	247
22.3. ユーザーの認証情報をもとにサービスやホストへのアクセス制限	254
22.4. ユーザーの公開 SSH キーの管理	256
22.5. SSSD が OPENSSH サービス用にキャッシュを提供するように設定する方法	260
22.6. IDENTITY MANAGEMENT でのスマートカード認証	262
22.7. ユーザー証明書	262
第23章 IDENTITY MANAGEMENT でのスマートカード認証	263
23.1. IDENTITY MANAGEMENT サーバーのスマートカードリンクの管理	263
23.2. スマートカードで IDENTITY MANAGEMENT クライアントへ認証する方法	273
23.3. スマートカードを使用してリモートで IDENTITY MANAGEMENT システムに対する認証を行う方法	276
23.4. スマートカード認証のユーザー名ヒントのポリシー設定	278
23.5. IDENTITY MANAGEMENT での PKINIT スマートカード認証	280
23.6. スマートカードを使用して IDENTITY MANAGEMENT WEB UI への認証を行う方法	282
23.7. WEB アプリケーションと IDENTITY MANAGEMENT のスマートカード認証の統合	286
第24章 ユーザー、ホスト、およびサービス向け証明書の管理	289
24.1. 統合 IDM CA での証明書の管理	289
24.2. 外部 CA 発行の証明書の管理	293
24.3. 証明書の一覧と表示	295

24.4. 証明書のプロファイル	297
24.5. 証明局の ACL ルール	302
24.6. IDM CA での証明書プロファイルおよび ACL を使用したユーザー証明書の発行	308
第25章 VAULT を使用した認証情報の秘密の保存	315
25.1. VAULT の仕組み	315
25.2. VAULT 使用における前提条件	317
25.3. VAULT コマンドのヘルプ	317
25.4. ユーザー個人の秘密の保存	318
25.5. VAULT でのサービスの秘密の保存	319
25.6. 複数ユーザー用の共通の秘密の保存	323
第26章 証明書と認証局の管理	326
26.1. 軽量のサブ証明局 (CA)	326
26.2. 証明書の更新	328
26.3. CA 証明書の手動インストール	330
26.4. 証明書チェーンの変更	331
26.5. IDM を有効期限の切れた証明書で起動できるようにする方法	331
26.6. HTTP または LDAP 用のサードパーティー証明書のインストール	332
26.7. OCSP 応答の設定	333
26.8. CA の既存の IDM ドメインへのインストール	334
26.9. WEB サーバーおよび LDAP サーバーの証明書の置き換え	335
パート VI. ポリシーの管理	336
第27章 パスワードポリシーの定義	337
27.1. パスワードポリシーのその役割	337
27.2. IDM におけるパスワードポリシー	337
27.3. 新規パスワードポリシーの追加	340
27.4. パスワードポリシー属性の編集	341
27.5. パスワードの有効期限を変更して即座に反映させる	342
第28章 KERBEROS ドメインの管理	343
28.1. KERBEROS チケットポリシーの管理	343
28.2. KERBEROS プリンシパルの鍵の変更	345
28.3. KEYTAB の保護	347
28.4. KEYTAB の削除	347
28.5. その他のリソース	348
第29章 SUDO の使用	349
29.1. IDENTITY MANAGEMENT の SUDO ユーティリティー	349
29.2. IDENTITY MANAGEMENT での SUDO ルール	349
29.3. SUDO ポリシーをルックアップする場所の設定	350
29.4. SUDO コマンド、コマンドグループ、およびルールの追加	352
29.5. SUDO コマンドとコマンドグループの編集	356
29.6. SUDO ルールの修正	356
29.7. SUDO コマンド、コマンドグループ、およびルールの表示	367
29.8. SUDO ルールの有効化および無効化	367
29.9. SUDO コマンド、コマンドグループ、およびルールの削除	368
第30章 ホストベースのアクセス制御の設定	370
30.1. IDM での HOST-BASED ACCESS CONTROL の機能	370
30.2. IDM ドメインでの HOST-BASED ACCESS CONTROL の設定	370
30.3. カスタムの HBAC サービス用に HBAC サービスエントリーの追加	380
30.4. HBAC サービスグループの追加	381

第31章 SELINUX ユーザーマップの定義	383
31.1. IDENTITY MANAGEMENT、SELINUX、およびユーザーのマッピング	383
31.2. SELINUX ユーザーマップの順序とデフォルト値の設定	385
31.3. SELINUX ユーザーの IDM ユーザーへのマッピング	388
パート VII. ネットワークサービスの管理	394
第32章 DNS の管理	395
32.1. IDENTITY MANAGEMENT における BIND	395
32.2. サポートされる DNS ゾーンタイプ	396
32.3. DNS 設定の優先順位	396
32.4. MASTER DNS ゾーンの管理	397
32.5. 動的 DNS 更新の管理	412
32.6. DNS 転送の管理	419
32.7. 逆引き DNS ゾーンの管理	425
32.8. DNS クエリーポリシーの定義	428
32.9. DNS の場所	428
32.10. 外部 DNS の使用時に DNS レコードを組織的に更新する手順	432
32.11. 既存のサーバーへの DNS サービスのインストール	434
第33章 AUTOMOUNT の使用	436
33.1. AUTOMOUNT と IDM	436
33.2. AUTOMOUNT の設定	436
33.3. KERBEROS 対応の NFS サーバーの設定	441
33.4. 場所の設定	444
33.5. マップの設定	446
パート VIII. セキュリティーの強化	454
第34章 IDENTITY MANAGEMENT 向け TLS の設定	455
34.1. HTTPD デモンの設定	455
34.2. DIRECTORY SERVER コンポーネントの設定	455
34.3. 証明書サーバーコンポーネントの設定	456
34.4. 結果	456
第35章 ANONYMOUS バインドの無効化	457
パート IX. パフォーマンスチューニング	458
第36章 エントリーの一括プロビジョニングのパフォーマンスチューニング	459
一括プロビジョニングの推奨事項と前提条件	459
現在の DS チューニングパラメーター値のバックアップ	460
データベース、ドメインエントリー、DN キャッシュサイズの調節	460
不必要なサービスの無効化およびデータベースロックの調節	462
エントリーのインポート	463
無効にしたサービスの再有効化および元の属性値の復元	463
パート X. 移行	466
第37章 LDAP ディレクトリーから IDM への移行	467
37.1. LDAP から IDM への移行に関する概要	467
37.2. IPA MIGRATE-DS の使用例	474
37.3. LDAP サーバーの IDENTITY MANAGEMENT への移行	477
37.4. SSL での移行	479
付録A トラブルシューティングのガイドライン	481

A.1. IPA ユーティリティー実行時のエラー	481
A.2. KINIT 認証エラー	483
A.3. IDM WEB UI での認証エラー	485
A.4. スマートカード認証の失敗	485
A.5. サービスが起動に失敗する理由の確認	486
A.6. DNS のトラブルシューティング	487
A.7. レプリケーションのトラブルシューティング	488
付録B トラブルシューティング: 特定問題の解決	490
B.1. IDENTITY MANAGEMENT サーバー	490
B.2. IDENTITY MANAGEMENT レプリカ	491
B.3. IDENTITY MANAGEMENT クライアント	496
B.4. ログインと認証の問題	498
付録C IDENTITY MANAGEMENT ファイルおよびログのリファレンス	501
C.1. IDENTITY MANAGEMENT 設定ファイルおよびディレクトリー	501
C.2. IDENTITY MANAGEMENT ログファイルおよびディレクトリー	503
C.3. IDM ドメインサービスとログローテーション	506
付録D ドメインレベル 0 でのレプリカの管理	508
D.1. レプリカ情報ファイル	508
D.2. レプリカの作成	508
D.3. レプリカとレプリカ合意の管理	512
D.4. レプリカのマスター CA サーバーへのプロモート	515
付録E 改訂履歴	517

パート I. RED HAT IDENTITY MANAGEMENT の概要

第1章 RED HAT IDENTITY MANAGEMENT について

本章では、Red Hat Identity Management の目的について説明します。また、Identity Management ドメイン (およびこのドメインの一部となるクライアントおよびサーバーのマシン) についての基本的な情報も提供しています。

1.1. RED HAT IDENTITY MANAGEMENT のゴール

Red Hat Identity Management (IdM) は、Linux ベースのドメイン内で ID ストア、認証ポリシーおよび承認ポリシーを一元管理する方法を提供します。IdM は異なるサービスを個別に管理するオーバーヘッドと異なるマシンで異なるツールを使用するオーバーヘッドを大幅に削減します。

IdM は以下をサポートする数少ない集中型 ID、ポリシー、および認証ソフトウェアです。

- Linux オペレーティングシステム環境の高度な機能
- Linux マシンの大規模なグループの一元化
- Active Directory を使用したネイティブな統合

IdM は Linux ベースおよび Linux 制御のドメインを作成します。

- IdM は、既存のネイティブ Linux ツールとプロトコルを基礎とします。独自のプロセスと設定がありますが、その基礎となるテクノロジーは Linux システム上で十分に確立されており、Linux 管理者から信頼されています。
- IdM サーバーとクライアントは、Red Hat Enterprise Linux マシンです。IdM は直接的には Windows クライアントに対応していませんが、Active Directory 環境との統合が可能になっています。



注記

本ガイドでは、Linux 環境における IdM の使用についてのみ説明しています。Active Directory との統合に関する詳細情報は、『[Windows 統合ガイド](#)』を参照してください。

Samba スイートを使用すると Linux マシンと Active Directory 環境との統合が可能になります。このスイートについての詳細は、『[Windows 統合ガイド](#)』の [Samba](#)、[Kerberos](#)、および [Winbind の使用](#)の章を参照してください。

1.1.1. IdM による利点

複数の Linux サーバーにおけるアイデンティティおよびポリシーの管理

IdM なしの場合: 各サーバーが個別に管理されます。パスワードはすべてローカルマシンに保存されます。IT 管理者は各マシン上でユーザーを管理し、個別に認証および承認ポリシーを設定し、ローカルのパスワードを維持します。

IdM を使用した場合: IT 管理者は以下が可能になります。

- IdM サーバーという1カ所でアイデンティティを管理。
- 複数のマシンに同時にポリシーを均一に適用。
- ホストベースのアクセス制御、委任、および他のルールを使用してユーザーに異なるアクセスレベルを設定。

- 権限昇格ルールの一元管理。
- ホームディレクトリーのマウント方法の定義。

エンタープライズシングルサインオン

IdM なしの場合: ユーザーはシステムにログインし、サービスやアプリケーションにアクセスする度にパスワードを求められます。これらのパスワードは同じものではない場合もあり、ユーザーは各アプリケーションごとに使用する認証情報を覚えている必要があります。

IdM を使用した場合: ユーザーはシステムにログインすると、認証情報を繰り返し聞かれることなく、複数のサービスやアプリケーションにアクセスできます。これにより、以下が可能になります。

- ユーザビリティの向上
- パスワードを書き留めたり安全でない場所に保存したりするセキュリティリスクの低減
- ユーザーの生産性向上

Linux と Windows の混合環境の管理

IdM なしの場合: Windows システムは Active Directory フォレストで管理されますが、開発、実稼働や他のチームには多くの Linux システムがあり、これらの Linux システムは Active Directory 環境から除外されます。

IdM を使用した場合: IT 管理者は以下が可能になります。

- ネイティブの Linux ツールを使った Linux システムの管理
- Linux システムと Windows システムの統合。これにより一元化されたユーザーストアが保持されます。
- Linux ベースを容易に拡大
- Linux と Active Directory マシンを別個に管理し、Linux と Windows 管理者が各自の環境を直接制御できます。

1.1.2. Identity Management と標準 LDAP ディレクトリーの比較

Red Hat Directory Server のような標準 LDAP ディレクトリーは汎用目的のディレクトリーで、幅広いユースケースに適用するようにカスタマイズが可能です。

- スキーマ: ユーザー、マシン、ネットワークエンティティ、物理的設備、建物といった非常に幅広いエントリー用にカスタマイズ可能な柔軟性のあるスキーマです。
- 典型的な使用例: インターネット上でサービスを提供するビジネスアプリケーションなど、他のアプリケーションのデータを保存するバックエンドのディレクトリーとして使用。

Identity Management (IdM) にはアイデンティティを管理し、その ID に関連する認証および承認ポリシーを管理するという特定の目的があります。

- スキーマ: ユーザーやマシンの ID のエントリーといった特定の目的に関連するエントリーセットを定義する特定のスキーマです。
- 典型的な使用例: 企業やプロジェクトの境界内におけるアイデンティティを管理する ID および認証サーバー。

基礎となるディレクトリーサーバーのテクノロジーは、Red Hat Directory Server と IdM で同じものです。ただし、IdM は ID 管理に最適化されています。これにより全般的な拡張性は制限されますが、シンプルな設定、リソース管理の自動化、ID 管理における効率性の向上などの利点がもたらされます。

その他のリソース

- 『Red Hat Enterprise Linux Blog』 上での [Identity Management or Red Hat Directory Server – Which One Should I Use?](#) ブログ記事。

1.2. IDENTITY MANAGEMENT ドメイン

Identity Management (IdM) ドメインは、同じ設定、ポリシー、および ID ストアを共有するマシンのグループで構成されます。プロパティを共有することで、ドメイン内のマシンは相互に認識可能となり、共同操作ができるようになります。

IdM の観点からは、ドメインには以下のタイプのマシンが含まれます。

- IdM サーバー。ドメインコントローラーとして機能します。
- IdM クライアント。これはサーバーに登録されます。

IdM サーバーは、IdM クライアントとしてサーバー自体に登録されます。サーバーマシンはクライアントと同等の機能を提供します。

IdM は、Red Hat Enterprise Linux マシンを IdM サーバーおよびクライアントとしてサポートします。



注記

本ガイドでは、Linux 環境における IdM の使用について説明しています。Active Directory との統合に関する詳細情報は、『[Windows 統合ガイド](#)』を参照してください。

1.2.1. Identity Management サーバー

IdM サーバーは、ID およびポリシー情報の中央リポジトリとして機能します。また、ドメインメンバーが使用するサービスもホストします。IdM は、IdM Web UI やコマンドラインユーティリティーなど、IdM 関連サービスをすべて 1 カ所で管理するための管理ツールセットを提供します。

IdM サーバーのインストールについての情報は、[2章Identity Management サーバーのインストールとアンインストール](#)を参照してください。

冗長性と負荷分散をサポートするために、ある IdM サーバーからこのサーバーのレプリカと呼ばれる別のサーバーにデータや設定を複製することができます。サーバーやレプリカは、クライアントに異なるサービスを提供するように設定することが可能です。IdM レプリカについての詳細情報は、[4章Identity Management のレプリカのインストールとアンインストール](#)を参照してください。

1.2.1.1. IdM サーバーでホストするサービス

以下のサービスのほとんどは、必ずしも IdM サーバー上にインストールする必要はありません。たとえば、認証局 (CA)、DNS サーバー、またはネットワークタイムプロトコル (NTP) サーバーなどのサービスは、IdM ドメイン外の外部サーバーにインストールすることができます。

Kerberos KDC

IdM は、Kerberos プロトコルを使ってシングルサインオンをサポートします。Kerberos を使用すると、ユーザーは正しいユーザー名とパスワードを 1 回提示するだけで済みます。この後は、システムが認証情報をプロンプトすることなく IdM サービスにアクセスできます。

- Kerberos の機能方法については、[システムレベルの認証ガイド](#) を参照してください。
- Kerberos を使用した IdM への認証方法については、「[Kerberos を使用した IdM へのログイン](#)」を参照してください。
- IdM での Kerberos の管理については、[28章Kerberos ドメインの管理](#) を参照してください。

LDAP ディレクトリーサーバー

IdM には内部の LDAP ディレクトリーサーバーが含まれており、ここには Kerberos 関連の情報、ユーザーアカウント、ホストエントリー、サービス、ポリシー、DNSなどの全 IdM 情報が保存されます。

LDAP ディレクトリーサーバーインスタンスは Red Hat Directory Server と同じテクノロジーをベースとしていますが、IdM 固有のタスクに調整されています。



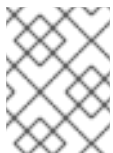
注記

本ガイドでは、このコンポーネントを Directory Server と呼びます。

認証局

ほとんどのデプロイメントでは、統合済み認証局 (CA) が IdM サーバーとインストールされます。必要な証明書すべてを独自に作成し、提供する場合は、統合 CA なしでサーバーをインストールすることもできます。

- 異なる CA 設定で IdM サーバーをインストールする詳細情報は、「[CA 設定の決定](#)」を参照してください。



注記

本ガイドでは、実装の際にはこのコンポーネントを Certificate System と呼び、実装によるサービスに対応する際には証明局と呼びます。

ドメインネームシステム (DNS)

IdM は、動的なサービス発見に DNS を使用します。IdM クライアントインストールユーティリティは、DNS からの情報を使ってクライアントマシンを自動的に設定することができます。クライアントが IdM ドメインに登録されたら、DNS を使用してドメイン内の IdM サーバーとサービスを検索します。

- サービス検索に関する詳細情報は、[システムレベルの認証ガイド](#) を参照してください。
- IdM における DNS の使用と重要な前提条件については、「[ホスト名および DNS の設定](#)」を参照してください。
- 統合 DNS ありまたはなしで IdM サーバーをインストールする詳細情報については、「[統合 DNS 使用の判断](#)」を参照してください。

ネットワークタイムプロトコルサーバー (NTP)

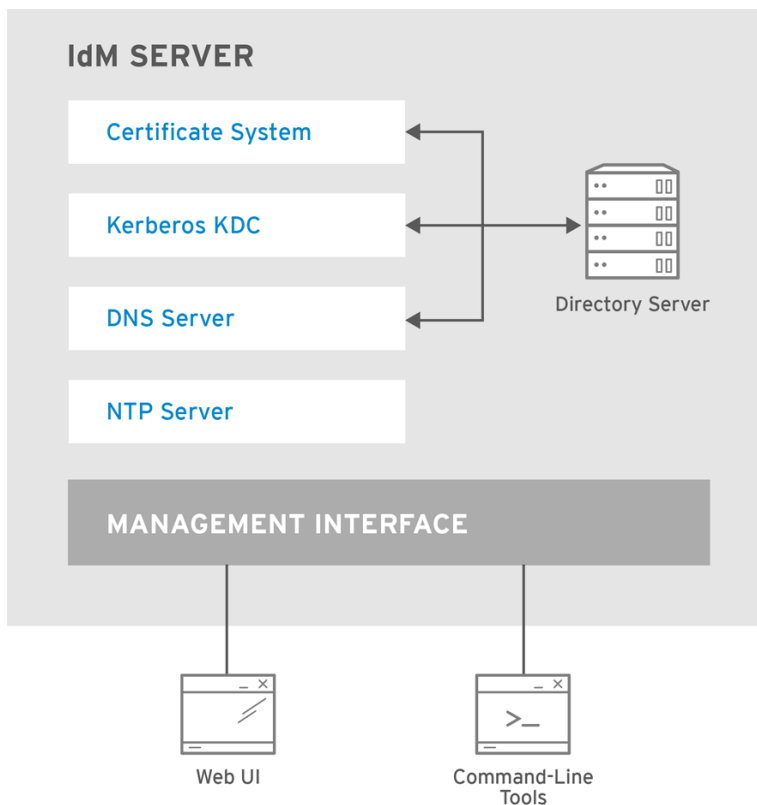
多くのサービスでは、特定の差異内でサーバーとクライアントが同一のシステムタイムを保持している必要があります。たとえば、Kerberos チケットはタイムスタンプを使ってその有効性を判断し、再生攻撃を防ぎます。サーバーとクライアントの時間の差異が許可された範囲内から逸脱すると、Kerberos チケットは無効になります。

デフォルトでは、IdM はネットワークタイムプロトコル (NTP) を使ってネットワークからクロックを同期します。NTP を使用すると、中央サーバーが権威クロックとして機能し、クライアントはこのサーバークロックに一致するようにそれぞれの時間を同期します。サーバーのインストールプロセス中は、IdM サーバーは IdM ドメイン向けの NTP サーバーとして設定されます。



注記

仮想マシン上にインストールされた IdM サーバーで NTP サーバーを稼働すると、環境によっては時間が正確に同期されない場合があります。この潜在的な問題を避けるには、仮想マシン上にインストールされた IdM サーバーで NTP を実行しないでください。仮想マシン上における NTP サーバーの信頼性については、[こちらのナレッジベースソリューション](#)を参照してください。



RHEL_404973_0516

図1.1 Identity Management サーバーによるサービスの一元管理

1.2.2. Identity Management クライアント

IdM クライアントは、IdM ドメイン内で稼働するように設定されたマシンです。IdM サーバーと対話して、ドメインのリソースにアクセスします。たとえば、クライアントがサーバー上で設定された Kerberos ドメインに属すると、サーバーが発行する証明書とチケットを受け取り、認証および承認のために他の集中化サービスを使用します。

IdM クライアントは、ドメインの一部として対話するために専用のクライアントソフトウェアを必要としません。必要になるのは、Kerberos や DNS などの特定サービスやライブラリーのシステム設定のみです。この設定で、クライアントマシンが IdM サービスを使用するように指示します。

IdM クライアントのインストールについての情報は、[3章Identity Management クライアントのインストールおよびアンインストール](#) を参照してください。

1.2.2.1. IdM クライアントがホストするサービス

System Security Services Daemon

System Security Services Daemon (SSSD) は、認証情報をキャッシュするクライアント側のアプリケーションです。クライアントマシンでの SSSD の使用は必須のクライアント設定を簡素化するので、推奨されます。SSSD は、以下のような機能も提供します。

- オフラインでのクライアント認証。中央 ID および認証ストアからの認証情報をローカルでキャッシュすることでこれを実現します。
- 認証プロセスの一貫性の改善。オフライン認証用に中央アカウントとローカルユーザーアカウントの両方を維持する必要がないため。
- **sudo** のような他のサービスとの統合。
- ホストベースのアクセス制御 (HBAC) 承認

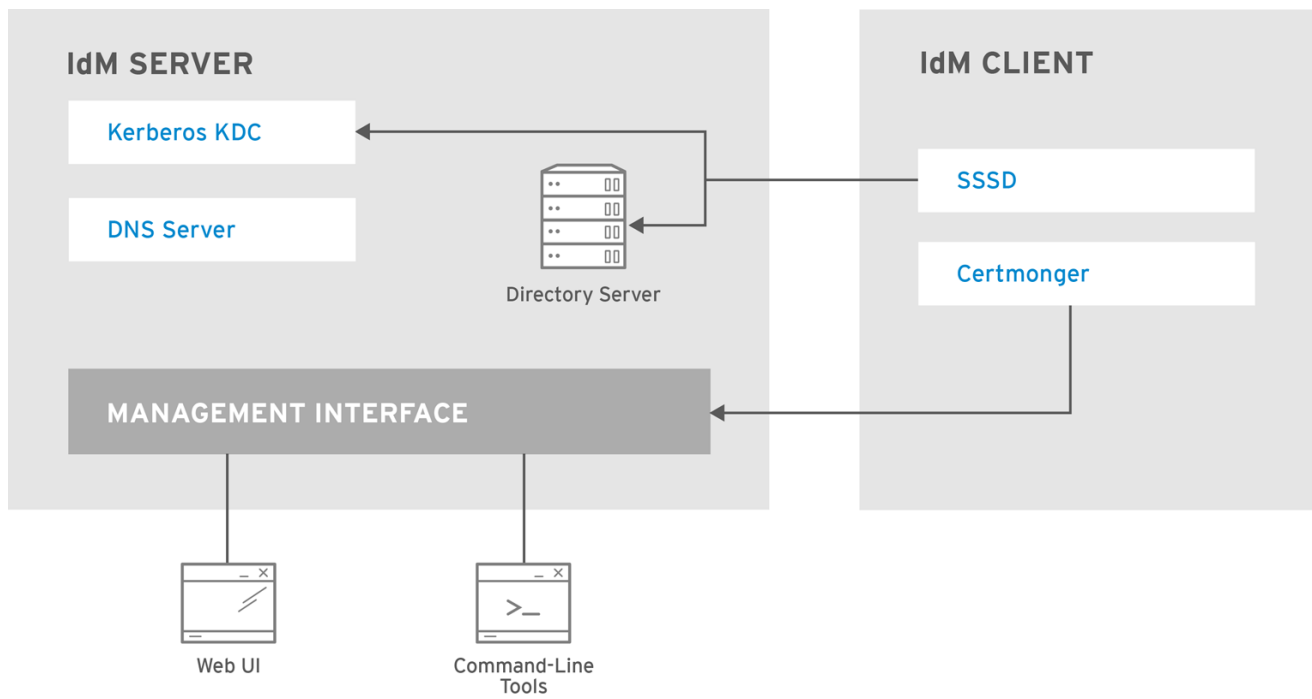
SSSD を使用すると、IdM 管理者は IdM サーバーで一括してすべてのアイデンティティ設定を定義できるようになります。IdM サーバーが利用できなくなったりクライアントがオフラインになった場合には、キャッシュによりローカルシステムは通常の認証作業を継続できます。

SSSD についての詳細情報は、[『システムレベルの認証ガイド』](#) を参照してください。SSSD は Windows Active Directory (AD) にも対応しています。AD での SSSD の使用については、[『Windows 統合ガイド』](#) を参照してください。

certmonger

certmonger サービスは、クライアント上の証明書を監視、更新します。これはシステム上のサービス向けに新規の証明書をリクエストすることができます。

certmonger についての詳細情報は、[『システムレベルの認証ガイド』](#) を参照してください。



RHEL_404973_0516

図1.2 IdM サービス間の対話

パート II. IDENTITY MANAGEMENT のインストール

第2章 IDENTITY MANAGEMENT サーバーのインストールとアンインストール

Identity Management (IdM) サーバーはドメインコントローラーで、IdM ドメインを定義して管理します。IdM サーバーを設定するには、以下を実行します。

1. 必要なパッケージをインストールします。
2. 設定スクリプトを使用してマシンを設定します。

Red Hat では、負荷分散と冗長性のためにドメイン内で複数のドメインコントローラーを設定することを強く推奨しています。これらの追加サーバーは、マスター IdM サーバーの *レプリカ* になります。

本章では、最初の IdM サーバーのインストールについて説明します。最初のサーバーからレプリカをインストールする方法については、[4章 Identity Management のレプリカのインストールとアンインストール](#) を参照してください。

2.1. サーバーインストールの前提条件

2.1.1. ハードウェア推奨事項

RAM のサイズ設定はハードウェアで最重要事項になります。必要な RAM サイズを判断するには、以下の推奨事項を考慮してください。

- 10,000 ユーザーおよび 100 グループには、最低 2GB の RAM と 1GB のスワップスペースを割り当てます。
- 100,000 ユーザーおよび 50,000 グループには、最低 16GB の RAM と 4GB のスワップスペースを割り当てます。



注記

基本的なユーザーエントリまたは証明書のあるシンプルなホストエントリのサイズは約 5 - 10 KiB になります。

大規模なデプロイメントでは、データのほとんどがキャッシュに保存されるため、ディスクスペースを増やすよりも RAM を増やす方が効果的です。

パフォーマンスを向上させるには、基礎となる Directory Server を調整することが可能です。詳細は、『Directory Server Performance Tuning Guide』の[Optimizing System Performance](#)を参照してください。

2.1.2. システム要件

Identity Management 4.4 は Red Hat Enterprise Linux 7 でサポートされています。DNS、Kerberos、または Directory Server などのサービスをカスタム設定していない、新規インストール直後のシステムに IdM サーバーをインストールします。

IdM サーバーをインストールすると、システムファイルを上書きして IdM ドメインを設定します。IdM は元のシステムファイルのバックアップを `/var/lib/ipa/sysrestore/` に作成します。

連邦情報処理標準 (FIPS: Federal Information Processing Standard) のサポート

Red Hat Enterprise Linux 7.4 以降を使用して設定した環境の場合:

- FIPS モードを有効化したシステムで、新規の IdM サーバー、レプリカまたはクライアントを設定できます。インストールスクリプトでは、管理者の介入なしに、自動的に FIPS が有効化されているシステムを検出し、IdM を設定します。

オペレーティングシステムで FIPS を有効化するには、『セキュリティーガイド』の「[FIPS モードの有効化](#)」を参照してください。



重要

以下の点に注意してください。

- FIPS モードを無効にしてインストールした既存の IdM サーバーで FIPS モードを有効にすることはできません。
- FIPS モードが無効になっている既存の IdM サーバーに FIPS サポートを有効にした新規レプリカをインストールすることはできません。

Red Hat Enterprise Linux 7.3 以前を使用して設定した環境の場合:

- IdM では、FIPS モードはサポートされません。システム上で FIPS を無効化してから、IdM サーバー、レプリカまたはクライアントをインストールし、インストール後も有効化しないでください。

FIPS モードに関する詳しい情報は『セキュリティーガイド』の「[連邦情報処理標準 \(FIPS: Federal Information Processing Standard\)](#)」を参照してください。

Name Service Cache Daemon (NSCD) の要件

Red Hat では、Identity Management マシン上で NSCD を無効にすることを推奨しています。NSCD を無効にできない場合は、代わりに SSSD がキャッシュを行わないマッピングに対して NSCD を有効化するようにしてください。

NSCD と SSSD の両サービスはキャッシングを実行するので、これら両方をシステムが同時に使用すると問題が発生します。NSCD と SSSD の競合を避ける方法については、「[システムレベルの認証ガイド](#)」を参照してください。

IPv6 がシステムで有効になっている必要がある

IdM サーバーをインストールして実行するには、IPv6 がネットワーク上で有効になっている必要があります。Red Hat Enterprise Linux 7 システムではデフォルトで IPv6 が有効になることに留意してください。

IPv6 を無効にしている場合は、Red Hat ナレッジベースの [Red Hat Enterprise Linux で IPv6 プロトコルを無効または有効にする](#) を参照して有効にします。

2.1.3. ホスト名および DNS の設定



警告

以下の点については、特に注意してください。

- テスト済みの機能する DNS サービスが利用可能であること。
- サービスが適切に設定されていること。

この要件は統合 DNS サービスのある IdM サーバーと、DNS なしでインストールされた IdM サーバーの両方に該当します。DNS レコードは、LDAP ディレクトリーサービス、Kerberos、および Active Directory 統合の実行を含むほとんどすべての IdM ドメイン機能において必須のものです。

プライマリー DNS ドメインと Kerberos レalmはインストール後には変更できないことに注意してください。

サーバーのホストは、DNS サーバーが IdM 内で統合されているか外部にホストされているかに関わらず、DNS を適切に設定する必要があります。

Identity Management は、サービスレコードに別の **DNS** ドメインを使用します。**DNS** レベルの競合を避けるために、**IdM** に使用する **プライマリー DNS** ドメインは他のシステムと共有できません。

IdM クライアントのホスト名は、プライマリー DNS ドメインの一部となる必要はないことに注意してください。



注記

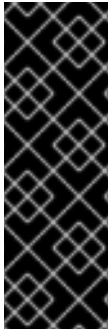
クライアント自体が IdM 参加する一方で、Active Directory DNS ドメインからのホスト名を使用してユーザーが IdM クライアントにアクセスできるようにする設定については、『Windows 統合ガイド』の [Active Directory を SSSD のアイデンティティプロバイダーとして使用する](#) を参照してください。

サーバーのホスト名の検証

ホスト名は **server.example.com** のように完全修飾ドメイン名である必要があります。使用中のマシンのホスト名を確認するには、**hostname** ユーティリティーを使用します。

```
[root@server ~]# hostname
server.example.com
```

hostname の出力は、**localhost** または **localhost6** になってはいけません。



重要

完全修飾ドメイン名は有効な DNS 名である必要があります。つまり、許可されるのは数字、アルファベット、ハイフン (-) のみです。ホスト名にアンダースコアのような他の文字があると、DNS エラーが発生します。また、ホスト名はすべて小文字を使用する必要があります。大文字は使用できません。

命名プラクティスに関する他の推奨事項については、[Red Hat Enterprise Linux セキュリティガイド](#) を参照してください。

完全修飾ドメイン名は、ループバックアドレスに解決してはいけません。マシンの公開 IP アドレスに解決する必要があります。127.0.0.1 に解決してはいけません。

正引きおよび逆引き DNS 設定の確認

1. サーバーの IP アドレスを取得します。**ip addr show** コマンドは、IPv4 と IPv6 の両方のアドレスを表示します。
 - IPv4 アドレスは、**inet** で始まる行に表示されます。以下の例では、設定済み IPv4 アドレスは **192.0.2.1** になります。
 - IPv6 アドレスは **inet6** で始まる行に表示されます。**scope global** のある IPv6 のみがこの手順では関連してきます。以下の例では、返される IPv6 アドレスは **2001:DB8::1111** になります。

```
[root@server ~]# ip addr show
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 00:1a:4a:10:4e:33 brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.1/24 brd 192.0.2.255 scope global dynamic eth0
        valid_lft 106694sec preferred_lft 106694sec
    inet6 2001:DB8::1111/32 scope global dynamic
        valid_lft 2591521sec preferred_lft 604321sec
    inet6 fe80::56ee:75ff:fe2b:def6/64 scope link
        valid_lft forever preferred_lft forever
```

2. **dig** ユーティリティにホスト名を加えて、正引き DNS 設定を確認します。

1. **dig +short server.example.com A** コマンドを実行します。返される IPv4 アドレスは、**ip addr show** が返す IP アドレスと一致する必要があります。

```
[root@server ~]# dig +short server.example.com A
192.0.2.1
```

2. **dig +short server.example.com AAAA** コマンドを実行します。コマンドがアドレスを返す場合は、**ip addr show** が返す IPv6 アドレスと一致する必要があります。

```
[root@server ~]# dig +short server.example.com AAAA
2001:DB8::1111
```



注記

AAAA レコードの出力が返されない場合でも、設定が間違っているわけではありません。出力がないということは、DNS 内でサーバーマシン向けに IPv6 アドレスが設定されていないというだけのことです。ネットワークで IPv6 プロトコルを使用する予定がない場合は、この状況でもインストールを続行できます。

3. **dig** ユーティリティに IP アドレスを加えて、逆引き DNS 設定 (PTR レコード) を確認します。

1. **dig +short -x IPv4 address** コマンドを実行します。サーバーのホスト名が出力に表示される必要があります。例を示します。

```
[root@server ~]# dig +short -x 192.0.2.1
server.example.com
```

2. 前のステップで **dig +short -x server.example.com AAAA** コマンドが IPv6 アドレスを返した場合は、**dig** を使って IPv6 アドレスもクエリします。ここでも、サーバーのホスト名が出力に表示される必要があります。例を示します。

```
[root@server ~]# dig +short -x 2001:DB8::1111
server.example.com
```



注記

前のステップで **dig +short -x server.example.com AAAA** コマンドが IPv6 アドレスを返さなかった場合は、AAAA レコードのクエリは何も出力しません。これは正常な動作で、設定が間違っていることを示すものではありません。

前のステップで **dig +short server.example.com** が IP アドレスを返した場合でも、別のホスト名が表示されたりホスト名が表示されない場合は、逆引き DNS 設定が間違っていることになります。

DNS フォワーダーの標準準拠の確認

統合 DNS の IdM を設定する際には、IdM DNS サーバーで使用するすべての DNS フォワーダーが [Extension Mechanisms for DNS \(EDNS0\)](#) と [DNS Security Extensions \(DNSSEC\)](#) の標準に準拠していることを確認してください。これを実行するには、各フォワーダーごとに個別に以下のコマンドの出力をチェックします。

```
$ dig +dnssec @IP_address_of_the_DNS_forwarder . SOA
```

コマンドの出力には、以下の情報が含まれているはずです。

- status: **NOERROR**
- flags: **ra**
- EDNS flags: **do**
- **ANSWER** セクションには **RRSIG** レコードがある必要があります。

これらのいずれかがない場合は、使用している DNS フォワーダーのドキュメントをチェックして、

EDNS0 と DNSSEC がサポートされかつ有効になっていることを確認してください。BIND サーバーの最新バージョンでは、`/etc/named.conf` ファイルで **dnssec-enable yes**；オプションが設定されている必要があります。

出力例は以下のようになります。

```
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 48655
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096

;; ANSWER SECTION:
. 31679 IN SOA a.root-servers.net. nstld.verisign-grs.com. 2015100701 1800
900 604800 86400
. 31679 IN RRSIG SOA 8 0 86400 20151017170000 20151007160000 62530 .
GNVz7SQs [...]
```

/etc/hosts ファイル



重要

`/etc/hosts` ファイルは手動で変更しないでください。`/etc/hosts` を変更した場合は、コンテンツが以下のルールに準拠していることを確認してください。

以下は、`/etc/hosts` ファイルが正しく設定されている例です。ホストの IPv4 および IPv6 localhost エントリーが適切に表示され、最初のエントリーで IdM サーバーの IP アドレスとホスト名がその後に続いています。IdM サーバーのホスト名は **localhost** エントリーに含めることができない点に注意してください。

```
127.0.0.1 localhost.localdomain localhost
::1 localhost6.localdomain6 localhost6
192.0.2.1 server.example.com server
2001:DB8::1111 server.example.com server
```

2.1.4. ポート要件

IdM はサービスとの通信に多くのポートを使用します。IdM が機能するには、これらのポートが開いて利用可能になっている必要があります。別のサービスが使用していたり、ファイアウォールがブロックしてはいけません。

- 必須ポートの一覧は、[「必須ポート一覧」](#) を参照してください。
- 必須ポートに対応している **firewalld** サービスの一覧は、[「firewalld サービスの一覧」](#) を参照してください。

必須ポート一覧

表2.1 Identity Management のポート

サービス	ポート	プロトコル
HTTP/HTTPS	80、443	TCP

サービス	ポート	プロトコル
LDAP/LDAPS	389、636	TCP
Kerberos	88、464	TCP および UDP
DNS	53	TCP および UDP
NTP	123	UDP

注記

IdM がポート 80 および 389 を使用していることについて心配は要りません。

- ポート 80 (HTTP) は、オンライン証明書ステータスプロトコル (OCSP) の応答と証明取り消し一覧 (CRL) を提供するために使用されます。これらは両方ともデジタル署名されているので、中間者攻撃に対して安全になっています。
- ポート 389 (LDAP) は暗号化に STARTTLS と GSSAPI を使用します。

これらに加えて、IdM はポート 8080 でリッスンすることができ、インストールによってはポート 8443 および 749 でリッスンできるものもあります。しかし、これら 3 つのポートは内部使用のみです。IdM はこれらをオープンにしておきますが、外部からアクセス可能である必要はありません。ポート 8080、8443、および 749 をオープンにすることは推奨されません。代わりにファイアウォールでこれらをブロックしてください。

firewalld サービスの一覧

表2.2 firewalld サービス

サービス名	詳細参照先
freeipa-ldap	/usr/lib/firewalld/services/freeipa-ldap.xml
freeipa-ldaps	/usr/lib/firewalld/services/freeipa-ldaps.xml
dns	/usr/lib/firewalld/services/dns.xml

必須ポートの開放

1. **firewalld** サービスが稼働していることを確認します。

- 。 **firewalld** が実行中かどうかを確認するには、以下を実行します。

```
# systemctl status firewalld.service
```

- 。 **firewalld** を起動し、システム起動時に自動的に起動するように設定するには、以下を実行します。

```
# systemctl start firewalld.service
# systemctl enable firewalld.service
```

2. **firewall-cmd** ユーティリティーを使って必須ポートを開きます。以下のいずれかのオプションを選択します。

- a. **firewall-cmd --add-port** コマンドを使用して個別ポートをファイアウォールに追加します。たとえば、デフォルトゾーンのポートを開くには、以下を実行します。

```
# firewall-cmd --permanent --add-port=
{80/tcp,443/tcp,list_of_ports}
```

- b. **firewall-cmd --add-service** コマンドを使用して **firewalld** サービスをファイアウォールに追加します。たとえば、デフォルトゾーンのポートを開くには、以下を実行します。

```
# firewall-cmd --permanent --add-service={freeipa-
ldap,list_of_services}
```

firewall-cmd を使用してシステム上でポートを開く方法についての詳細は、[『セキュリティガイド』](#) または `firewall-cmd(1) man` ページを参照してください。

3. **firewall-cmd** 設定をリロードして、変更が直ちに反映されるようにします。

```
# firewall-cmd --reload
```

実稼働環境のシステムで **firewalld** を再読み込みすると、DNS 接続がタイムアウトされてしまう可能性があります。[『Security Guide』の「コマンドラインインターフェース \(CLI\) を使ったファイアウォール設定のリロード」](#) も参照してください。必要であれば、タイムアウトのリスクを回避するため、**--permanent** オプションなしでこのコマンドを再度実行して、実行中のシステムに変更を適用します。

4. これはオプションです。ポートが現在使用可能であることを確認するには、**nc**、**telnet**、または **nmap** のユーティリティーを使用してポートに接続するか、ポートスキャンを実行します。

2.2. IDM サーバーのインストールに必要なパッケージ

統合 DNS サービスなしでサーバーに必須のパッケージをインストールするには、以下を実行します。

```
# yum install ipa-server
```

統合 DNS サービスのあるサーバーに必須のパッケージをインストールするには、以下を実行します。

```
# yum install ipa-server ipa-server-dns
```



注記

ご自分のユースケースに DNS が適切かどうかを判断するには、[「統合 DNS 使用の判断」](#) を参照してください。

ipa-server パッケージは自動的に以下のような他の必須のパッケージを依存関係としてインストールします。

- Directory Server LDAP サービス向けの 389-ds-base
- Kerberos サービス向けの krb5-server パッケージ
- 各種の IdM 固有ツール

2.3. IDM サーバーのインストール: はじめに



注記

以下のインストールの手順および例は相互排他的ではなく、組み合わせて、求める結果を得ることができます。たとえば、統合 DNS のあるサーバーを外部にホストされた root CA とともにインストールすることが可能です。

ipa-server-install で IdM のインストールと設定を行います。

サーバーのインストール前に、以下のセクションを参照してください。

- [「統合 DNS 使用の判断」](#)
- [「CA 設定の決定」](#)

ipa-server-install ユーティリティーでは非対話式のインストールモードが提供され、これを使用することで自動かつ無人のサーバー設定が可能になります。詳細は、[「非対話式でのサーバーのインストール」](#) を参照してください。

ipa-server-install は、ログファイルを **/var/log/ipaserver-install.log** に作成します。インストールが失敗した場合は、このログファイルが問題の特定に役立ちます。

2.3.1. 統合 DNS 使用の判断

IdM は、統合 DNS ありまたはなしの両方のサーバーのインストールをサポートしています。

統合 DNS サービスのある IdM サーバー

IdM が提供する統合 DNS サーバーは、汎用目的の DNS サーバーとして使用する設計にはなっていません。IdM デプロイメントとメンテナンスに関連する機能のみをサポートしています。高度な DNS 機能のいくつかはサポートされていません。

Red Hat では、IdM デプロイメント内で基本的な IdM-統合 DNS の使用を強く推奨しています。IdM サーバーが DNS も管理する場合は、DNS とネイティブの IdM ツールは緊密に統合され、DNS レコード管理の一部の自動化が可能になります。

IdM サーバーがマスター DNS サーバーとして使用される場合でも、他の外部の DNS サーバーはスレーブサーバーとして使用することが可能であることに留意してください。

たとえば、Active Directory 統合 DNS サーバーのような別の DNS サーバーを自分の環境で既に使用している場合、IdM 統合 DNS に委任できるのは IdM プライマリドメインのみになります。DNS ゾーンを IdM 統合 DNS に移行する必要はありません。

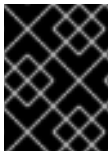
統合 DNS のあるサーバーをインストールする方法については、[「統合 DNS のあるサーバーのインストール」](#) を参照してください。

統合 DNS サービスのない IdM サーバー

DNS サービスの提供に外部の DNS サーバーが使用されます。以下の場合、DNS なしで IdM サーバーをインストールすることを検討してください。

- IdM DNS のスコープ外となる高度な DNS 機能を必要とする場合。
- 外部の DNS サーバーが使用可能となっている確立された DNS インフラストラクチャーがある環境。

統合 DNS のないサーバーをインストールする方法については、「[統合 DNS のなしでのサーバーのインストール](#)」を参照してください。



重要

「[ホスト名および DNS の設定](#)」に記載の DNS 要件をシステムが満たしていることを確認してください。

統合または外部 DNS のメンテナンス要件

統合 DNS サーバーを使用する場合は、ほとんどの DNS レコードのメンテナンスは自動で行われます。必要なのは以下の点のみになります。

- 親ドメインから IdM サーバーへの適切な委任の設定

たとえば、IdM ドメイン名が **ipa.example.com** だったとすると、**example.com** ドメインからの適切な委任が必要になります。



注記

以下のコマンドを使用すると委任を確認できます。

```
# dig @IP_address +norecurse +short ipa.example.com. NS
```

IP_address は、**example.com** DNS ドメインを管理するサーバーの IP アドレスです。委任が適切であれば、このコマンドにより DNS サーバーがインストール済みの IdM サーバーが一覧表示されます。

外部 DNS サーバーを使用する場合は、以下の点が必須になります。

- DNS サーバー上に新規ドメインを手動で作成する。
- 新規ドメインを、IdM インストーラーで生成されたゾーンファイルからのレコードで手動で満たす。
- Active Directory の信頼設定後などのサービス設定の変更後や、レプリカのインストールもしくは削除後にレコードを手動で更新する。

DNS アンプ攻撃の回避

IdM-統合 DNS サーバーのデフォルト設定では、全クライアントが DNS サーバーに再帰クエリーを発行することが可能になります。サーバーが信頼されないクライアントのあるネットワークにデプロイされている場合は、再帰を承認済みクライアントのみに制限するようにサーバーの設定を変更してください。[1]

承認クライアントのみが再帰クエリーを発行できるようにするには、適切なアクセス制御リスト (ACL) ステートメントをサーバー上の `/etc/named.conf` ファイルに追加します。例を示します。

```
acl authorized { 192.0.2.0/24; 198.51.100.0/24; };
options {
    allow-query { any; };
    allow-recursion { authorized; };
};
```

2.3.2. CA 設定の決定

IdM は、統合 IdM 証明局 (CA) あり、または CA なしでのサーバーのインストールをサポートしています。

統合 IdM CA のあるサーバー

これはほとんどのデプロイメントに適切なデフォルトの設定です。Certificate System は CA 署名証明書を使用して IdM ドメイン内の証明書を作成し、これに署名します。



警告

Red Hat では、複数のサーバーに CA サービスをインストールしておくことを強く推奨しています。CA サービスを含む最初のサーバーのレプリカをインストールする方法についての情報は、[「CA を設定したレプリカのインストール」](#)を参照してください。

CA が 1 つのサーバーにしかインストールされていないと、CA サーバーが故障した際に CA 設定が失われて回復できない恐れがあります。詳細については、[「失われた CA サーバーの復旧」](#)を参照してください。

CA 署名証明書は、CA 階層の中で最高位の CA である *root CA* で署名される必要があります。root CA は IdM CA 自体であったり、外部でホストされている CA である場合もあります。

IdM CA を root CA とする

これがデフォルト設定になります。

この設定でサーバーをインストールする方法については、[「統合 DNS のあるサーバーのインストール」](#)と[「統合 DNS のなしでのサーバーのインストール」](#)を参照してください。

外部 CA を root CA とする

IdM CA は外部 CA の下位となります。ただし、IdM ドメインの証明書はすべて、Certificate System インスタンスが発行します。

外部 CA は、企業 CA や、Verisign や Thawte などのサードパーティー CA とすることが出来ます。IdM ドメイン内で発行される証明書は、有効期間など外部 root CA の属性が設定する制限の影響を受ける可能性があります。

外部にホストされている root CA のあるサーバーをインストールする方法については、[「外部 CA を Root CA としてサーバーをインストールする手順」](#)を参照してください。

CA なしのサーバー

この設定オプションは、インフラストラクチャー内の制限により証明書サービスのあるサーバーをインストールできない場合に適しています。

インストール前に以下の証明書をサードパーティー機関にリクエストする必要があります。

- LDAP サーバー証明書および秘密キー
- Apache サーバー証明書および秘密キー
- LDAP および Apache サーバー証明書を発行した CA の完全な CA 証明書チェーン

統合 IdM CA なしで証明書を管理しようとする、多大なメンテナンス負担になります。たとえば、

- 証明書の作成、アップロード、更新プロセスが手動になります。
- 証明書の追跡に **certmonger** サービスが使用されません。このため、証明書の有効期限が迫っても警告が出されません。

統合 CA のないサーバーをインストールする方法については、[「CA なしでのインストール」](#) を参照してください。

2.3.3. 統合 DNS のあるサーバーのインストール



注記

どの DNS または CA 設定がご使用の環境に適切かわからない場合は、[「統合 DNS 使用の判断」](#) と [「CA 設定の決定」](#) を参照してください。

統合 DNS のあるサーバーをインストールするには、インストールプロセス中に以下の情報を提供する必要があります。

DNS フォワーダー

以下の DNS フォワーダー設定がサポートされています。

- 1 つ以上のフォワーダー (非対話式インストールでの **--forwarder** オプション)
- フォワーダーなし (非対話式インストールでの **--no-forwarders** オプション)

ご自分のユースケースに DNS フォワーダーを使用すべきかどうかを判断するには、[「DNS 転送の管理」](#) を参照してください。

逆引き DNS ゾーン

以下の DNS ゾーン設定がサポートされています。

- IdM DNS 内で作成する必要がある逆引きゾーンの自動検出 (対話式インストールでのデフォルト設定、非対話式インストールでの **--auto-reverse** オプション)
- 逆引きゾーンを自動検出しない (対話式インストールでの **--no-reverse** オプション)

非対話式インストールでは、**--setup-dns** オプションも追加してください。

-

例2.1 統合 DNS のあるサーバーのインストール

この手順では、以下のサーバーをインストールします。

- 統合 DNS のあるサーバー
- IdM CA を root CA とするサーバー。これがデフォルトの CA 設定です。

1. **ipa-server-install** ユーティリティを実行します。

```
# ipa-server-install
```

2. このスクリプトは統合 DNS サービスを設定するよう要求するので、**yes** と入力します。

```
Do you want to configure integrated DNS (BIND)? [no]: yes
```

3. さらにいくつかの設定プロンプトが出ます。

- 括弧内のデフォルト値を許可するには、**Enter** を押します。
- デフォルト値とは別の値を使用する場合は、必要な値を入力します。

```
Server host name [server.example.com]:
Please confirm the domain name [example.com]:
Please provide a realm name [EXAMPLE.COM]:
```



警告

Red Hat では、Kerberos レalm 名をプライマリー DNS ドメイン名をすべて大文字にしたものにすることを強く推奨しています。たとえば、プライマリー DNS ドメインが **ipa.example.com** の場合、Kerberos レalm 名は **IPA.EXAMPLE.COM** とします。

異なる命名規則を使用すると Active Directory 信頼が使用できなくなるほか、その他のマイナス面が発生する可能性があります。

4. Directory Server スーパーユーザー、**cn=Directory Manager**、および **admin** IdM システムユーザーアカウントのパスワードを入力します。

```
Directory Manager password:
IPA admin password:
```

5. DNS フォワーダー設定のプロンプトが出されます。

```
Do you want to configure DNS forwarders? [yes]:
```

- DNS フォワーダーを設定する場合は、**yes** を入力してコマンドラインの指示に従います。

インストールプロセスでフォワーダー IP アドレスがインストールされる IdM サーバーの `/etc/named.conf` ファイルに追加されます。

- 転送ポリシーのデフォルト設定については、`ipa-dns-install(1)` man ページの `--forward-policy` の記述を参照してください。
- 詳細は、「[転送ポリシー](#)」も参照してください。

。DNS 転送を使用しない場合は、**no** と入力します。

6. サーバーと関連する IP アドレスの DNS 逆引き (PTR) レコードを設定する必要性を確認するプロンプトが出されます。

```
Do you want to search for missing reverse zones? [yes]:
```

検索を実行して逆引きゾーンが見つかり、PTR レコードの逆引きゾーンを作成するかどうかを聞かれます。

```
Do you want to create reverse zone for IP 192.0.2.1 [yes]:
Please specify the reverse zone name [2.0.192.in-addr.arpa.]:
Using reverse zone(s) 2.0.192.in-addr.arpa.
```



注記

逆引きゾーンの管理に IdM を使用することはオプションです。外部 DNS サービスを使用することもできます。

7. サーバー設定をする場合は、**yes** と入力します。

```
Continue to configure the system with these values? [no]: yes
```

8. これでインストールスクリプトがサーバーを設定します。動作が完了するまで待機します。
9. 親ドメインからのDNS 委任を IdM DNS ドメインに追加します。たとえば、IdM DNS ドメインが `ipa.example.com` の場合、ネームサーバー (NS) レコードを `example.com` の親ドメインに追加します。



重要

IdM DNS サーバーがインストールされるたびに毎回このステップを繰り返す必要があります。

このスクリプトは、CA 証明書をバックアップし、必要なネットワークポートを解放することを提案します。IdM ポートの要件およびこれらのポートを解放する方法に関する情報は、「[ポート要件](#)」を参照してください。

以下の手順で新規サーバーをテストします。

1. `admin` の認証情報を使って Kerberos レルムに認証を行います。これで **admin** が適切に設定され、Kerberos レルムがアクセス可能であることを確認します。

```
# kinit admin
```

2. **ipa user-find** のようなコマンドを実行します。新規サーバーでは、このコマンドは唯一の設定済みユーザーである **admin** をプリントします。

```
# ipa user-find admin
-----
1 user matched
-----
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
UID: 939000000
GID: 939000000
Account disabled: False
Password: True
Kerberos keys available: True
-----
Number of entries returned 1
-----
```

2.3.4. 統合 DNS のなしでのサーバーのインストール



注記

どの DNS または CA 設定がご使用の環境に適切か分からない場合は、「[統合 DNS 使用の判断](#)」と「[CA 設定の決定](#)」を参照してください。

統合 DNS のないサーバーをインストールするには、DNS 関連のオプションなしで **ipa-server-install** ユーティリティを実行します。

例2.2 統合 DNS のなしでのサーバーのインストール

この手順では、以下のサーバーをインストールします。

- 統合 DNS のないサーバー
- IdM CA を root CA とするサーバー。これがデフォルトの CA 設定です。

1. **ipa-server-install** ユーティリティを実行します。

```
# ipa-server-install
```

2. このスクリプトは統合 DNS サービスを設定するよう要求するので、**Enter** を押してデフォルトの **no** オプションを選択します。

```
Do you want to configure integrated DNS (BIND)? [no]:
```

3. さらにいくつかの設定プロンプトが出ます。
 - 括弧内のデフォルト値を許可するには、**Enter** を押します。
 - デフォルト値とは別の値を使用する場合は、必要な値を入力します。

```
Server host name [server.example.com]:
Please confirm the domain name [example.com]:
Please provide a realm name [EXAMPLE.COM]:
```



警告

Red Hat では、Kerberos レalm 名をプライマリー DNS ドメイン名をすべて大文字にしたものにすることを強く推奨しています。たとえば、プライマリー DNS ドメインが **ipa.example.com** の場合、Kerberos レalm 名は **IPA.EXAMPLE.COM** とします。

異なる命名規則を使用すると Active Directory 信頼が使用できなくなるほか、その他のマイナス面が発生する可能性があります。

4. Directory Server スーパーユーザー、**cn=Directory Manager**、および **admin** IdM システムユーザーアカウントのパスワードを入力します。

```
Directory Manager password:
IPA admin password:
```

5. サーバー設定をする場合は、**yes** と入力します。

```
Continue to configure the system with these values? [no]: yes
```

6. これでインストールスクリプトがサーバーを設定します。動作が完了するまで待機します。
7. 以下の出力例のように、インストールスクリプトでは DNS リソースレコードが含まれるファイルが (**/tmp/ipa.system.records.UFRPto.db**) が作成されます。これらのレコードを既存の外部 DNS サーバーに追加します。DNS レコードの更新プロセスは、特定の DNS ソリューションにより異なります。

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



重要

既存の DNS サーバーに DNS レコードを追加した時点で、サーバーのインストールは完了します。

このスクリプトは、CA 証明書をバックアップし、必要なネットワークポートを解放することを提案します。IdM ポートの要件およびこれらのポートを解放する方法に関する情報は、[「ポート要件」](#)を参照してください。

以下の手順で新規サーバーをテストします。

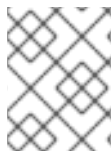
1. admin の認証情報を使って Kerberos レalmに認証を行います。これで **admin** が適切に設定され、Kerberos レalmがアクセス可能であることを確認します。

```
# kinit admin
```

2. **ipa user-find** のようなコマンドを実行します。新規サーバーでは、このコマンドは唯一の設定済みユーザーである **admin** をプリントします。

```
# ipa user-find admin
-----
1 user matched
-----
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
UID: 939000000
GID: 939000000
Account disabled: False
Password: True
Kerberos keys available: True
-----
Number of entries returned 1
-----
```

2.3.5. 外部 CA を Root CA としてサーバーをインストールする手順



注記

どの DNS または CA 設定がご使用の環境に適切か分からない場合は、「[統合 DNS 使用の判断](#)」と「[CA 設定の決定](#)」を参照してください。

サーバーをインストールして、root CA として外部 CA とそのサーバーをつなぐには、**ipa-server-install** ユーティリティーで以下のオプションを渡します。

- **--external-ca** で外部 CA を使用することを指定します。
- **--external-ca-type** では、外部 CA のタイプを指定します。詳細については、ipa-server-install(1) man ページを参照してください。

この他については、インストール手順のほとんどは「[統合 DNS のあるサーバーのインストール](#)」または「[統合 DNS のなしでのサーバーのインストール](#)」の場合と同じになります。

Certificate System インスタンスの設定中、このユーティリティーは証明書署名要求 (CSR) の位置を出力します: **/root/ipa.csr**:

```
...
```

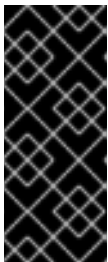
```
Configuring certificate server (pki-tomcatd): Estimated time 3 minutes 30
seconds
[1/8]: creating certificate server user
```

[2/8]: configuring certificate server instance

The next step is to get /root/ipa.csr signed by your CA and re-run /sbin/ipa-server-install as: /sbin/ipa-server-install --external-cert-file=/path/to/signed_certificate --external-cert-file=/path/to/external_ca_certificate

これが発生したら、以下を実行します。

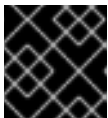
1. **/root/ipa.csr** にある CSR を外部 CA に提出します。このプロセスは、外部 CA として使用するサービスによって異なります。



重要

場合によっては、証明書に適切な拡張子を要求する必要がある場合もあります。Identity Management サーバー用に生成された CA 署名証明書は、有効な CA 証明書である必要があります。つまり、Basic Constraint が **CA=true** と設定されているか、Key Usage Extension が署名証明書に設定されて、証明書の署名が可能となっている必要があります。

2. 発行された証明書と Base64 エンコードされたプロブ (PEM ファイルか Windows CA からの Base_64 証明書) で CA を発行するための CA 証明書チェーンを取得します。プロセスは証明書サービスによって異なりますが、通常はウェブページか通知メールにダウンロードリンクがあり、管理者が必要な証明書すべてをダウンロードできるようになっています。



重要

CA 証明書のみではなく、CA 用の完全な証明書チェーンを取得してください。

3. **ipa-server-install** を再度実行し、新規に発行された CA 証明書と CA チェーンファイルの場所と名前を指定します。例を示します。

```
# ipa-server-install --external-cert-
file=/tmp/servercert20110601.pem --external-cert-
file=/tmp/cacert.pem
```



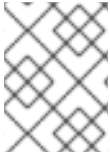
注記

ipa-server-install --external-ca コマンドは、以下のエラーが出て失敗する場合があります。

```
ipa          : CRITICAL failed to configure ca instance Command
'/usr/sbin/pkispawn -s CA -f /tmp/configuration_file' returned
non-zero exit status 1
Configuration of CA failed
```

これは *_**proxy** 環境変数が設定されている場合に発生します。この問題の解決策については、「[外部 CA インストールの失敗](#)」を参照してください。

2.3.6. CA なしでのインストール



注記

どの DNS または CA 設定がご使用の環境に適切か分からない場合は、「[統合 DNS 使用の判断](#)」と「[CA 設定の決定](#)」を参照してください。

CA なしでサーバーをインストールするには、オプションを **ipa-server-install** ユーティリティーに追加することで必要な証明書を手動で提供する必要があります。この他については、インストール手順のほとんどは「[統合 DNS のあるサーバーのインストール](#)」または「[統合 DNS のなしでのサーバーのインストール](#)」の場合と同じになります。



重要

サードパーティーの自己署名サーバー証明書を使用して、サーバーまたはレプリカをインストールできません。

LDAP サーバー証明書および秘密キーを提供するには、以下のオプションを使用します。

- **--dirsrv-cert-file** は、LDAP サーバー証明書用の証明書ファイルおよび秘密キーファイルを提供します。
- **--dirsrv-pin** では、**--dirsrv-cert-file** で指定されたファイル内の秘密キーにアクセスするためのパスワードを提供します。

Apache サーバー証明書および秘密鍵を提供するには、以下のオプションを使用します。

- **--http-cert-file** では、Apache サーバー証明書用の証明書ファイルおよび秘密鍵ファイルを提供します。
- **--http-pin** では、**--http-cert-file** で指定されたファイル内の秘密鍵にアクセスするためのパスワードを提供します。

LDAP および Apache サーバー証明書を発行した CA の完全な CA 証明書チェーンを提供するには、以下のオプションを使用します。

- **--dirsrv-cert-file** および **--http-cert-file** で、完全な CA 証明書チェーンもしくはその一部を備えた証明書ファイルを提供します。

これらのオプションは複数回使用することができ、以下を受け付けます。

- PEM エンコード化および DER エンコード化された X.509 証明書ファイル
- PKCS#1 および PKCS#8 秘密鍵ファイル
- PKCS#7 証明書チェーンファイル
- PKCS#12 ファイル

--dirsrv-cert-file と **--http-cert-file** を使用して提供されるファイルには、厳密に 1 つのサーバー証明書と 1 つの秘密キーが含まれている必要があります。**--dirsrv-cert-file** と **--http-cert-file** を使用して提供されるファイルのコンテンツは同一であることがよくあります。

- 必要に応じて **--ca-cert-file** を証明書ファイルに追加し、完全な CA 証明書チェーンを完成させます。

このオプションは複数回使用することができ、以下を受け付けます。

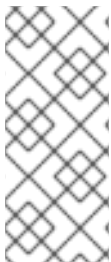
- PEM エンコード化および DER エンコード化された X.509 証明書ファイル
- PKCS#7 証明書チェーンファイル

--ca-cert-file で提供されるファイルと組み合わせて、**--dirsrv-cert-file** と **--http-cert-file** で提供されるファイルには、LDAP および Apache サーバー証明書を発行した CA の完全 CA 証明書チェーンが含まれる必要があります。

以下に例を示します。

```
[root@server ~]# ipa-server-install \
--http-cert-file /tmp/server.crt \
--http-cert-file /tmp/server.key \
--http-pin secret \
--dirsrv-cert-file /tmp/server.crt \
--dirsrv-cert-file /tmp/server.key \
--dirsrv-pin secret \
--ca-cert-file ca.crt
```

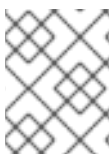
このセクションのコマンドラインオプションは、**--external-ca** オプションと互換性がないことに注意してください。



注記

Identity Management の以前のバージョンでは、**--root-ca-file** オプションを使って root CA 証明書の PEM ファイルを指定していました。信頼できる CA は常に DS および HTTP サーバー証明書の発行者なので、これはもう不要になりました。今では IdM は、**--dirsrv-cert-file**、**--http-cert-file**、および **--ca-cert-file** で指定される証明書からの root CA 証明書を自動的に認識します。

2.3.7. 非対話式でのサーバーのインストール



注記

どの DNS または CA 設定がご使用の環境に適切か分からない場合は、[「統合 DNS 使用の判断」](#) と [「CA 設定の決定」](#) を参照してください。

非対話式でのインストールで最小限必要となるオプションは以下の通りです。

- **--ds-password** では、Directory Server のスーパーユーザーである Directory Manager (DM) のパスワードを指定します。
- **--admin-password** では、IdM 管理者である **admin** のパスワードを指定します。
- **--realm** では、Kerberos レalm名を指定します。
- **--unattended** を使用すると、インストールプロセスでホスト名とドメイン名のデフォルトオプションを選択することができます。

任意で、以下の設定でカスタム値を指定できます。

- **--hostname** でサーバーのホスト名を指定します。
- **--domain** でドメイン名を指定します。



警告

Red Hat では、Kerberos レalm 名をプライマリー DNS ドメイン名をすべて大文字にしたものにすることを強く推奨しています。たとえば、プライマリー DNS ドメインが **ipa.example.com** の場合、Kerberos レalm 名は **IPA.EXAMPLE.COM** とします。

異なる命名規則を使用すると Active Directory 信頼が使用できなくなるほか、その他のマイナス面が発生する可能性があります。

ipa-server-install が受け取るオプションの完全なリストを表示するには、**ipa-server-install --help** コマンドを実行してください。

例2.3 非対話式の基本的なインストール

1. **ipa-server-install** ユーティリティーを実行して、必要な設定を指定します。たとえば、以下では統合 DNS がなく統合 CA のあるサーバーがインストールされます。

```
# ipa-server-install --realm EXAMPLE.COM --ds-password
DM_password --admin-password admin_password --unattended
```

2. これで設定スクリプトがサーバーを設定します。動作が完了するまで待機します。
3. 以下の出力例のように、インストールスクリプトでは DNS リソースレコードが含まれるファイルが (**/tmp/ipa.system.records.UFRPto.db**) が作成されます。これらのレコードを既存の外部 DNS サーバーに追加します。DNS レコードの更新プロセスは、特定の DNS ソリューションにより異なります。

```
...
Restarting the KDC
Please add records in this file to your DNS system:
/tmp/ipa.system.records.UFRBto.db
Restarting the web server
...
```



重要

既存の DNS サーバーに DNS レコードを追加した時点で、サーバーのインストールは完了します。

このスクリプトは、CA 証明書をバックアップし、必要なネットワークポートを解放することを提案します。IdM ポートの要件およびこれらのポートを解放する方法に関する情報は、[「ポート要件」](#)を参照してください。

以下の手順で新規サーバーをテストします。

1. **admin** の認証情報を使って Kerberos レalm に認証を行います。これで **admin** が適切に設定され、Kerberos レalm がアクセス可能であることを確認します。

```
# kinit admin
```

2. **ipa user-find** のようなコマンドを実行します。新規サーバーでは、このコマンドは唯一の設定済みユーザーである **admin** をプリントします。

```
# ipa user-find admin
-----
1 user matched
-----
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
UID: 939000000
GID: 939000000
Account disabled: False
Password: True
Kerberos keys available: True
-----
Number of entries returned 1
-----
```

2.4. IDM サーバーのアンインストール



注記

ドメインレベル **0** では、この手順は異なります。[「レプリカの削除」](#)を参照してください。

server.example.com のアンインストール手順:

1. 別のサーバーで **ipa server-del** コマンドを使用して、トポロジーから **server.example.com** を削除します。

```
[root@another_server ~]# ipa server-del server.example.com
```

2. **server.example.com** で **ipa-server-install --uninstall** コマンドを使用します。

```
[root@server ~]# ipa-server-install --uninstall
```

3. **server.example.com** を参照するすべてのネームサーバー (NS) の DNS レコードが DNS ゾーンから削除されていることを確認します。使用する DNS が IdM で管理される統合 DNS か、外部 DNS かに関わらず、確認してください。

2.5. サーバーの名前変更

設定後には、IdM サーバーのホスト名は変更できませんが、別の名前のレプリカでサーバーを置き換えることはできます。

1. 認証局、新たに必要なホスト名または IP アドレスを指定して、サーバーのレプリカを新たに作成します。これについては、[4章Identity Management のレプリカのインストールとアンインストール](#)に説明されています。
2. 最初の IdM サーバーインスタンスを停止します。

```
[root@old_server ~]# ipactl stop
```

3. 他のレプリカやクライアントは変わらず動作していることを確認します。
4. 「[IdM サーバーのアンインストール](#)」で説明されているように、最初の IdM サーバーをアンインストールします。

[1] 詳細は [DNS Amplification Attacks](#) ページを参照してください。

第3章 IDENTITY MANAGEMENT クライアントのインストールおよびアンインストール

本章では、サーバーに登録されているクライアントマシンとして Identity Management (IdM) ドメインに参加するようにシステムを設定する方法を説明します。



注記

IdM ドメインのクライアントおよびサーバーの詳細は「[Identity Management ドメイン](#)」を参照してください。

3.1. クライアントインストールの前提条件

DNS 要件

適切な DNS の委譲を使用すること。IdM の DNS 要件に関する詳細は「[ホスト名および DNS の設定](#)」を参照してください。

クライアント上の **resolv.conf** ファイルを変更しないこと

ポート要件

IdM クライアントは、複数のポートに接続してサーバーと通信します。これらのポートは、**IdM サーバー上で** 開放して機能できるようにしておく必要があります。IdM が必要とするポートについての情報は、「[ポート要件](#)」を参照してください。

クライアント上で、送信方向のポートを開放します。**firewalld** など、送信パケットをフィルタリングしないファイアウォールを使用している場合には、これらのポートはすでに送信方向で利用できる状態です。

連邦情報処理標準 (FIPS: Federal Information Processing Standard) のサポート

Red Hat Enterprise Linux 7.4 以降を使用して設定した環境の場合:

- FIPS モードを有効化したシステムで、新規の IdM サーバー、レプリカまたはクライアントを設定できます。インストールスクリプトでは、管理者の介入なしに、自動的に FIPS が有効化されているシステムを検出し、IdM を設定します。

オペレーティングシステムで FIPS を有効化するには、『セキュリティガイド』の「[FIPS モードの有効化](#)」を参照してください。



重要

以下の点に注意してください。

- FIPS モードを無効にしてインストールした既存の IdM サーバーで FIPS モードを有効にすることはできません。
- FIPS モードが無効になっている既存の IdM サーバーに FIPS サポートを有効にした新規レプリカをインストールすることはできません。

Red Hat Enterprise Linux 7.3 以前を使用して設定した環境の場合:

- IdM では、FIPS モードはサポートされません。システム上で FIPS を無効化してから、IdM サーバー、レプリカまたはクライアントをインストールし、インストール後も有効化しない

てください。

FIPS モードに関する詳しい情報は『セキュリティーガイド』の「[連邦情報処理標準 \(FIPS: Federal Information Processing Standard\)](#)」を参照してください。

Name Service Cache Daemon (NSCD) の要件

Red Hat では、Identity Management マシン上で NSCD を無効にすることを推奨しています。NSCD を無効にできない場合は、代わりに SSSD がキャッシュを行わないマッピングに対して NSCD を有効化するようにしてください。

NSCD と SSSD の両サービスはキャッシングを実行するので、これら両方をシステムが同時に使用すると問題が発生します。NSCD と SSSD の競合を避ける方法については、「[システムレベルの認証ガイド](#)」を参照してください。

3.2. クライアントのインストールに必要なパッケージ

ipa-client パッケージをインストールします。

```
# yum install ipa-client
```

ipa-client パッケージは、System Security Services Daemon (SSSD) パッケージなど、依存関係として他に必要なパッケージを自動的にインストールします。

クライアントマシンから IdM ドメインを管理するには、ipa-admintools パッケージもインストールします。このパッケージは、IdM 管理者向けにコマンドラインツールをインストールします。通常のクライアントシステムとしてクライアントマシンを使用する場合には、ipa-admintools は必要ありません。

3.3. クライアントのインストール

ipa-client-install ユーティリティーは、IdM クライアントをインストール、設定します。インストールプロセスでは、クライアントの登録に使用可能な認証情報を指定する必要があります。以下の認証方法がサポートされています。

admin などクライアントの登録を許可するユーザーの認証情報

デフォルトでは **ipa-client-install** はこのオプションを使用することが想定されています。例については「[クライアントの対話型インストール](#)」を参照してください。

ipa-client-install に直接ユーザーの認証情報をわたすには、**--principal** と **--password** のオプションを使用します。

サーバー上で無作為に事前に生成されるワンタイムパスワード

この認証方法を使用するには、**--random** オプションを **ipa-client-install** コマンドに追加します。詳細は[例3.1「無作為のパスワードを使用して非対話的にクライアントをインストールする手順」](#)を参照してください。

以前の登録からのプリンシパル

この認証方法を使用するには **--keytab** オプションを **ipa-client-install** に追加します。詳細は「[クライアントの IdM ドメインへの再登録](#)」を参照してください。

詳しい情報は、ipa-client-install(1) の man ページを参照してください。

以下のセクションでは、基本的なインストールのシナリオについてまとめています。**ipa-client-install** の使用や対応オプションの完全一覧については `ipa-client-install(1)` の `man` ページを参照してください。

3.3.1. クライアントの対話型インストール

以下の手順では、必要に応じてユーザーに入力を求めながらクライアントをインストールします。ユーザーは、**admin** など、ドメインへのクライアントの登録が許可されているユーザーの認証情報を指定します。

1. **ipa-client-install** ユーティリティを実行します。

以下のいずれかが当てはまる場合、**--enable-dns-updates** オプションを追加して、クライアントマシンの IP アドレスで DNS レコードを更新します。

- 。クライアントを登録する IdM サーバーが、統合 DNS とインストールされている場合。
- 。ネットワーク上の DNS サーバーが、GSS-TSIG プロトコルによる DNS エントリー更新を受け付ける場合。

--no-krb5-offline-passwords オプションを追加して、SSSD キャッシュに Kerberos パスワードを保存できないようにします。

2. このインストールスクリプトでは、必要な設定を自動的に取得するように試みます。

- a. DNS ゾーンおよび SRV レコードがシステム上で正しく設定されている場合には、スクリプトは自動的に必要な値がすべて検出され、出力されます。**yes** と入力して確定します。

```
Client hostname: client.example.com
Realm: EXAMPLE.COM
DNS Domain: example.com
IPA Server: server.example.com
BaseDN: dc=example,dc=com
```

```
Continue to configure the system with these values? [no]: yes
```

別の値でシステムをインストールするには現在のインストールをキャンセルし、**ipa-client-install** をもう一度実行して、コマンドラインオプションを使用して必要な値を指定します。

詳細は、`ipa-client-install(1)` の `man` ページの **DNS Autodiscovery** セクションを参照してください。

- b. スクリプトで自動的に設定が取得されなかった場合には、値を入力するようにプロンプトが表示されます。



重要

完全修飾ドメイン名は有効な DNS 名である必要があります。つまり、許可されるのは数字、アルファベット、ハイフン (-) のみです。ホスト名にアンダースコアのような他の文字があると、DNS エラーが発生します。また、ホスト名はすべて小文字を使用する必要があります、大文字は使用できません。

命名プラクティスに関する他の推奨事項については、[Red Hat Enterprise Linux セキュリティガイド](#) を参照してください。

3. このスクリプトは、クライアントの登録に使用するユーザー ID の入力を求めるプロンプトを表示します。デフォルトでは、このユーザーは **admin** です。

```
User authorized to enroll computers: admin
Password for admin@EXAMPLE.COM
```

4. インストールスクリプトでクライアントが設定されます。動作が完了するまで待機します。

```
Client configuration complete.
```

5. **ipa-client-automount** ユーティリティを実行します。これで NFS が IdM 向けに自動的に設定されます。詳細は、「[NFS の自動設定](#)」を参照してください。

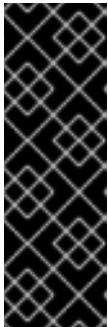
3.3.2. 非対話式なクライアントのインストール

非対話式のインストールの場合には、コマンドラインオプションを使用して **ipa-client-install** ユーティリティに必要な情報すべて渡します。非対話式のインストールで最小限必要となるオプションは以下のとおりです。

- クライアントの登録に使用する認証情報を指定するオプション。詳細は「[クライアントのインストール](#)」を参照してください。
- **--unattended**: ユーザーの確認なしにインストールを実行します。

DNS ゾーンおよび SRV レコードがシステム上で正しく設定されている場合には、スクリプトは自動的に必要な値をすべて検出します。スクリプトが自動的に値を検出できない場合には、コマンドラインオプションを使用して指定してください。

- **--hostname**: クライアントマシンの静的ホスト名を指定します。



重要

完全修飾ドメイン名は有効な DNS 名である必要があります。つまり、許可されるのは数字、アルファベット、ハイフン (-) のみです。ホスト名にアンダースコアのような他の文字があると、DNS エラーが発生します。また、ホスト名はすべて小文字を使用する必要があり、大文字は使用できません。

命名プラクティスに関する他の推奨事項については、[Red Hat Enterprise Linux セキュリティガイド](#) を参照してください。

- **--server**: クライアントの登録先の IdM サーバーのホスト名を指定します。
- **--domain**: クライアントの登録先の IdM サーバーの DNS ドメイン名を指定します。
- **--realm**: Kerberos レalm名を指定します。

以下のいずれかが当てはまる場合、**--enable-dns-updates** オプションを追加して、クライアントマシンの IP アドレスで DNS レコードを更新します。

- クライアントを登録する IdM サーバーが、統合 DNS とインストールされている場合。
- ネットワーク上の DNS サーバーが、GSS-TSIG プロトコルによる DNS エントリー更新を受け付ける場合。

--no-krb5-offline-passwords オプションを追加して、SSSD キャッシュに Kerberos パスワー

ドを保存できないようにします。

ipa-client-install に対応のオプションに関する完全一覧は、ipa-client-install(1) の man ページを参照してください。

例3.1 無作為のパスワードを使用して非対話式にクライアントをインストールする手順

以下の手順では、ユーザーに入力を促さずにクライアントをインストールします。このプロセスでは、サーバーで無作為に生成されたワンタイムパスワードが含まれており、このパスワードを登録の認証に使用します。

1. 既存のサーバー上で:

- a. 管理者としてログインします。

```
$ kinit admin
```

- b. IdM ホストとして新規マシンを追加します。--random オプションを指定して **ipa host-add** コマンドを使用して、無作為のパスワードを生成します。

```
$ ipa host-add client.example.com --random
```

```
-----  
Added host "client.example.com"
```

```
-----  
Host name: client.example.com  
Random password: W5YpARl=7M.n  
Password: True  
Keytab: False  
Managed by: server.example.com
```

生成パスワードは、IdM ドメインへのマシン登録に使用した後は無効になります。登録が完了すると正しいホストの keytab に置き換えられます。

2. クライアントをインストールするマシンで、**ipa-client-install** を実行します。以下のオプションを使用します。

- **--password:** **ipa host-add** 出力からの無索引パスワードを使用します。



注記

パスワードには特殊文字が含まれることが多いので、一重引用符 (') で括弧するようにしてください。

- **--unattended:** ユーザーの確認なしにインストールを実行します。

DNS ゾーンおよび SRV レコードがシステム上で正しく設定されている場合には、スクリプトは自動的に必要な値をすべて検出します。スクリプトが自動的に値を検出できない場合には、コマンドラインオプションを使用して指定してください。

例を示します。

```
# ipa-client-install --password 'W5YpARl=7M.n' --domain  
example.com --server server.example.com --unattended
```

3. **ipa-client-automount** ユーティリティーを実行します。これで NFS が IdM 向けに自動的に設定されます。詳細は、「[NFS の自動設定](#)」を参照してください。

3.4. キックスタートを使用した IDM クライアントの設定

キックスタートで登録すると、Red Hat Enterprise Linux がインストールされた時点で IdM ドメインに新規システムが自動的に追加されます。キックスタートの詳細は、『インストールガイド』の「[キックスタートのインストール](#)」を参照してください。

キックスタートでのクライアントのインストールを準備する際には、以下の手順が含まれます。

1. 「[IdM サーバーにおけるクライアントのホストエントリーの事前作成](#)」
2. 「[クライアント向けのキックスタートファイルの作成](#)」

3.4.1. IdM サーバーにおけるクライアントのホストエントリーの事前作成

1. 管理者としてログインします。

```
$ kinit admin
```

2. IdM サーバー上でホストエントリーを作成し、エントリーの一時パスワードを設定します。

```
$ ipa host-add client.example.com --password=secret
```

キックスタートがこのパスワードを使用して、クライアントのインストール時に認証し、最初の認証試行後に無効にします。クライアントが正常にインストールされたら、keytab を使用して認証が行われます。

3.4.2. クライアント向けのキックスタートファイルの作成

IdM クライアントの設定に使用するキックスタートファイルには、以下を追加する必要があります。

- インストールするパッケージ一覧に含まれる ipa-client パッケージ

```
%packages
@ X Window System
@ Desktop
@ Sound and Video
ipa-client
...
```

詳細は、『インストールガイド』の「[パッケージの選択](#)」を参照してください。

- インストール後の手順
 - 登録後に SSH 鍵が生成されていることを確認する
 - 以下を指定して **ipa-client-install** ユーティリティーを実行する
 - IdM ドメインサービスへのアクセスおよび設定に必要な全情報

- 「[IdM サーバーにおけるクライアントのホストエントリーの事前作成](#)」の記載どおりに IdM サーバーにクライアントホストを事前に作成する際に設定するパスワード

例を示します。

```
%post --log=/root/ks-post.log

# Generate SSH keys to ensure that ipa-client-install uploads
them to the IdM server
/usr/sbin/sshd-keygen

# Run the client install script
/usr/sbin/ipa-client-install --hostname=client.example.com --
domain=EXAMPLE.COM --enable-dns-updates --mkhomedir -w secret --
realm=EXAMPLE.COM --server=server.example.com
```

非対話式のインストールの場合には、**--unattended** オプションも追加します。

クライアントのインストールスクリプトがマシンの証明書を要求できるようにするには、以下を行います。

- **--request-cert** オプションを **ipa-client-install** に追加します。
- キックスタートの **chroot** 環境で、**getcrt** と **ipa-client-install** ユーティリティ両方に対して **/dev/null** にシステムバスのアドレスを設定します。これには、インストール後の指示ファイルの **ipa-client-install** 指示文の前に以下の行を追加します。

```
# env DBUS_SYSTEM_BUS_ADDRESS=unix:path=/dev/null getcert list
# env DBUS_SYSTEM_BUS_ADDRESS=unix:path=/dev/null ipa-client-
install
```



注記

Red Hat は、キックスタートの登録前に **sshd** サービスを起動することは推奨していません。登録前に **sshd** を開始すると、クライアントは自動的に SSH 鍵を生成するので、上記のスクリプトの使用が推奨されます。

詳細は、『インストールガイド』の「[インストール後のスクリプト](#)」を参照してください。

キックスタートの詳細は、『インストールガイド』の「[キックスタートを使ったインストールの実行方法](#)」を参照してください。キックスタートのファイルの例については、「[キックスタート設定の例](#)」を参照してください。

3.5. クライアントのインストール後の検討事項

3.5.1. Identity Management の事前設定の削除

ipa-client-install スクリプトは、**/etc/openldap/ldap.conf** および **/etc/sss/sss.conf** ファイルから、以前の LDAP および SSSD 設定を削除します。これらのファイルの設定を変更すると、スクリプトにより新規クライアントの値が追加されますが、コメントアウトされます。以下に例を示します。

```
BASE    dc=example,dc=com
URI      ldap://ldap.example.com

#URI ldaps://server.example.com # modified by IPA
#BASE dc=ipa,dc=example,dc=com # modified by IPA
```

新しい Identity Management 設定値を適用します。

1. `/etc/openldap/ldap.conf` および `/etc/sss/sss.conf` を開きます。
2. 以前の設定を削除します。
3. 新規の Identity Management 設定をアンコメントします。
4. システム全体の LDAP 設定に依存するサーバープロセスでは、変更を適用するのに再起動が必要な場合があります。**openldap** ライブラリーを使用するアプリケーションでは通常、起動時に設定がインポートされます。

3.6. 新規クライアントのテスト

クライアントはサーバーが定義したユーザーに関する情報を取得できることを確認します。たとえば、デフォルトの **admin** ユーザーをチェックするには、以下を実行します。

```
[user@client ~]$ id admin
uid=1254400000(admin) gid=1254400000(admins) groups=1254400000(admins)
```

3.7. クライアントのアンインストール

クライアントをアンインストールすると、IdM ドメインからクライアントと、SSSD などのシステムサービスの IdM 固有の設定すべてが削除されます。これにより、クライアントマシンの以前の設定が復元されます。

1. **ipa-client-install --uninstall** コマンドを実行します。

```
# ipa-client-install --uninstall
```

2. サーバーからクライアントホストの DNS エントリーを手動で削除します。[「DNS ゾーンからレコードを削除する」](#) を参照してください。

3.8. クライアントの IDM ドメインへの再登録

クライアントの仮想マシンが破棄され、その keytab がまだある場合には、クライアントを再登録することができます。

- 対話式の場合には管理者の認証情報を使用します。[「管理者のアカウントを使用して対話式にクライアントを再登録する方法」](#) を参照してください。
- 非対話式の場合は、以前にバックアップした keytab ファイルを使用します。[「クライアントの keytab を使用して非対話式にクライアントを再登録する方法」](#) を参照してください。



注記

ドメインエントリーがアクティブな状態のクライアントは再登録しかできません。クライアントをアンインストールした場合 (**ipa-client-install --uninstall** の使用) またはホストのエントリーを無効にした場合は (**ipa host-disable** の使用) 再登録できません。

再登録時には IdM は以下を実行します。

- 元のホスト証明書を破棄します。
- 新規ホストの証明書を生成します。
- 新規の SSH 鍵を作成します。
- 新規の keytab を生成します。

3.8.1. 管理者のアカウントを使用して対話式にクライアントを再登録する方法

1. 同じホスト名のクライアントマシンを再作成します。
2. クライアントマシンで **ipa-client-install --force-join** コマンドを実行します。

```
# ipa-client-install --force-join
```

3. このスクリプトは、クライアントの登録に使用するユーザー ID の入力を求めるプロンプトを表示します。デフォルトでは、このユーザーは **admin** です。

```
User authorized to enroll computers: admin
Password for admin@EXAMPLE.COM
```

3.8.2. クライアントの **keytab** を使用して非対話式にクライアントを再登録する方法

自動インストールや、その他、管理者パスワードの利用ができない状況では、クライアントの keytab を使用して再登録するのが適切です。

1. 元のクライアントの keytab ファイルを **/tmp** または **/root** ディレクトリーなどにバックアップします。
2. 同じホスト名のクライアントマシンを再作成します。
3. クライアントを再登録して、**--keytab** オプションを使用して keytab の場所を指定します。

```
# ipa-client-install --keytab /tmp/krb5.keytab
```



注記

登録を開始するために認証する場合には、**--keytab** オプションで指定した keytab のみが使用されます。再登録時には IdM はクライアントの新規 keytab を生成します。

3.9. クライアントマシンの名前変更

以下のセクションでは、IdM クライアントの名前の変更方法を説明します。使用するプロセスは以下のとおりです。

- 「現在のサービスや keytab 設定の特定」
- 「IdM ドメインからのクライアントマシンの削除」
- 「新しいホスト名が指定されたクライアントの再登録」



警告

クライアントの名前は手動で変更します。Red Hat は、ホスト名の変更が絶対に必要な場合以外は推奨していません。

現在のサービスや keytab 設定の特定

現在のクライアントをアンインストールする前に、クライアントの特定の設定をメモします。新しいホスト名のマシンを再登録した後にこの設定を適用します

1. マシンで実行されているサービスを特定します。
 - a. **ipa service-find** コマンドを使用して、証明書のあるサービスを特定して出力します。

```
$ ipa service-find client.example.com
```

- b. さらに、各ホストには、**ipa service-find** の出力に表示されないデフォルトの **ホストサービス** があります。ホストサービスのサービスプリンシパル (ホストプリンシパルとも呼ばれる) は、**host/client.example.com** です。

2. マシンが所属するすべてのホストグループを特定します。

```
# ipa hostgroup-find client.example.com
```

3. **ipa service-find client.example.com** で表示されるサービスプリンシパルすべてについて、**client.example.com** の適切な keytab の場所を特定します。

クライアントシステム上の各サービスには、**ldap/client.example.com@EXAMPLE.COM** といったように **service_name/hostname@REALM** の形式で Kerberos プリンシパルがあります。

IdM ドメインからのクライアントマシンの削除

1. IdM ドメインからクライアントマシンの登録を解除します。「[クライアントのアンインストール](#)」を参照してください。
2. **/etc/krb5.keytab** 以外の特定された各 keytab で、古いプリンシパルを削除します。

```
[root@client ~]# ipa-rmkeytab -k /path/to/keytab -r EXAMPLE.COM
```

[「Keytab の削除」](#)を参照してください。

3. IdM サーバー上でホストエントリを削除します。これで全サービスが削除され、そのホスト向けに発行されたすべての証明書が破棄されます。

```
[root@server ~]# ipa host-del client.example.com
```

この時点で、ホストは IdM から完全に削除されました。

新しいホスト名が指定されたクライアントの再登録

1. 必要に応じてマシンの名前を変更します。
2. IdM クライアントとしてマシンを再登録します。[「クライアントの IdM ドメインへの再登録」](#)を参照してください。
3. IdM サーバーで、[「現在のサービスや keytab 設定の特定」](#)で特定した各サービスの新規 keytab を追加します。

```
[root@server ~]# ipa service-add service_name/new_host_name
```

4. [「現在のサービスや keytab 設定の特定」](#)で割り当てた証明書のあるサービスに対して証明書を生成します。方法は以下のとおりです。
 - IdM 管理者ツールを使用すること。[24章ユーザー、ホスト、およびサービス向け証明書の管理](#)を参照してください。
 - **certmonger** ユーティリティーを使用すること。『システムレベルの認証ガイド』[「CERTMONGER を使った作業」](#) または certmonger(8) の man ページを参照してください。
5. [「現在のサービスや keytab 設定の特定」](#)で特定したホストグループにクライアントを再度追加します。[「ユーザーまたはホストグループメンバーの追加および削除」](#)を参照してください。

第4章 IDENTITY MANAGEMENT のレプリカのインストールとアンインストール

レプリカは、既存の Identity Management サーバー設定をクローンして作成します。そのため、サーバーとそのレプリカは全く同じコア設定を共有します。レプリカのインストールプロセスは既存のサーバー設定をコピーして、その設定をベースにレプリカをインストールします。

"[Backup and Restore in IdM/IPA](#)" Knowledgebase solution で説明されているように、バックアップソリューションとして複数のサーバーレプリカを維持してデータの損失を回避することが推奨されます。

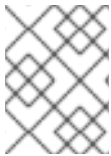


注記

9章 [Identity Management のバックアップと復元](#) に記載されているように、別のバックアップソリューションとして **ipa-backup** ユーティリティがあります。これは、レプリカから IdM デプロイメントを再構築できない場合に主に推奨されるソリューションです。

4.1. IDM レプリカの説明

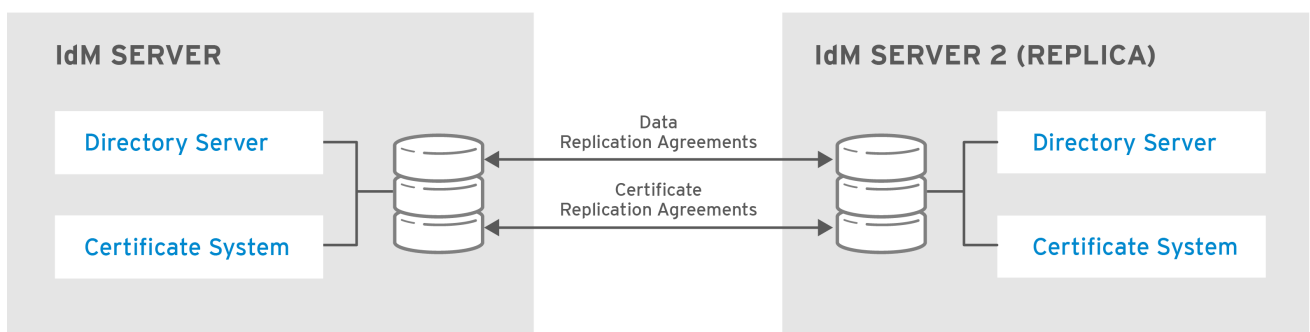
レプリカは、最初のマスターサーバーのクローンとして作成されます。レプリカが作成されると、レプリカにはマスターサーバーと全く同じ機能が搭載されます。サーバーと、このサーバーから作成されたレプリカは、ユーザー、マシン、証明書、設定済みのポリシーの同じ内部情報を共有します。



注記

IdM トポロジー内のマシンタイプに関する情報は「[Identity Management ドメイン](#)」を参照してください。

レプリケーションとは、レプリカとレプリカの間でデータをコピーするプロセスのことです。レプリカ間の情報は、マルチマスターレプリケーションを使用して共有されます。レプリカ合意で参加したレプリカはすべて、更新を受信するため、データのマスターと考えられます。



RHEL_404973_0516

図4.1 サーバーとレプリカの合意

4.2. レプリカに関するデプロイメントの考慮事項

4.2.1. トポロジーにおけるサーバーサービスのディストリビューション

IdM サーバーは、証明局 (CA) または DNS など複数のサーバーを実行できます。レプリカは、レプリカをベースとして作成したサーバーとして同じサービスを実行できますが、必須ではありません。

たとえば、最初のサーバーが DNS を稼働している場合でも、DNS サービスなしでレプリカをインストールすることができます。同様に、最初のサーバーが DNS なしにインストールされている場合でも、DNS サーバーとしてレプリカを設定することができます。

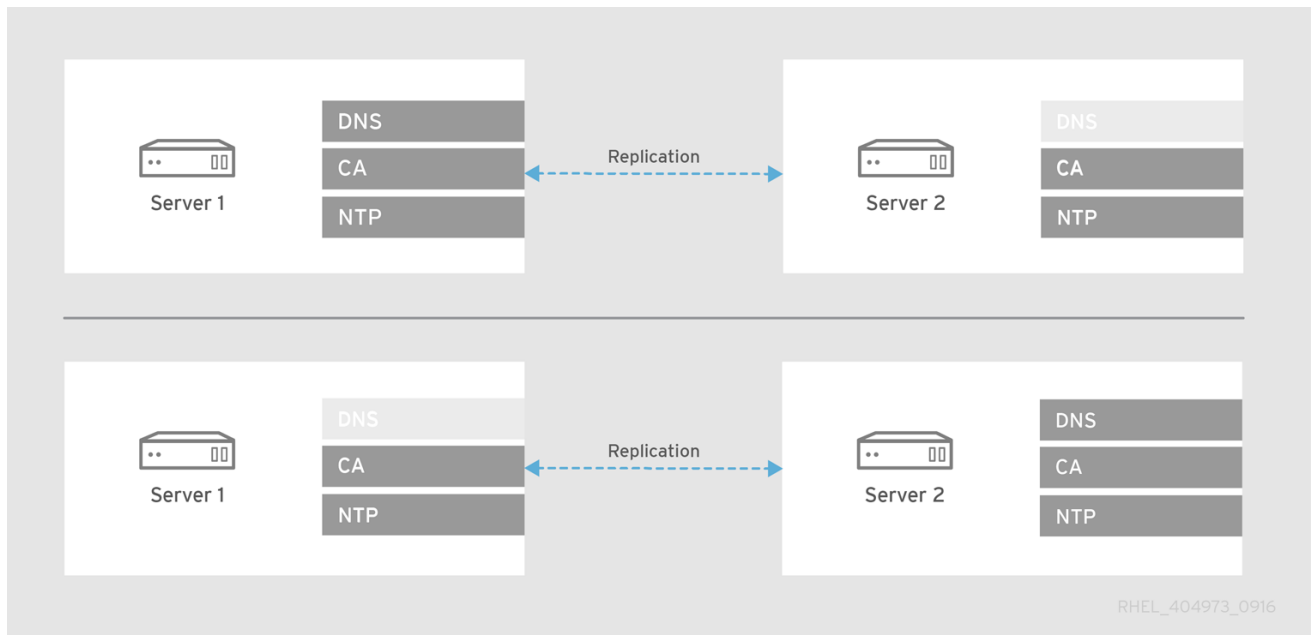


図4.2 異なるサービスが含まれるレプリカ

レプリカ上の CA サービス

CA なしにレプリカを設定する場合は、証明書の操作の要求はすべて、トポロジー内の CA サーバーに転送されます。



警告

Red Hat では、複数のサーバーに CA サービスをインストールしておくことを強く推奨しています。CA サービスを含む最初のサーバーのレプリカをインストールする方法についての情報は、「[CA を設定したレプリカのインストール](#)」を参照してください。

CA が 1 つのサーバーにしかインストールされていないと、CA サーバーが故障した際に CA 設定が失われて回復できない恐れがあります。詳細については、「[失われた CA サーバーの復旧](#)」を参照してください。

レプリカで CA を設定する場合は、設定は最初のサーバーの CA 設定をミラーリングする必要があります。

- たとえば、統合された IdM CA がルート証明局としてサーバーに含まれる場合には、レプリカはこの統合された CA をルート証明局としてインストールする必要があります。
- サポートされる CA 設定オプションは「[CA 設定の決定](#)」を参照してください。

4.2.2. レプリカトポロジーの推奨事項

Red Hat は以下のガイドラインに準拠することを推奨します。

単一の IdM ドメインでレプリカ 61 個以上設定しない

Red Hat は、レプリカが最大 60 個含まれる環境のサポートを保証します。

レプリカごとに 最低で 2 つ、最大 4 つ のレプリカ合意を設定する

追加のレプリカ合意を設定すると、最初のレプリカとマスターサーバーの間だけでなく、他のレプリカとの間でも情報が複製されます。

- サーバー A からレプリカ B を作成してから、サーバー A からレプリカ C を作成する場合には、レプリカ B と C は直接連携されていないので、レプリカ B からのデータを先にサーバー A に複製してからレプリカ C に伝搬する必要があります。

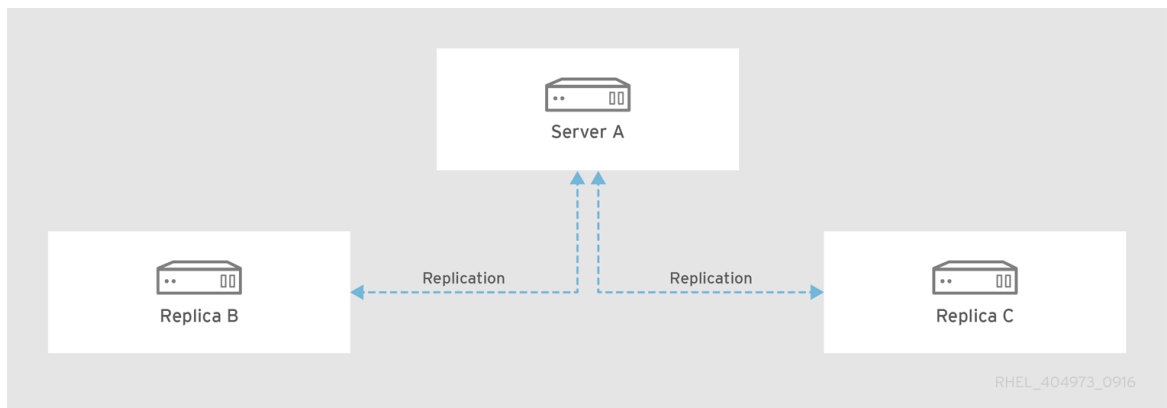


図4.3 レプリカ B と C はレプリカ合意で連携されていない

レプリカ B とレプリカ C の間で追加のレプリカ合意を設定すると、データは直接複製され、データの可用性、一貫性、フェールオーバーの耐性、パフォーマンスを向上します。

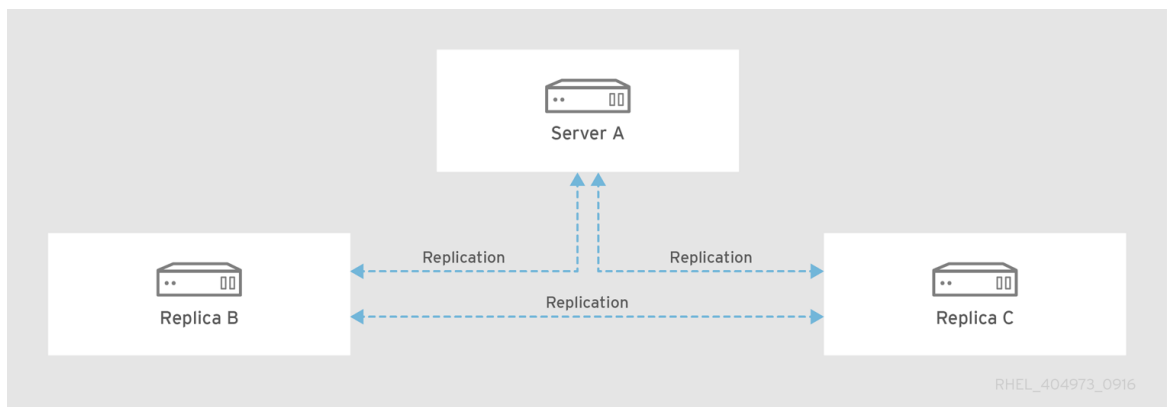
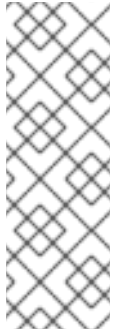


図4.4 レプリカ B および C がレプリカ合意で連携されている

レプリカ合意の管理に関する詳細は[6章 レプリケーショントポロジーの管理](#)を参照してください。

レプリカごとにレプリカ合意を 5 つ以上設定する必要はありません。サーバーに設定されるレプリカ合意が増えても、追加で大幅な利点がもたらされるわけではありません。これは、1 台のマスターが一度に更新できるのは、消費者サーバー 1 台のみで、他の合意はその間アイドルかつ待機状態となっているからです。また、レプリカ合意を多く設定しすぎると、全体的なパフォーマンスに悪影響を与える可能性があります。



注記

ipa topologysuffix-verify コマンドは、トポロジーが最も重要な推奨事項に対応しているかどうかをチェックします。詳細は、**ipa topologysuffix-verify --help** を実行してください。

このコマンドでは、トポロジーのサフィックスを指定する必要があります。詳細は「[レプリカ合意](#)、[トポロジーサフィックス](#)、および[トポロジーセグメント](#)」を参照してください。

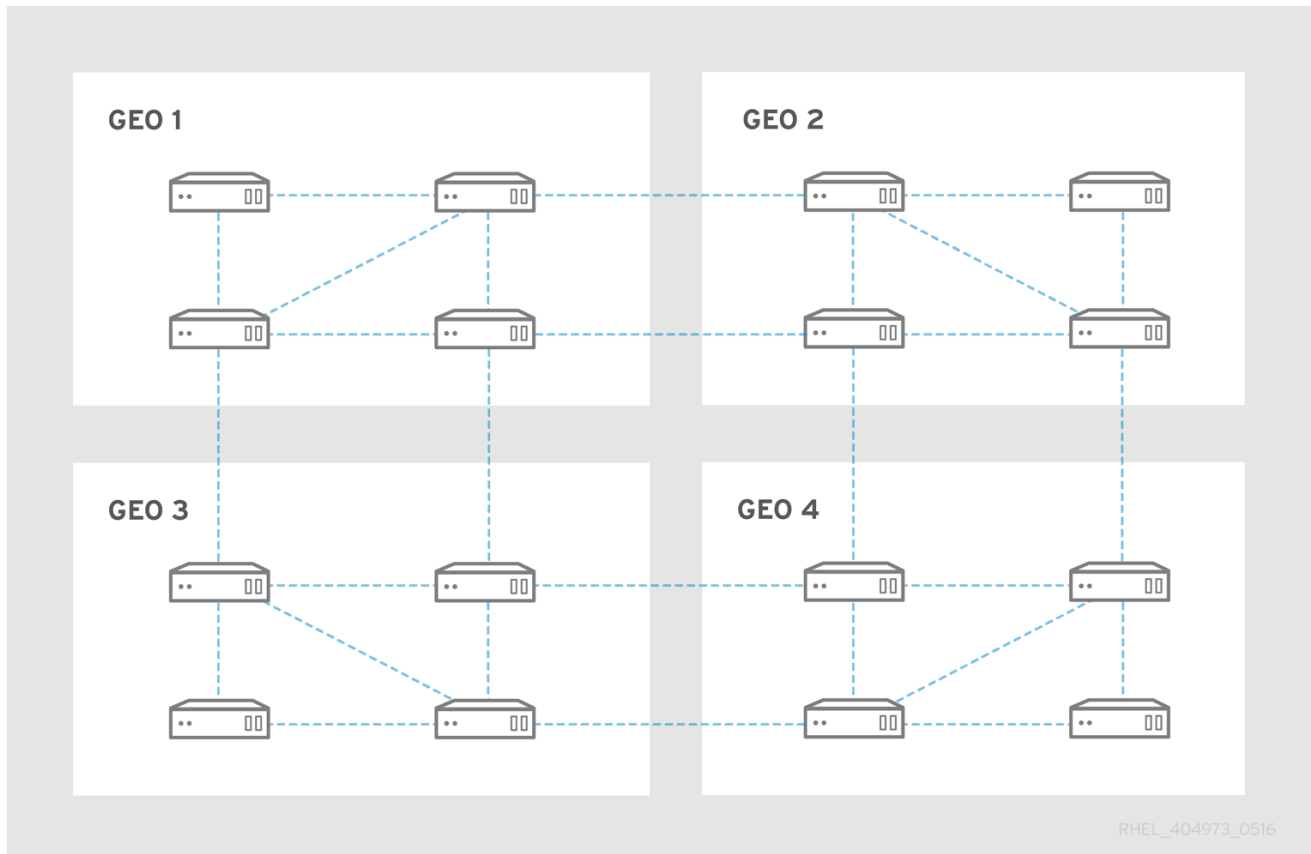


図4.5 トポロジーの例

4.2.2.1. タイトセルトポロジー

最も回復力のあるトポロジーを設定するには、セルに含めるサーバー数を少数にしてサーバーとレプリカのセル構成を作成します。

- 各セルは **タイトセル** になります。タイトセルでは、全サーバーに相互のレプリカ合意が設定されています。
- 各サーバーは、セルの **外部** にある別のサーバーとのレプリカ合意があります。これにより、セルはすべてドメイン内の他の全セルと疎結合されるようになります。

タイトセルトポロジーを設定するには以下を行います。

- 各メインオフィス、データセンター、地域に、少なくとも 1 つの IdM サーバーを用意します。可能であれば、2 台の IdM サーバーを用意します。
- 各データセンターに用意するサーバーは 4 台までとします。

- 小規模なオフィスでは、レプリカを使用するのではなく、SSSD を使用して認証情報をキャッシュして、データのバックエンドとしてオフサイトの IdM サーバーを使用します。

4.3. レプリカのインストールの前提条件

レプリカのインストール要件は、IdM サーバーのものと同じです。レプリカのマシンが「[サーバーインストールの前提条件](#)」に記載の前提条件すべてを満たすようにしてください。

一般的なサーバー要件に加え、以下の条件を満たす必要があります。

レプリカは、IdM と同じか、最新バージョンを実行している必要があります。

たとえば、マスターサーバーは Red Hat Enterprise Linux 7 で実行されており、IdM 4.4 パッケージを使用する場合には、レプリカは Red Hat Enterprise Linux 7 以降のバージョンで実行し、IdM のバージョン 4.4 以降を使用する必要があります。これにより、サーバーからレプリカに設定が正しくコピーされるようになります。



重要

IdM は、マスターのバージョンより前のバージョンのレプリカ作成をサポートしません。以前のバージョンを使ってレプリカを作成しようとすると、インストールが失敗します。

レプリカには、追加でポートを解放しておく必要があります。

「[ポート要件](#)」の説明にある標準の IdM サーバーポート要件に加え、以下も満たす必要があります。

- ドメインレベル 0 では、レプリカ設定プロセス中は **TCP ポート 22** を解放した状態にしておいてください。このポートは、マスターサーバーへの接続に SSH を使用するために必要です。



注記

ドメインレベルの詳細は [7章 ドメインレベルの表示と引き上げ](#) を参照してください。

- サーバーの中の 1 台において Red Hat Enterprise Linux 6 を実行し、CA をインストールしている場合には、レプリカの設定時とその後は **TCP ポート 7389** も解放するようにしてください。Red Hat Enterprise Linux 7 環境では、ポート 7389 は必要ありません。

firewall-cmd ユーティリティを使用してポートを解放する方法は「[ポート要件](#)」を参照してください。

4.4. レプリカのインストールに必要なパッケージ

レプリカパッケージの要件は、サーバーパッケージの要件と同じです。「[IdM サーバーのインストールに必要なパッケージ](#)」を参照してください。

4.5. レプリカの作成: 概要

ipa-replica-install ユーティリティを使用して、既存の IdM サーバーから新しいレプリカをインストールします。



注記

本章は Red Hat Enterprise Linux 7.3 で導入された簡単なレプリカインストールの方法について説明します。この手順ではドメインレベル 1 ([7章 ドメインレベルの表示と引き上げ](#)を参照) が必要です。

ドメインレベル 0 でのレプリカのインストールに関するドキュメントは[付録D ドメインレベル 0 でのレプリカの管理](#)を参照してください。

以下に、新規レプリカをインストールできます。

- 既存の IdM クライアント。クライアントをレプリカに プロモート してください (「[既存のクライアントからレプリカへのプロモート](#)」参照)。
- IdM ドメインに登録されていないマシン (「[クライアントでないマシンへのレプリカのインストール](#)」参照)。

いずれの場合でも **ipa-replica-install** にオプションを追加することでレプリカをカスタマイズできます (「[ユースケースに適したレプリカの設定の際に ipa-replica-install を使用する方法](#)」参照)。



重要

複製する IdM サーバーに Active Directory とのトラストがある場合は、**ipa-replica-install** を実行してからトラストエージェントとしてレプリカを設定します。『Windows 統合ガイド』の「[信頼コントローラーおよび信頼エージェント](#)」を参照してください。

既存のクライアントからレプリカへのプロモート

既存のクライアントにレプリカをインストールするには、クライアントのプロモートが認証されていることを確認する必要があります。これには、以下のいずれかを選択します。

特権ユーザーの認証情報を指定

デフォルトの特権ユーザーは **admin** です。ユーザーの認証情報を提示する方法は複数あります。

- IdM がプロンプトで対話的に認証情報を求める方法



注記

これは特権ユーザーの認証情報を指定するデフォルトの方法です。**ipa-replica-install** の実行時に認証情報が提示されていない場合には、インストールの際に自動的にプロンプトが表示されます。

- クライアントで **ipa-replica-install** を実行する前にユーザーとしてログインします。

```
$ kinit admin
```

- ユーザーのプリンシパル名とパスワードを **ipa-replica-install** に直接追加する方法

```
# ipa-replica-install --principal admin --admin-password  
admin_password
```

ipaservers ホストグループへのクライアントの追加

ipaservers のメンバーは、特権ユーザーの認証情報とよく似た特権にマシンを昇格します。ユーザーの認証情報を指定する必要はありません。

例: 「ホストの **keytab** を使用してレプリカにクライアントをプロモートする方法」

クライアントでないマシンへのレプリカのインストール

IdM ドメインに登録されていないマシン上で実行する場合は **ipa-replica-install** では最初にクライアントとしてマシンを登録してから、レプリカのコンポーネントをインストールします。

この状況でレプリカをインストールするには、以下のいずれかを実行します。

特権ユーザーの認証情報を指定

デフォルトの特権ユーザーは **admin** です。認証情報を指定するには、プリンシパル名とパスワードを **ipa-replica-install** に直接追加します。

```
# ipa-replica-install --principal admin --admin-password admin_password
```

クライアントの任意パスワードの指定

レプリカをインストールする前にサーバーで無作為のパスワードを生成する必要があります。インストール時にはユーザーの認証情報を指定する必要はありません。

例: 「無作為のパスワードを使用したレプリカのインストール」

デフォルトでは、クライアントのインストーラーが最初に検出した IdM サーバーにレプリカはインストールされます。特定のサーバーにレプリカをインストールするには、**ipa-replica-install** に以下のオプションを追加します。

- **--server:** サーバーの完全修飾ドメイン名 (FQDN)
- **--domain:** IdM DNS ドメイン

ユースケースに適したレプリカの設定の際に **ipa-replica-install** を使用方法

オプションなしで **ipa-replica-install** を実行すると、基本的なサーバーサービスのみが設定されます。DNS および証明局 (CA) などの追加のサービスをインストールするには **ipa-replica-install** にオプションを追加します。



警告

Red Hat では、複数のサーバーに CA サービスをインストールしておくことを強く推奨しています。CA サービスを含む最初のサーバーのレプリカをインストールする方法についての情報は、「[CA を設定したレプリカのインストール](#)」を参照してください。

CA が 1 つのサーバーにしかインストールされていないと、CA サーバーが故障した際に CA 設定が失われて回復できない恐れがあります。詳細については、「[失われた CA サーバーの復旧](#)」を参照してください。

主要なオプションを使用してレプリカをインストールするシナリオ例については、以下を参照してください。

- 「DNS ありのレプリカのインストール」: **--setup-dns** と **--forwarder** の使用
- 「CA を設定したレプリカのインストール」: **--setup-ca** の使用
- 「CA がインストールされていないサーバーからのレプリカのインストール」: **--dirsrv-cert-file**、**--dirsrv-pin**、**--http-cert-file** および **--http-pin** の使用

レプリカの設定に使用するオプションの完全な一覧については `ipa-replica-install(1)` の `man` ページを参照してください。

4.5.1. ホストの **keytab** を使用してレプリカにクライアントをプロモートする方法

この手順では、プロモートを許可する独自のホスト **keytab** を使用して、既存の IdM クライアントをレプリカにプロモートします。

以下の手順では、管理者またはディレクトリーマネージャー (DM) の認証情報を指定する必要はありません。そのため、機密情報がコマンドラインに公開されないのが、よりセキュリティが高くなります。

1. 既存のサーバー上で:

- a. 管理者としてログインします。

```
$ kinit admin
```

- b. クライアントマシンを **ipaservers** のホストグループに追加します。

```
$ ipa hostgroup-add-member ipaservers --hosts client.example.com
Host-group: ipaservers
Description: IPA server hosts
Member hosts: server.example.com, client.example.com
-----
Number of members added 1
-----
```

ipaservers のメンバーは、管理者の認証情報とよく似た特権にマシンを昇格します。

2. クライアント上で **ipa-replica-install** ユーティリティを実行します。

```
# ipa-replica-install
```

4.5.2. 無作為のパスワードを使用したレプリカのインストール

この手順では、レプリカは IdM クライアントとしてまだ設定されていないマシンを最初からインストールします。登録を認証するには、クライアントの登録 1 回だけ有効な、クライアント固有の無作為パスワードを使用します。

以下の手順では、管理者またはディレクトリーマネージャー (DM) の認証情報を指定する必要はありません。そのため、機密情報がコマンドラインに公開されないのが、よりセキュリティが高くなります。

1. 既存のサーバー上で:

- a. 管理者としてログインします。

```
$ kinit admin
```

- b. IdM ホストとして新規マシンを追加します。--**random** オプションを指定して **ipa host-add** コマンドを使用して、レプリカのインストールに使用する無作為のワンタイムパスワードを生成します。

```
$ ipa host-add client.example.com --random
```

```
-----  
Added host "client.example.com"
```

```
-----  
Host name: client.example.com  
Random password: W5YpARl=7M.n  
Password: True  
Keytab: False  
Managed by: server.example.com
```

生成パスワードは、IdM ドメインへのマシン登録に使用した後は無効になります。登録が完了すると正しいホストの keytab に置き換えられます。

- c. マシンを **ipaservers** のホストグループに追加します。

```
$ ipa hostgroup-add-member ipaservers --hosts client.example.com  
Host-group: ipaservers  
Description: IPA server hosts  
Member hosts: server.example.com, client.example.com
```

```
-----  
Number of members added 1  
-----
```

ipaservers のメンバーは、必須サーバーサービスの設定に必要な特権にマシンを昇格します。

2. レプリカのインストール先のマシンで **ipa-replica-install** を実行して、--**password** オプションを使用して無作為パスワードを指定します。特殊文字が含まれることが多いので、一重引用符 (') でパスワードを括弧のようにしてください。

```
# ipa-replica-install --password 'W5YpARl=7M.n'
```

4.5.3. DNS ありのレプリカのインストール

以下の手順は、まだ IdM ドメインに含まれないマシンやクライアントにレプリカをインストールする際に使用します。詳細は「[レプリカの作成: 概要](#)」を参照してください。

- 以下のオプションを指定して **ipa-replica-install** を実行します。
 - setup-dns** は、存在しない場合には DNS ゾーンを作成して、DNS サーバーとしてレプリカを設定します。
 - forwarder** はフォワーダーを指定します。フォワーダーを使用しない場合には --**no-forwarder** を指定します。

フェールオーバーに対応するために複数のフォワーダーを指定するには **--forwarder** を複数回使用します。

例を示します。

```
# ipa-replica-install --setup-dns --forwarder 192.0.2.1
```



注記

ipa-replica-install ユーティリティーは、**--no-reverse** または **--no-host-dns** など、DNS 設定関連する、その他の複数のオプションに対応します。詳しい情報は `ipa-replica-install(1)` の man ページを参照してください。

2. DNS を有効にして最初のサーバーを作成した場合には、適切な DNS エントリーを使用して自動的にレプリカが作成されます。これらのエントリーにより、IdM クライアントは新規サーバーを検出できるようになります。

DNS を有効にせずに最初のサーバーを作成した場合には、手動で DNS レコードを追加します。以下の DNS レコードがドメインサービスに必要です。

- **_ldap._tcp**
- **_kerberos._tcp**
- **_kerberos._udp**
- **_kerberos-master._tcp**
- **_kerberos-master._udp**
- **_ntp._udp**
- **_kpasswd._tcp**
- **_kpasswd._udp**

以下の例では、エントリーの存在を確認する方法を説明します。

- a. DOMAIN および NAMESERVER 変数に適切な値を設定します。

```
# DOMAIN=example.com
# NAMESERVER=replica
```

- b. 以下のコマンドを使用して、DNS エントリーの有無を確認します。

```
# for i in _ldap._tcp _kerberos._tcp _kerberos._udp _kerberos-
master._tcp _kerberos-master._udp _ntp._udp ; do
dig @${NAMESERVER} ${i}.${DOMAIN} srv +nocmd +noquestion
+nocomments +nostats +noaa +noadditional +noauthority
done | egrep "^_"

_ldap._tcp.example.com. 86400      IN      SRV      0 100 389
server1.example.com.
_ldap._tcp.example.com. 86400      IN      SRV      0 100 389
server2.example.com.
```

```
_kerberos._tcp.example.com. 86400 IN      SRV      0 100 88
server1.example.com.
...
```

3. **オプションであるが推奨:** レプリカが利用できないときのために、バックアップサーバーとして他の DNS サーバーを手動で追加してください。「[ネームサーバーの追加設定](#)」を参照してください。これは、新規レプリカが IdM ドメインで最初の DNS サーバーの場合には特に推奨されます。

4.5.4. CA を設定したレプリカのインストール

以下の手順は、まだ IdM ドメインに含まれないマシンやクライアントにレプリカをインストールする際に使用します。詳細は「[レプリカの作成: 概要](#)」を参照してください。

1. **--setup-ca** オプションを指定して、**ipa-replica-install** を実行します。

```
[root@replica ~]# ipa-replica-install --setup-ca
```

2. サーバー上の IdM CA が root CA でも、外部の CA に従属する CA の場合でも、**--setup-ca** オプションを指定すると、最初のサーバーの設定から CA 設定がコピーされます。



注記

サポートされる CA 設定に関する詳細は「[CA 設定の決定](#)」を参照してください。

4.5.5. CA がインストールされていないサーバーからのレプリカのインストール

以下の手順は、まだ IdM ドメインに含まれないマシンやクライアントにレプリカをインストールする際に使用します。詳細は「[レプリカの作成: 概要](#)」を参照してください。



重要

サードパーティーの自己署名サーバー証明書を使用して、サーバーまたはレプリカをインストールできません。

- **ipa-replica-install** を実行して、以下のオプションを追加して必要な証明書ファイルを指定します。
 - **--dirsrv-cert-file**
 - **--dirsrv-pin**
 - **--http-cert-file**
 - **--http-pin**

これらのオプションを使用して指定するファイルに関する詳細は「[CA なしでのインストール](#)」を参照してください。

例を示します。

```
[root@replica ~]# ipa-replica-install \
```

```
--dirsrv-cert-file /tmp/server.crt \  
--dirsrv-cert-file /tmp/server.key \  
--dirsrv-pin secret \  
--http-cert-file /tmp/server.crt \  
--http-cert-file /tmp/server.key \  
--http-pin secret
```



注記

--ca-cert-file オプションを追加しないでください。**ipa-replica-install** ユーティリティーは、マスターサーバーから証明書のこの部分の情報を自動的に取得します。

4.6. 新規レプリカのテスト

レプリカの作成後に予想通りにレプリカが機能することを確認します。

1. サーバーの 1 つでユーザーを作成します。

```
[admin@server1 ~]$ ipa user-add test_user --first=Test --last=User
```

2. ユーザーが他のサーバーにも表示されるようにします。

```
[admin@server2 ~]$ ipa user-show test_user
```

4.7. レプリカのアンインストール

[「IdM サーバーのアンインストール」](#) を参照してください。

パート III. サーバーの管理

第5章 IDM サーバーおよびサービスの基本的な管理

本章では、IdM への認証方法など、IdM サーバーおよびサービスの管理に利用可能な IdM サーバーとサービス Identity Management コマンドと UI ツールについて説明します。

5.1. IDM サーバーの起動と停止

Directory Server、認証局 (CA)、DNS、Kerberos など、複数の異なるサービスが IdM サーバーにインストールされています。インストールされている全サービスと共に、IdM サーバー全体を停止、起動、再起動するには、**ipactl** ユーティリティを使用します。

IdM サーバー全体を起動します。

```
# ipactl start
```

IdM サーバー全体を停止します。

```
# ipactl stop
```

IdM サーバー全体を再起動します。

```
# ipactl restart
```

「[システム管理者のガイド](#)」に記載されているとおりに、個別のサービスだけを停止、起動、再起動するには **systemctl** ユーティリティを使用してください。たとえば、Directory Server の動作をカスタマイズする際など、**systemctl** を使用して個別サービスを管理すると便利です。設定を変更すると、Directory Server インスタンスを再起動する必要がありますが、全 IdM サービスを再起動する必要はありません。



重要

Red Hat は、複数の IdM ドメインサービスを再起動する際には **ipactl** を使用することを推奨しています。IdM サーバーにインストールされているサービス間での依存関係により、サービスを停止、開始する順番は極めて重要です。**ipactl** ユーティリティは、サービスが適切な順番に開始、停止されるようにします。

5.2. KERBEROS を使用した IDM へのログイン

IdM は、Kerberos プロトコルを使ってシングルサインオンをサポートします。Kerberos を使用すると、ユーザーは正しいユーザー名とパスワードを 1 回提示するだけで済みます。この後は、システムが認証情報をプロンプトすることなく IdM サービスにアクセスできます。

デフォルトでは、IdM ドメインに所属するマシンのみが Kerberos を使用して IdM に認証を行うことができます。ただし、Kerberos 認証が使用できるように、外部システムも設定できます。詳しい情報は「[Web UI への Kerberos 認証用の外部システム設定](#)」を参照してください。

kinit の使用

コマンドラインから IdM にログインするには、**kinit** ユーティリティを使用します。

**注記**

kinit を使用するには、krb5-workstation パッケージをインストールしておく必要があります。

ユーザー名を指定せずに実行すると、**kinit** では、ローカルシステムで現在ログイン中のユーザー名を使用して IdM にログインします。たとえば、ローカルシステムに **local_user** としてログインしている場合に、**kinit** を実行すると、**local_user** IdM ユーザーとして認証が試行されます。

```
[local_user@server ~]$ kinit
Password for local_user@EXAMPLE.COM:
```

**注記**

ローカルユーザーのユーザー名と IdM のユーザーエントリが一致しない場合は、認証に失敗します。

別の IdM ユーザーでログインして、必要なユーザー名をパラメーターとして指定して、**kinit** ユーティリティを実行してください。たとえば、**admin** ユーザーとしてログインするには、以下を実行します。

```
[local_user@server ~]$ kinit admin
Password for admin@EXAMPLE.COM:
```

Kerberos チケットの自動取得

IdM クライアントマシンのデスクトップ環境に正常にログインした後に、ユーザーの TGT を自動取得するように、**pam_krb5** の PAM (Pluggable Authentication Module) および SSSD を設定することができます。これにより、ログイン後に **kinit** を実行する必要がなくなります。

SSSD に IdM をアイデンティティおよび認証プロバイダーとして設定した IdM システムで、ユーザーが適切な kerberos プリンシパル名でログインした後に、SSSD は自動的に TGT を取得します。

pam_krb5 の設定に関する情報は、**pam_krb5(8)** の man ページを参照してください。PAM に関する一般情報は、「[システムレベルの認証ガイド](#)」を参照してください。

複数の Kerberos チケットの保管

デフォルトでは Kerberos は認証キャッシュに、ログインユーザー毎にチケット 1 つだけを保存します。ユーザーが **kinit** を実行すると必ず、Kerberos は現在保存されているチケットを新しいチケットに置き換えます。たとえば **kinit** を使用して **user_A** として認証を行った場合に、**user_B** に対して認証を行うと **user_A** のチケットはなくなります。

ユーザーの別の TGT を取得および保存するには、以前のキャッシュの内容が上書きされないように異なる認証キャッシュを設定してください。これには、以下のいずれかの方法を利用してください。

- **export KRB5CCNAME=path_to_different_cache** コマンドを使用してから **kinit** を実行し、チケットを取得します。
- **kinit -c path_to_different_cache** コマンドを実行してから、**KRB5CCNAME** 変数をリセットしてください。

デフォルトの認証キャッシュに保存されている元の TGT を復元するには以下を実行します。

1. **kdestroy** コマンドを実行します。

2. **unset \$KRB5CCNAME** コマンドを使用して、デフォルトの認証キャッシュの場所を復元します。

現在ログインしているユーザーの確認

現在認証用に保存/使用されている TGT を確認するには、**klist** ユーティリティーを使用してキャッシュされたチケットを表示します。以下の例では、キャッシュに **user_A** のチケットが含まれ、**user_A** のみが IdM サービスにアクセスできます。

```
$ klist
Ticket cache: KEYRING:persistent:0:0
Default principal: user_A@EXAMPLE.COM

Valid starting    Expires          Service principal
11/10/2015 08:35:45  11/10/2015 18:35:45  krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

5.3. IDM コマンドラインユーティリティー

IdM の基本的なコマンドラインのスクリプトは、**ipa** と呼ばれます。**ipa** スクリプトは、複数のサブコマンドの親スクリプトで、これらのサブコマンドを使用して IdM を管理します。たとえば、**ipa user-add** コマンドは新規ユーザーを追加します。

```
$ ipa user-add user_name
```

コマンドライン管理は、UI での管理に比べて利点がいくつかあります。たとえば、コマンドラインユーティリティーでは、人の介入なしに一貫性を持って管理タスクを自動化し、繰り返し実行することができます。また、コマンドラインでも、Web UI でも実行可能な管理操作がほとんどですが、タスクによってはコマンドラインからしか実行できないものもあります。



注記

以下のセクションでは、**ipa** サブコマンドの概要のみを説明します。詳しい情報は、各 IdM 管理エリア専用の他のセクションを参照してください。たとえば、**ipa** サブコマンドを使用したユーザーエントリーの詳しい情報は [11章 ユーザーアカウントの管理](#) を参照してください。

5.3.1. ipa コマンドのヘルプ

ipa スクリプトでは、サブコマンドの特定のセット (トピック) に関するヘルプを表示できます。利用可能なトピックの一覧を表示するには、**ipa help topics** コマンドを使用します。

```
$ ipa help topics

automember      Auto Membership Rule.
automount       Automount
caacl           Manage CA ACL rules.
...
```

特定のトピックのヘルプを表示するには、**ipa help topic_name** コマンドを使用します。たとえば、**automember** トピックに関する情報を表示するには、以下を実行します。

```
$ ipa help automember

Auto Membership Rule.
```

```
Bring clarity to the membership of hosts and users by configuring
inclusive
or exclusive regex patterns, you can automatically assign a new entries
into
a group or hostgroup based upon attribute information.
```

```
...
```

EXAMPLES:

```
Add the initial group or hostgroup:
ipa hostgroup-add --desc="Web Servers" webservers
ipa group-add --desc="Developers" devel
...
```

ipa スクリプトは、利用可能な **ipa** コマンドの一覧も表示できます。これには、**ipa help commands** コマンドを使用します。

```
$ ipa help commands
automember-add                Add an automember rule.
automember-add-condition      Add conditions to an automember
rule.
...
```

個別の **ipa** コマンドに関する詳しいヘルプは、以下のようにコマンドに **--help** オプションを指定します。

```
$ ipa automember-add --help

Usage: ipa [global-options] automember-add AUTOMEMBER-RULE [options]

Add an automember rule.
Options:
  -h, --help                show this help message and exit
  --desc=STR                 A description of this auto member rule
  ...
```

ipa ユーティリティーに関する詳しい情報は、ipa(1) の man ページを参照してください。

5.3.2. 値のリストの設定

IdM は、以下のようにリストにエントリー属性を保存します。

```
ipaUserSearchFields: uid,givenname,sn,telephonenumber,ou,title
```

属性一覧を更新すると、以前のリストが上書きされます。たとえば、この属性だけを指定して 1 つの属性を追加すると、以前に定義されている一覧全体が、この新しい単一属性に置き換えられます。そのため、属性一覧を変更する場合は、更新一覧すべてを指定する必要があります。

属性一覧の指定方法の中で IdM がサポートしているのは、以下のとおりです。

- 以下のように、同じコマンド呼び出しで複数回、引数を指定します。


```
$ ipa permission-add --permissions=read --permissions=write --
permissions=delete
```

- リストを中括弧で囲みます。これでシェルによる拡張が可能になります。例を示します。

```
$ ipa permission-add --permissions={read,write,delete}
```

5.3.3. 特殊文字の使用

ipa コマンドで、山括弧 (< および >)、アンパサンド (&)、アスタリスク (*)、パイプ (|) などの特殊文字が含まれるコマンドラインの引数を指定すると、バックスラッシュ (\) を使用してこのような特殊文字をエスケープする必要があります。以下のように、アスタリスク (*) をエスケープします。

```
$ ipa certprofile-show certificate_profile --out=exported\*profile.cfg
```

コマンドにエスケープされていない特殊文字を含めると、シェルがこれらの文字を正しく解析できないので、予想通りに機能しなくなります。

5.3.4. IdM エントリーの検索

IdM エントリーの表示

ipa *-find コマンドを使用して、特定のタイプの IdM エントリーを検索します。以下に例を示します。

- 全ユーザーを表示します。

```
$ ipa user-find
-----
4 users matched
-----
...
```

- 指定の属性に **keyword** が含まれるユーザーグループを表示するには、以下を実行します。

```
$ ipa group-find keyword
-----
2 groups matched
-----
...
```

IdM がユーザーおよびユーザーグループを検索するための属性を設定するには、[「ユーザーおよびユーザーグループの検索属性の設定」](#)を参照してください。

ユーザーグループの検索の際には、特定のユーザーを含むグループに検索結果を絞り込むことも可能です。

```
$ ipa group-find --user=user_name
```

また、特定のユーザーを含まないグループを検索することもできます。

```
$ ipa group-find --no-user=user_name
```

特定のエントリーの詳細表示

特定の IdM エントリーに関する詳細を表示するには、以下のように **ipa *-show** コマンドを使用します。

```
$ ipa host-show server.example.com
Host name: server.example.com
Principal name: host/server.example.com@EXAMPLE.COM
...
```

5.3.4.1. 検索サイズおよび時間制限の調整

ユーザー一覧の表示など、検索結果によっては、エントリー数が大量に返される可能性があります。これらの検索操作を調節して、**ipa user-find** などの **ipa *-find** コマンドを実行時や、Web UI で適切な一覧を表示する際に、全体的なサーバーのパフォーマンスを向上できます。

検索サイズ

- クライアント、IdM コマンドラインツール、IdM Web UI からサーバーに送信される要求に対して、返される最大エントリー数を定義します。
- デフォルト値: エントリー数 100 件

検索の時間制限:

- 検索の実行までにサーバーが待機する最大時間を定義します。検索がこの制限に到達したら、サーバーは検索を停止し、停止するまでの期間に検出されたエントリーを返します。
- デフォルト値: 2 秒

値が **-1** に設定されている場合、IdM は検索時に制限を適用しません。



重要

検索のサイズや時間制限を高く設定しすぎると、サーバーのパフォーマンスにマイナスの影響が出る可能性があります。

Web UI: 検索サイズおよび時間制限の調整

全クエリーに対してグローバルに制限を調節するには以下を実行します。

1. **IPA Server → Configuration** を選択します。
2. **Search Options** エリアに必要な値を設定します。
3. ページ上部にある **Save** をクリックします。

コマンドライン: 検索サイズおよび時間制限の調整

全クエリーに対してグローバルに制限を調節するには、**--searchrecordslimit** および **--searchtimelimit** オプションを指定して、**ipa config-mod** コマンドを実行します。

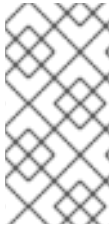
```
$ ipa config-mod --searchrecordslimit=500 --searchtimelimit=5
```

コマンドラインから、特定のクエリーに対する制限のみを調節することもできます。これには、以下のようにコマンドに **--sizelimit** または **--timelimit** オプションを指定します。

```
$ ipa user-find --sizelimit=200 --timelimit=120
```

5.4. IDM WEB UI

Identity Management Web UI とは、IdM 管理の Web アプリケーションのことで、**ipa** コマンドラインユーティリティにある大半の機能が含まれます。そのため、ユーザーは IdM の管理に UI またはコマンドラインのいずれかを選択できます。



注記

ログイン中のユーザーが利用できる管理操作は、そのユーザーに割り当てられている権限により異なります。**admin** ユーザーおよびその他に管理者権限のあるユーザーは、すべての管理タスクを使用できます。通常のユーザーが使用できるのは、自身のユーザーアカウントに関する操作のみに限られます。

5.4.1. サポート対象の Web ブラウザー

Identity Management では、以下のブラウザーを使用して Web UI に接続することができます。

- Mozilla Firefox 38 以降
- Google Chrome 46 以降

5.4.2. Web UI へのアクセスおよび認証

Web UI には、IdM サーバーおよびクライアントマシンの両方から、さらに IdM ドメイン外のマシンからもアクセスできます。ただし、ドメイン外のマシンから UI にアクセスするには、IdM 以外のシステムで IdM Kerberos ドメインに接続できるように設定しておく必要があります。詳細は、「[Web UI への Kerberos 認証用の外部システム設定](#)」を参照してください。

5.4.2.1. Web UI へのアクセス

Web UI をアクセスして、ブラウザーアドレスバーに IdM サーバーの URL を入力します。

```
https://server.example.com
```

これで、ブラウザーに Web UI ログイン画面が表示されます。

図5.1 Web UI ログイン画面

5.4.2.2. 利用可能なログイン方法

ユーザーは、以下の方法で Web UI に対して認証を行うことができます。

有効な Kerberos チケットの使用

ユーザーが **kinit** ユーティリティから TGT を取得して、その TGT が有効な場合には **Login** ボタンをクリックすることで自動的にユーザーが認証されます。ブラウザーは、Kerberos 認証をサポートするように正しく設定しておく必要がある点にご注意ください。

Kerberos TGT の取得に関する情報は、「[Kerberos を使用した IdM へのログイン](#)」を参照してください。ブラウザーの設定に関する情報は、「[Kerberos 認証用のブラウザーの設定](#)」を参照してください。

ユーザー名およびパスワードの提示

ユーザー名およびパスワードを使用して認証するには、Web UI のログイン画面でユーザー名とパスワードを入力します。

IdM は、ワンタイムパスワード (OTP) 認証もサポートします。詳しい情報は「[ワンタイムパスワード](#)」を参照してください。

スマートカードの使用

詳しい情報は「[スマートカードを使用して Identity Management Web UI への認証を行う方法](#)」を参照してください。

ユーザー認証に成功すると、IdM 管理ウィンドウが開きます。

RED HAT IDENTITY MANAGEMENT

Red Hat Access

Administrator

Identity

Policy

Authentication

Network Services

IPA Server

Users

User Groups

Hosts

Host Groups

Netgroups

Services

Automember

User categories

Active users

Stage users

Preserved users

Active users

Search

Refresh

Delete

+ Add

- Disable

Enable

Actions

	User login	First name	Last name	Status	UID	Email address
<input type="checkbox"/>	admin		Administrator	Enabled	1890600000	

図5.2 IdM Web UI のレイアウト

5.4.2.3. AD ユーザーとしての IdM Web UI への認証

Active Directory (AD) ユーザーは、ユーザー名とパスワードを使用して IdM web UI にログインできます。web UI では、AD ユーザーは管理者権限に関連付けられた管理操作を実行できる IdM ユーザーとは違い、自分のユーザーアカウントに関連する操作だけを実行できます。

AD ユーザー向けの web UI ログインを有効にするには、IdM 管理者は、各 AD ユーザーの ID オーバーライドを Default Trust View で定義する必要があります。以下に例を示します。

```
[admin@server ~]$ ipa idoverrideuser-add 'Default Trust View'
ad_user@ad.example.com
```

AD における ID ビューの詳細については、『Windows 統合ガイド』の「[Active Directory 環境での ID ビューの使用](#)」を参照してください。

5.4.3. Kerberos 認証用のブラウザーの設定

Kerberos 認証情報での認証を有効化するには、ブラウザーが Kerberos との交渉をサポートして IdM ドメインにアクセスできるように設定する必要があります。お使いのブラウザーを Kerberos 認証用に適切に設定されていない場合は、IdM Web UI ログイン画面で **Login** をクリックした後にエラーメッセージが表示されます。

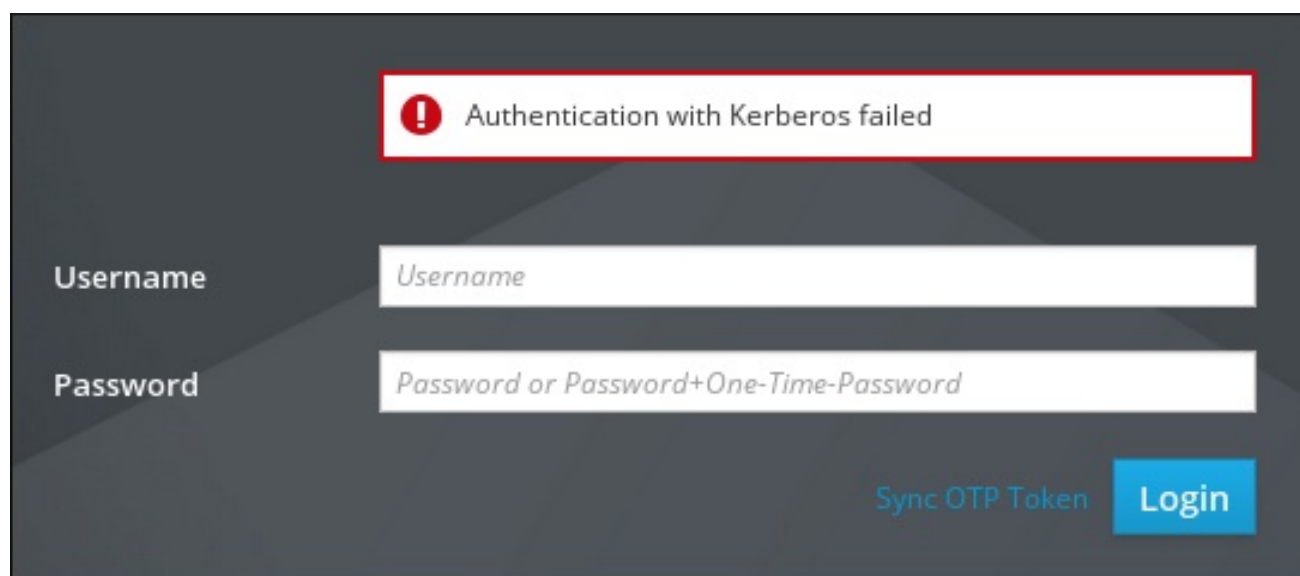


図5.3 Kerberos 認証エラーメッセージ

Kerberos 認証ようにブラウザーを設定する方法には 3 通りあります。

- IdM web UI から自動的に設定。このオプションは、Firefox でのみ利用できます。詳細は「[Web UI の Firefox での自動設定](#)」を参照してください。
- IdM クライアントのインストール中にコマンドラインから自動的に設定。このオプションは、Firefox でのみ利用できます。詳細は「[コマンドラインからの Firefox 自動設定](#)」を参照してください。
- Firefox オプション設定で手動設定。このオプションは、サポートされているブラウザーすべてで利用できます。詳細は「[手動によるブラウザーの設定](#)」を参照してください。



注記

『システムレベルの認証ガイド』には、「[Firefox でシングルサインオンに Kerberos を使用するように設定する手順](#)」が含まれます。Kerberos 認証が予想通りに機能しない場合には、このトラブルシューティングガイドを参照してください。

Web UI の Firefox での自動設定

IdM web UI から Firefox を自動で設定するには以下を実行します。

1. Web UI のログイン画面でブラウザー設定のリンクをクリックします。

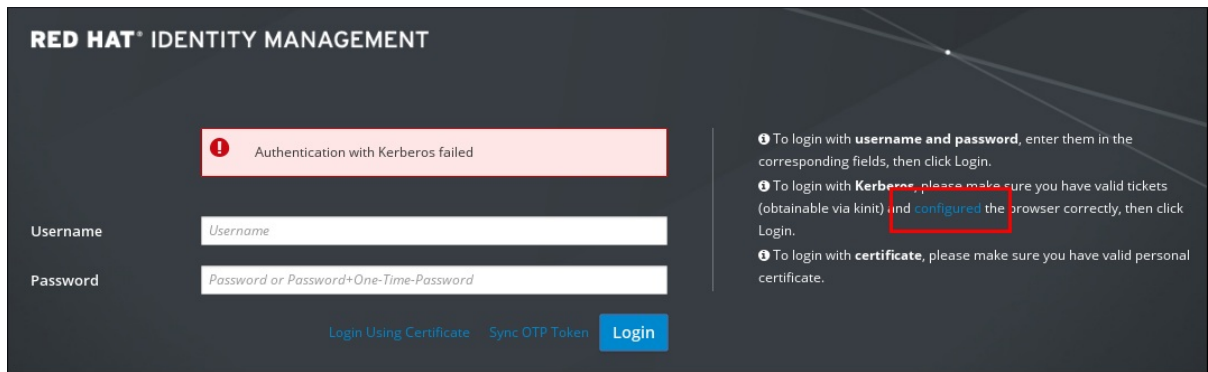


図5.4 Web UI でのブラウザ設定へのリンク

2. Firefox の設定リンクを選択して、Firefox 設定ページを開きます。

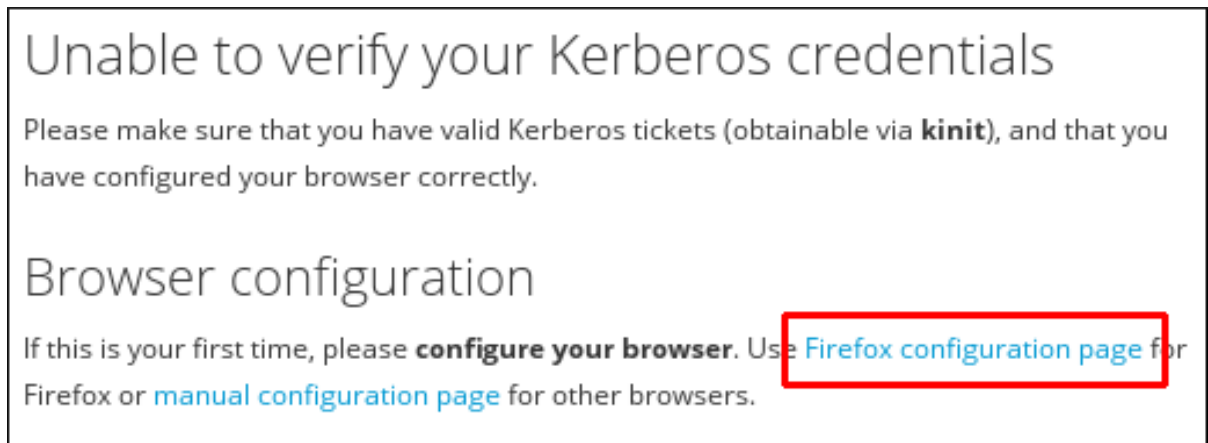


図5.5 Firefox 設定ページへのリンク

3. Firefox 設定ページの手順に従います。

コマンドラインからの Firefox 自動設定

Firefox は、IdM クライアントのインストール時にコマンドラインから設定できます。これには、**ipa-client-install** ユーティリティで IdM クライアントをインストールする際に **--configure-firefox** オプションを使用します。

```
# ipa-client-install --configure-firefox
```

--configure-firefox オプションは、シングルサインオン (SSO) での Kerberos を有効化するデフォルトの Firefox 設定で、グローバル設定ファイルを作成します。

手動によるブラウザの設定

ブラウザを手動で設定するには以下を実行します。

1. Web UI のログイン画面でブラウザ設定のリンクをクリックします。

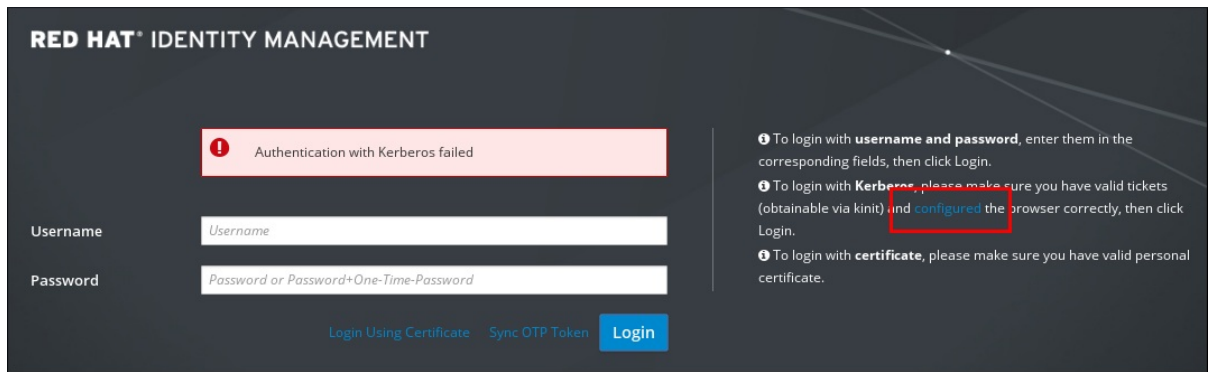


図5.6 Web UI でのブラウザ設定へのリンク

2. 手動でのブラウザ設定のリンクを選択します。

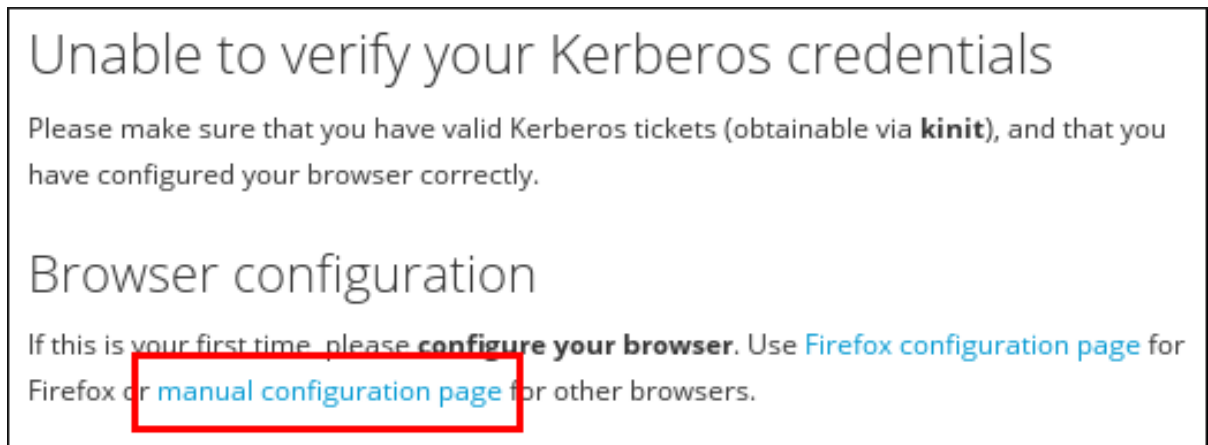


図5.7 手動設定ページへのリンク

3. ブラウザーの設定の説明を探して、手順に従います。

5.4.4. Web UI への Kerberos 認証用の外部システム設定

IdM ドメインのメンバーでないシステムから Web UI への Kerberos 認証を有効にするには、外部マシンで IdM 固有の Kerberos 設定ファイルを定義する必要があります。特にインフラストラクチャーに複数のレルムや重複ドメインが含まれる場合に、外部マシンで Kerberos 認証を有効化すると便利です。

Kerberos 設定ファイルを作成するには、以下を実行します。

1. 以下のように、IdM サーバーから外部マシンに `/etc/krb5.conf` ファイルをコピーします。

```
# scp /etc/krb5.conf
root@externalmachine.example.com:/etc/krb5_ipa.conf
```



警告

外部マシンの既存の `krb5.conf` ファイルは上書きしないでください。

2. 外部マシン上で、端末のセッションがコピーされた IdM Kerberos 設定ファイルを使用するよう設定します。

```
$ export KRB5_CONFIG=/etc/krb5_ipa.conf
```

3. 「[Kerberos 認証用のブラウザーの設定](#)」で記載されているように、外部マシンのブラウザーを設定します。

外部システムのユーザーは、IdM サーバーに対して **kinit** ユーティリティーを使用できるようになりました。

5.4.5. Web UI でのプロキシサーバーおよびポート転送

Web UI へのアクセスにプロキシサーバーを使用する場合には、IdM での追加設定は必要ありません。

ポート転送は IdM サーバーではサポートされていませんが、IdM ではプロキシサーバーを使用することができるので、OpenSSH と SOCKS オプションでプロキシ転送を使用して、ポート転送に似た操作を設定することができます。これは、**ssh** ユーティリティーに **-D** オプションを指定して設定できます。**-D** オプションの使用に関する詳細は、ssh(1) の man ページを参照してください。

第6章 レプリケーショントポロジーの管理

本章では、Identity Management (IdM) ドメインでのサーバー間のレプリケーションを管理する方法について説明します。



注記

本章では、Red Hat Enterprise Linux 7.3 で導入された簡単なトポロジーの管理について説明します。この手順では、ドメインレベル 1 ([7章 ドメインレベルの表示と引き上げ](#) を参照) が必要になります。

ドメインレベル 0 でのトポロジー管理については、「[レプリカとレプリカ合意の管理](#)」を参照してください。

最初のレプリカのインストールとレプリケーションについての基本的な情報は、[4章 Identity Management のレプリカのインストールとアンインストール](#) を参照してください。

6.1. レプリカ合意、トポロジーサフィックス、およびトポロジーセグメント

レプリカ合意

IdM サーバーに保存されているデータは、レプリカ合意に基づいて複製されます。2 サーバー間でレプリカ合意が設定されていると、これらのサーバーのデータは共有されます。

レプリカ合意は常に双方向のものです。1 台目のレプリカから 2 台目のレプリカにデータが複製されるほかに、2 台目のレプリカから 1 台目のレプリカにもデータが複製されます。



注記

詳細については、「[IdM レプリカの説明](#)」を参照してください。

トポロジーサフィックス

トポロジーサフィックスは、複製されたデータを保存します。IdM では、**domain** および **ca** の 2 つのタイプのサフィックスをサポートしています。各サフィックスは、別個のバックエンドと別個のレプリカトポロジーを表しています。

レプリカ合意が設定されると、2 台のサーバー上の同じタイプの 2 つのトポロジーサフィックスを結びつけます。

domain サフィックス: **dc=example,dc=com**

domain サフィックスには、ドメイン関連データすべてが含まれます。

2 つのレプリカの **domain** サフィックス間でレプリカ合意が設定されると、ユーザー、グループ、およびポリシーなどのディレクトリーデータが共有されます。

ca サフィックス: **o=ipaca**

ca サフィックスには、Certificate System コンポーネント用のデータが含まれます。これは、証明局 (CA) がインストールされているサーバーにのみ存在します。

2 つのレプリカの **ca** サフィックス間でレプリカ合意が設定されると、証明書データが共有されます。

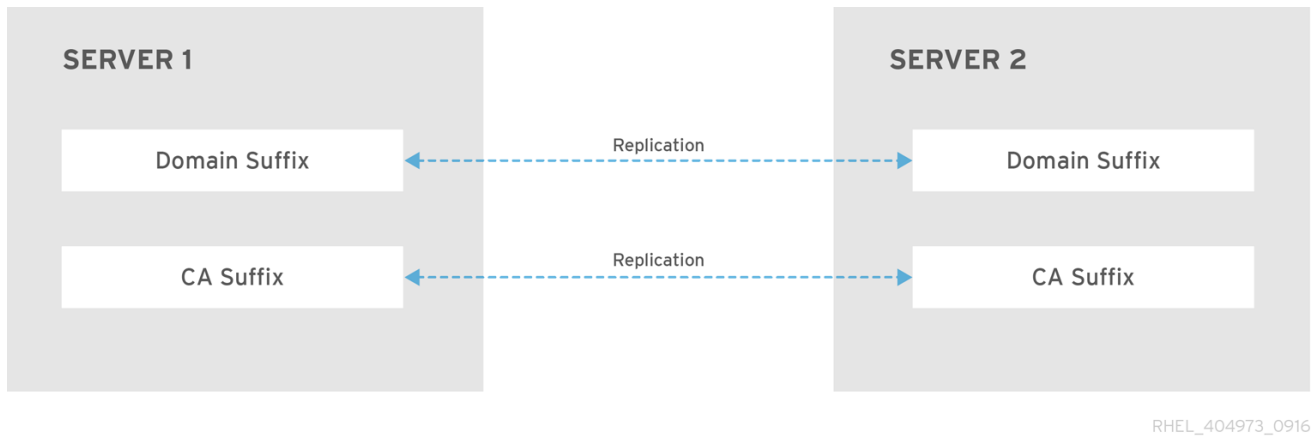


図6.1 トポロジーサフィックス

新規レプリカのインストール時には、**ipa-replica-install** スクリプトが 2 つのサーバー間に初期トポロジーセグメントをセットアップします。

例6.1 トポロジーサフィックスの表示

ipa topologysuffix-find コマンドでトポロジーサフィックスの一覧が表示されます。

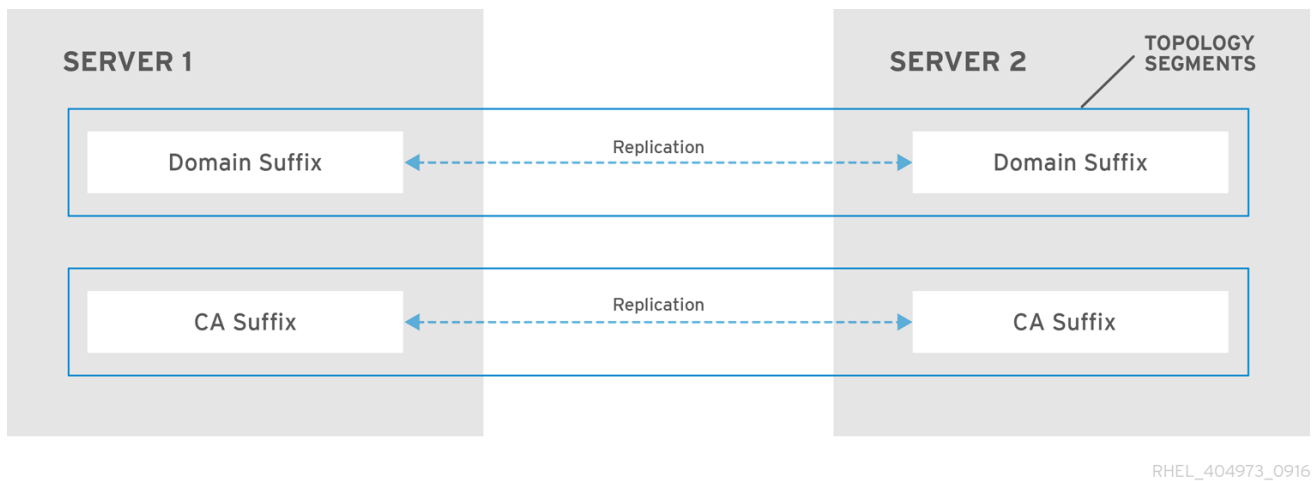
```
$ ipa topologysuffix-find
-----
2 topology suffixes matched
-----
  Suffix name: ca
  Managed LDAP suffix DN: o=ipaca

  Suffix name: domain
  Managed LDAP suffix DN: dc=example,dc=com
-----
Number of entries returned 2
-----
```

トポロジーセグメント

2 つのレプリカのサフィックス間にレプリカ合意がある場合、サフィックスは **トポロジーセグメント** を形成します。このトポロジーセグメントは、**左ノード** と **右ノード** で構成されます。ノードは、レプリカ合意に参加したサーバーを表しています。

IdM のトポロジーセグメントは常に双方向です。各セグメントは、サーバー A からサーバー B と、サーバー B からサーバー A という 2 つのレプリカ合意を表しています。このため、データは双方向で複製されます。



RHEL_404973_0916

図6.2 トポロジーセグメント

例6.2 トポロジーセグメントの表示

ipa topologysegment-find コマンドで、domain または CA サフィックスに設定されたトポロジーセグメントが表示されます。たとえば、domain サフィックスの場合は以下のようになります。

```
$ ipa topologysegment-find
Suffix name: domain
-----
1 segment matched
-----
Segment name: server1.example.com-to-server2.example.com
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
-----
Number of entries returned 1
-----
```

この例では、ドメイン関連のデータのみが **server1.example.com** と **server2.example.com** の 2 つのサーバー間で複製されます。

特定セグメントの詳細を表示するには、**ipa topologysegment-show** コマンドを使用します。

```
$ ipa topologysegment-show
Suffix name: domain
Segment name: server1.example.com-to-server2.example.com
Segment name: server1.example.com-to-server2.example.com
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
```

6.2. WEB UI: トポロジーグラフを使用したレプリカトポロジーの管理

トポロジーグラフへのアクセス

web UI のトポロジーグラフでは、ドメイン内におけるサーバー間の関係が表示できます。

1. **IPA Server** → **Topology** → **Topology Graph** を選択します。

2. トポロジーに加えた変更がグラフに反映されていない場合は、**Refresh** をクリックします。

トポロジービューのカスタマイズ

トポロジーノードは、マウスでドラッグすると移動できます。

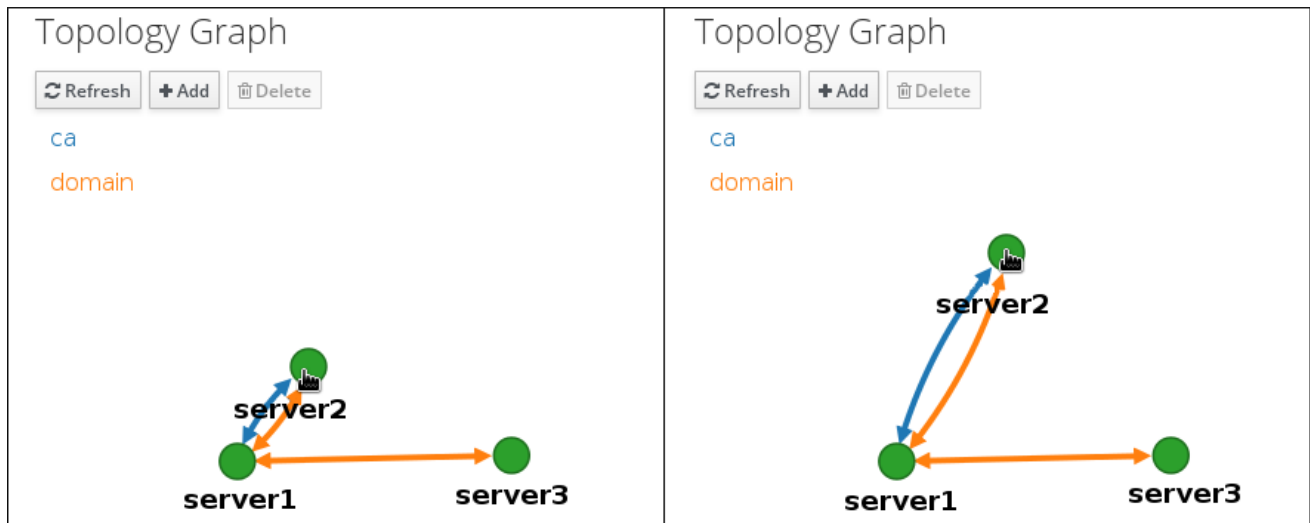


図6.3 トポロジーグラフでのノードの移動

マウスホイールを使うと、グラフの拡大、縮小ができます。

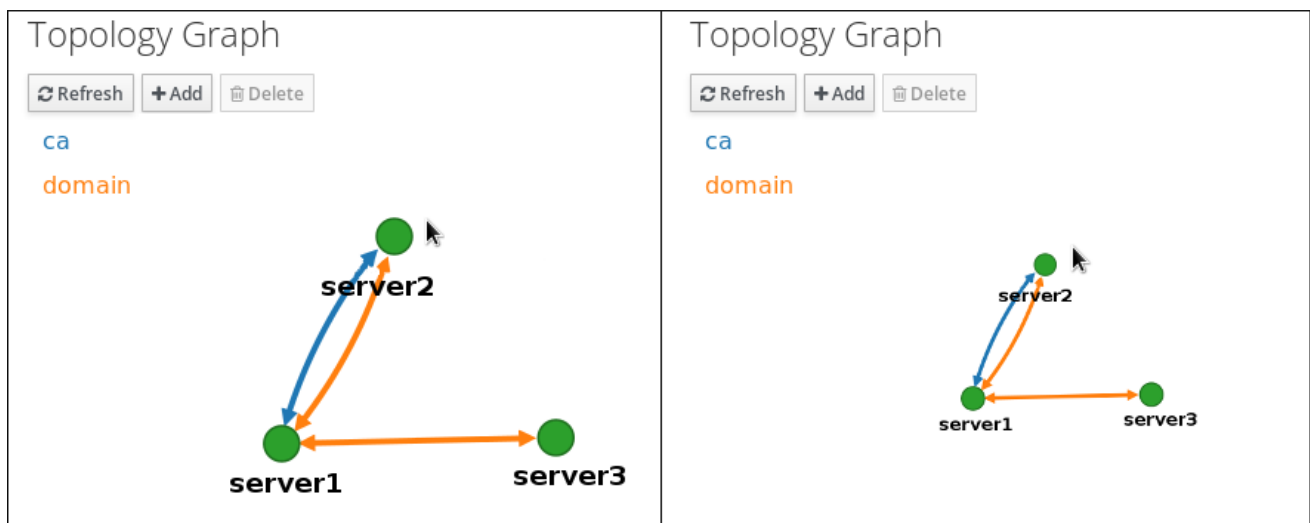


図6.4 トポロジーグラフの拡大

マウスの左ボタンを長押しすると、トポロジーグラフ自体を動かすことができます。

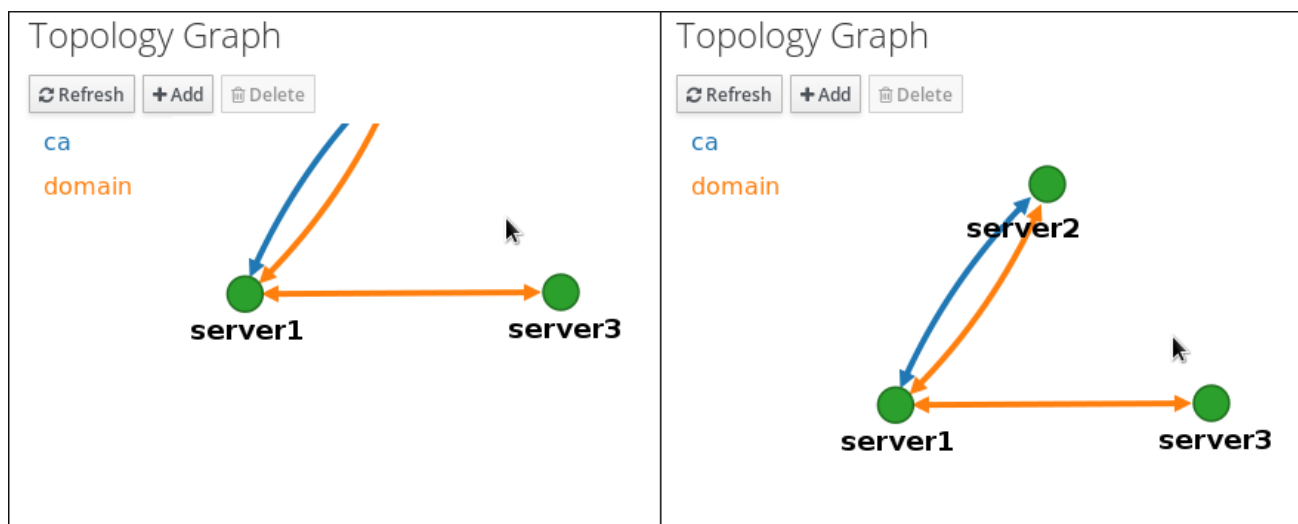


図6.5 トポロジーグラフの移動

トポロジーグラフの意味

ドメインのレプリカ合意に参加しているサーバーは、オレンジ色の矢印で結ばれています。CA のレプリカ合意に参加しているサーバーは、青色の矢印で結ばれています。

トポロジーグラフの推奨例

図6.6「トポロジーの推奨例」は、4 つのサーバーにおけるトポロジーの推奨例を示しています。各サーバーは少なくとも 2 台のサーバーに連結されており、複数のサーバーが CA マスターとなっています。

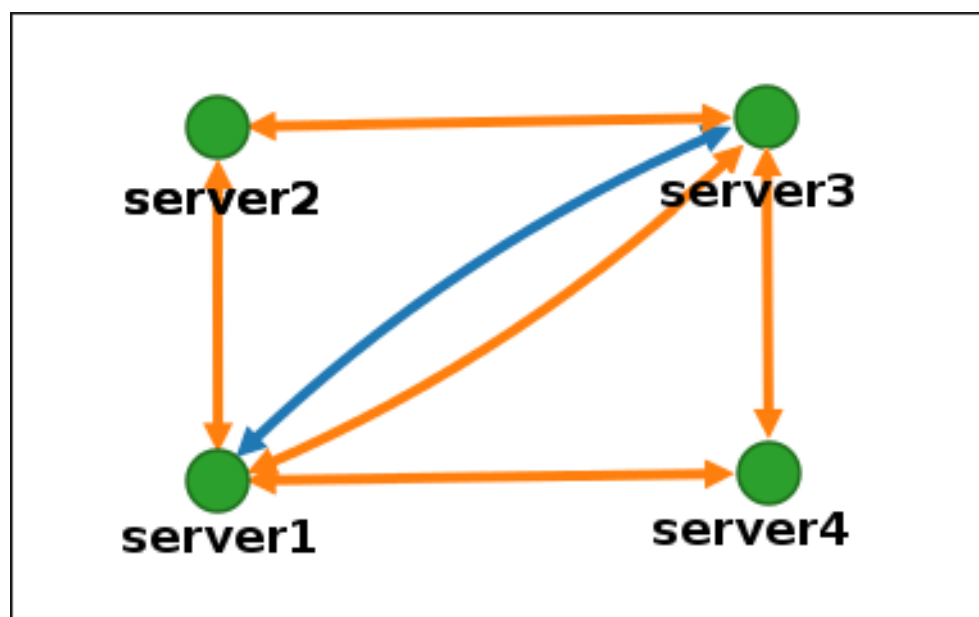


図6.6 トポロジーの推奨例

トポロジーグラフ: 非推奨例

図6.7「非推奨のトポロジー例: 単一障害点」では、**server1** が単一障害点となっています。他のすべてのサーバーはこのサーバーとレプリカ合意を結んでいますが、これ以外には合意が設定されていません。このため、**server1** に障害が発生すると、他のサーバーは孤立してしまいます。

このようなトポロジーは作成しないでください。

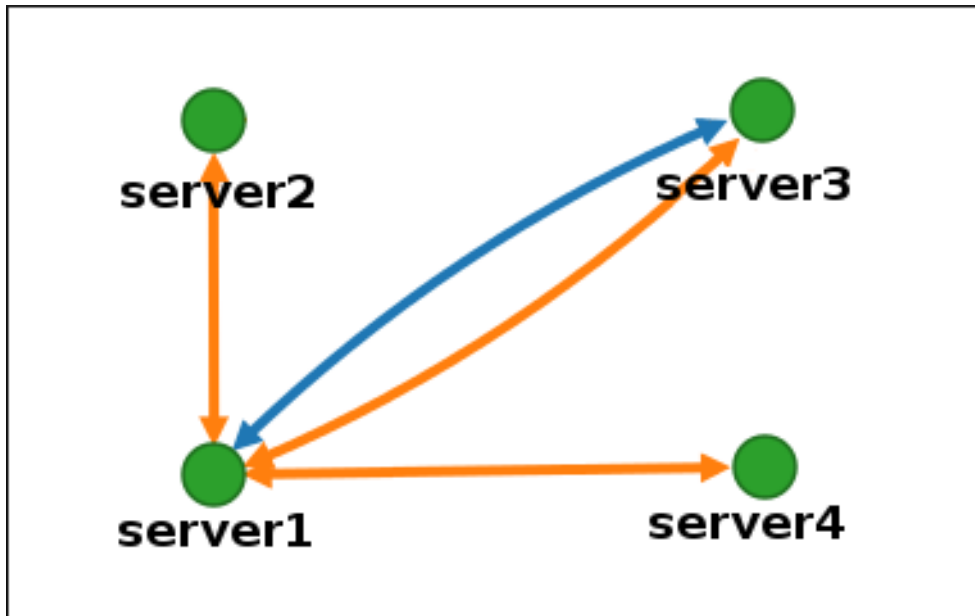


図6.7 非推奨のトポロジー例：単一障害点

推奨されるトポロジーの詳細については、「[レプリカに関するデプロイメントの考慮事項](#)」を参照してください。

6.2.1.2 サーバー間でのレプリカ設定

1. トポロジーグラフで、ノード上にカーソルを持ていきます。

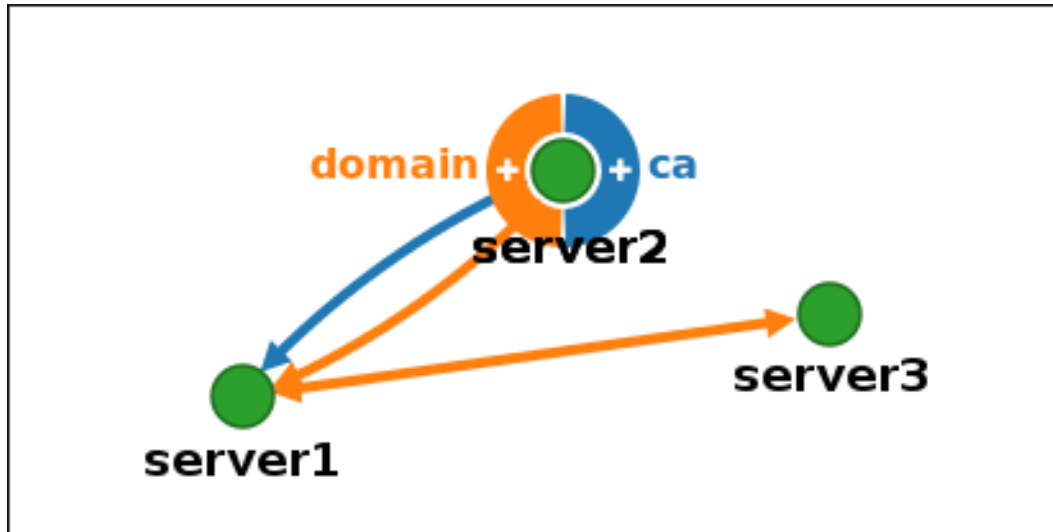


図6.8 Domain または CA オプション

2. 作成するトポロジーセグメントのタイプに合わせて、円の **domain** または **ca** をクリックします。
3. カーソルの下に新たなレプリカ合意を表す矢印が表示されます。カーソルを別のサーバーノードに移動して、そこでクリックします。

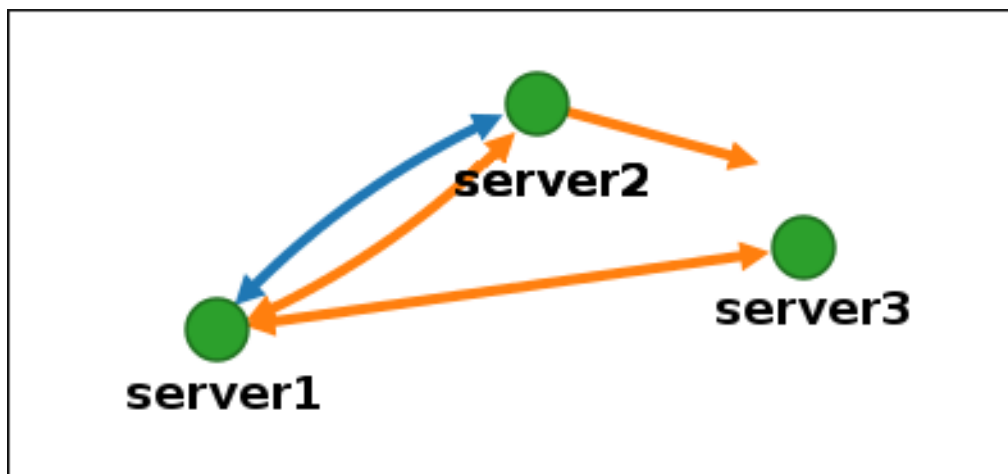


図6.9 新セグメントの作成

4. **Add Topology Segment** ウィンドウで **Add** をクリックして、新セグメントの属性を確認します。

IdM が 2 つのサーバー間に新規トポロジーセグメントを作成し、これらのサーバーはレプリカ合意に参加します。トポロジーグラフに更新されたレプリカトポロジーが表示されます。

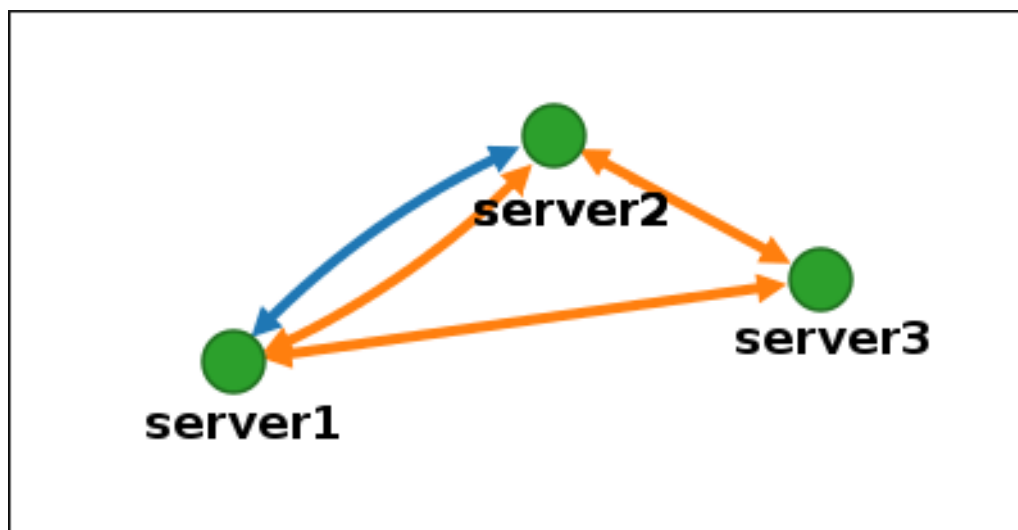


図6.10 新規セグメント作成後のグラフ

6.2.2.2 サーバー間のレプリカの削除

1. 削除するレプリカ合意を表す矢印をクリックします。矢印がハイライト表示されます。

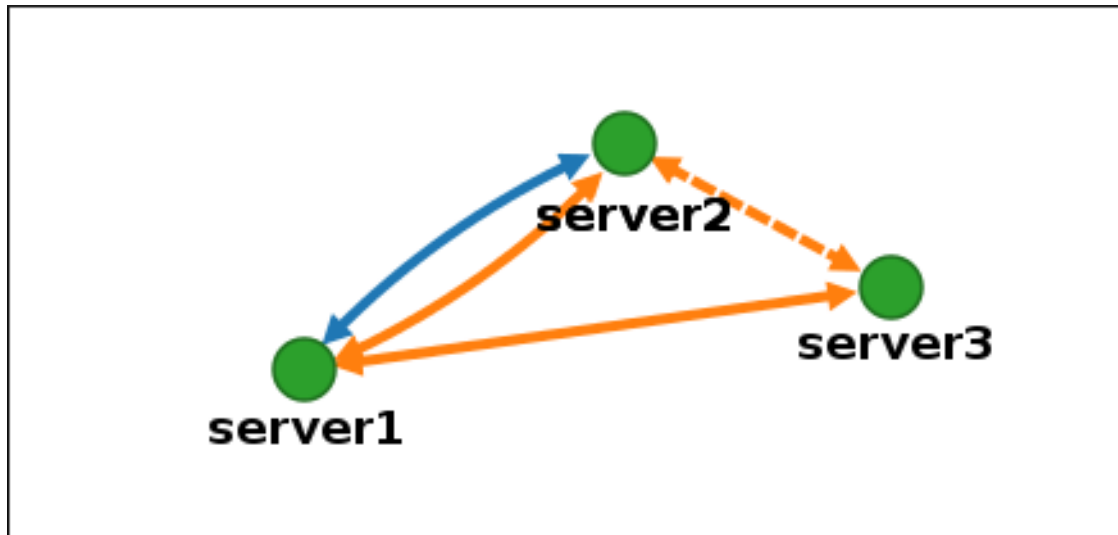


図6.11 トポロジーセグメントのハイライト表示

2. **Delete** をクリックします。
3. **Confirmation** ウィンドウで **OK** をクリックします。

IdM が 2 つのサーバー間のトポロジーセグメントを削除し、これでサーバーのレプリカ合意も削除されます。トポロジーグラフに更新されたレプリカトポロジーが表示されます。

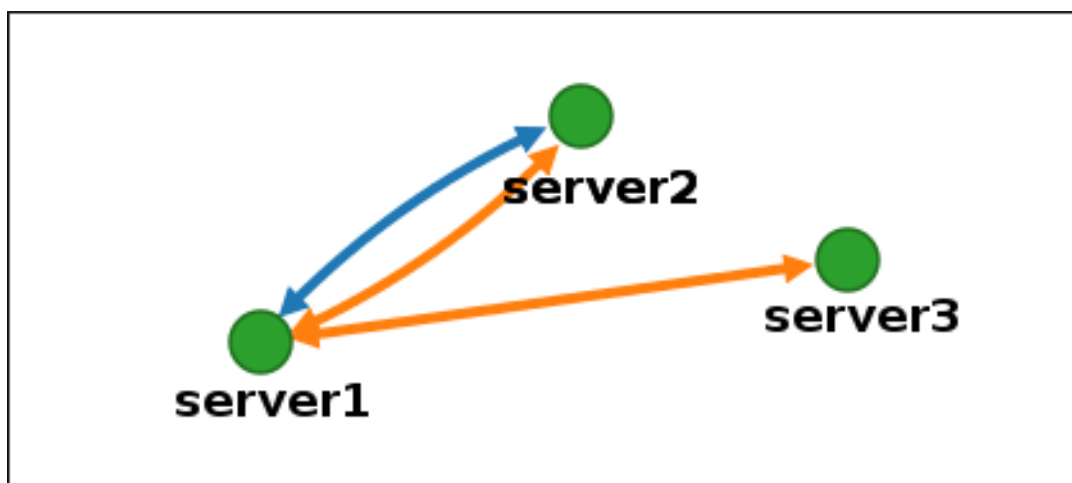


図6.12 トポロジーセグメントの削除

6.3. コマンドライン: IPA TOPOLOGY* コマンドを使用したトポロジーの管理

6.3.1. トポロジー管理コマンドのヘルプ

レプリカトポロジーの管理に使用する全コマンドを表示するには、以下を実行します。

```
$ ipa help topology
```

特定のコマンドに関する詳細のヘルプを表示するには、そのコマンドを **--help** オプションと実行します。

```
$ ipa topologysuffix-show --help
```


6.3.2. 2 サーバー間でのレプリカ設定

1. 2 サーバー間のトポロジーセグメントを作成するには、**ipa topologysegment-add** コマンドを使用します。プロンプトに応じて、以下を提供します。

- **domain** または **ca** のトポロジーサフィックス



注記

ca サフィックス間でセグメントを作成する場合は、両方のサーバーに CA がインストールされている必要があります。[「CA の既存の IdM ドメインへのインストール」](#) を参照してください。

- 各サーバーを表す左ノードと右ノード
- オプションで、セグメントのカスタム名

以下に例を示します。

```
$ ipa topologysegment-add
Suffix name: domain
Left node: server1.example.com
Right node: server2.example.com
Segment name [server1.example.com-to-server2.example.com]:
new_segment
-----
Added segment "new_segment"
-----
Segment name: new_segment
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
```

新規セグメントを追加すると、サーバーがレプリカ合意に参加します。

2. オプションで **ipa topologysegment-show** コマンドを使用すると、新規セグメントが設定されたことを確認することができます。

```
$ ipa topologysegment-show
Suffix name: domain
Segment name: new_segment
Segment name: new_segment
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both
```

6.3.3. 2 サーバー間のレプリカの削除

1. レプリケーションを停止するには、対応するサーバー間のレプリカセグメントを削除する必要があります。これを実行するには、セグメント名が必要になります。

セグメント名が分からない場合は、**ipa topologysegment-find** コマンドを実行して全セグメントを表示し、該当するセグメントを見つけます。プロンプトに応じて、**domain** または **ca** のトポロジーサフィックスを提供します。例を示します。

```
$ ipa topologysegment-find
Suffix name: domain
-----
8 segments matched
-----
Segment name: new_segment
Left node: server1.example.com
Right node: server2.example.com
Connectivity: both

...

-----
Number of entries returned 8
-----
```

2. 2 サーバー間のトポロジーセグメントを削除するには、**ipa topologysegment-del** コマンドを使用します。

```
$ ipa topologysegment-del
Suffix name: domain
Segment name: new_segment
-----
Deleted segment "new_segment"
-----
```

セグメントを削除すると、レプリカ合意が削除されます。

3. オプションで **ipa topologysegment-find** コマンドを使用すると、セグメントが削除されたことを確認することができます。

```
$ ipa topologysegment-find
Suffix name: domain
-----
7 segments matched
-----
Segment name: server2.example.com-to-server3.example.com
Left node: server2.example.com
Right node: server3.example.com
Connectivity: both

...

-----
Number of entries returned 7
-----
```

6.4. トポロジーからサーバーを削除する

以下のいずれかの条件が当てはまる場合は、IdM ではトポロジーからサーバーを削除することができません。

- 削除されるサーバーが、トポロジーと他のサーバーを結んでいる唯一のサーバーである場合。この場合に当該サーバーが削除されると、他のサーバーは孤立してしまうため、これは許可されません。
- 削除されるサーバーが最後の CA または DNS サーバーである場合。

このような状況では、削除を試みるとエラーが出て失敗します。たとえば、コマンドラインでは以下のようになります。

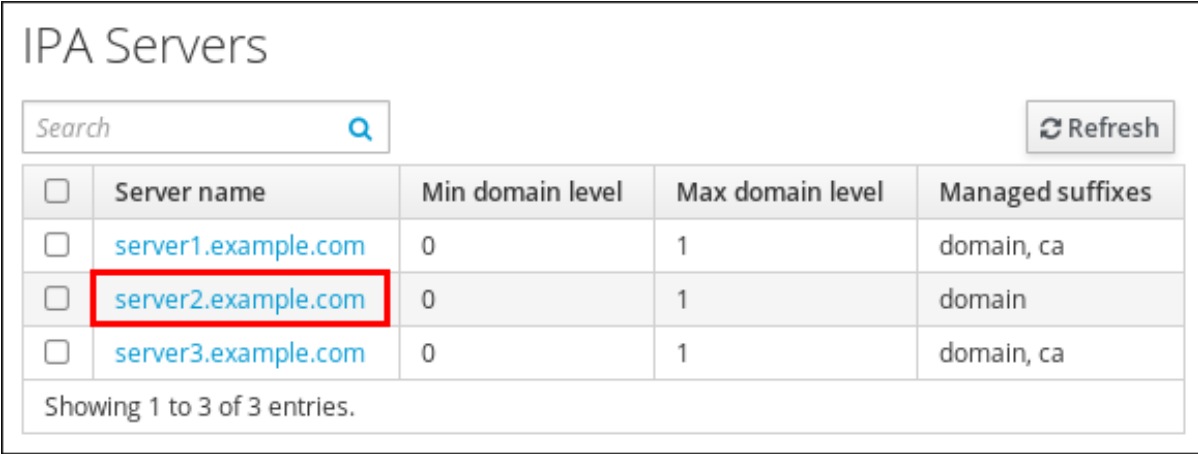
```
$ ipa server-del
Server name: server1.example.com
Removing server1.example.com from replication topology, please wait...
ipa: ERROR: Server removal aborted:

Removal of 'server1.example.com' leads to disconnected topology in suffix
'domain':
Topology does not allow server server2.example.com to replicate with
servers:
    server3.example.com
    server4.example.com
...
```

6.4.1. Web UI: トポロジーからサーバーを削除する

マシンからサーバーコンポーネントをアンインストールせずに、トポロジーからサーバーを削除するには、以下の手順に従います。

1. **IPA Server** → **Topology** → **IPA Servers** を選択します。
2. 削除するサーバー名をクリックします。



<input type="checkbox"/>	Server name	Min domain level	Max domain level	Managed suffixes
<input type="checkbox"/>	server1.example.com	0	1	domain, ca
<input type="checkbox"/>	server2.example.com	0	1	domain
<input type="checkbox"/>	server3.example.com	0	1	domain, ca

Showing 1 to 3 of 3 entries.

図6.13 サーバーの選択

3. **Delete Server** をクリックします。

6.4.2. コマンドライン: トポロジーからサーバーを削除する



重要

サーバーを削除すると、元に戻すことはできません。サーバーを削除した後にこれをトポロジーに再度導入する唯一の方法は、マシンに新規レプリカをインストールすることになります。

server1.example.com を削除するには、以下を実行します。

1. 別のサーバーで **ipa server-del** コマンドを実行して **server1.example.com** を削除します。このコマンドで、当該サーバーをポイントしているすべてのトポロジーセグメントが削除されます。

```
[user@server2 ~]$ ipa server-del
Server name: server1.example.com
Removing server1.example.com from replication topology, please
wait...
-----
Deleted IPA server "server1.example.com"
-----
```

2. **server1.example.com** で **ipa server-install --uninstall** コマンドを実行して、マシンからこのサーバーコンポーネントをアンインストールします。

```
[root@server1 ~]# ipa server-install --uninstall
```

6.5. サーバーロールの管理

IdM サーバーは、インストールされているサービスを基に、CA サーバー、DNS サーバー、またはキー回復機関 (KRA) サーバーなどの各種の *サーバーロール* を実行することができます。

6.5.1. サーバーロールの表示

Web UI: サーバーロールの表示

サポートされるサーバーロールの全一覧を確認するには、**IPA Server → Topology → Server Roles** をクリックします。

- Role status が **absent** の場合は、トポロジー内でそのロールを実行しているサーバーがないことを示しています。
- Role status が **enabled** の場合は、トポロジー内でそのロールを実行しているサーバーが 1 台以上あることを示しています。

Server Roles		Refresh
Role name	Role status	
AD trust agent	absent	
AD trust controller	absent	
CA server	enabled	

図6.14 Web UI でのサーバーロール

コマンドライン: サーバーロールの表示

ipa config-show コマンドを実行すると、すべての CA サーバー、NTP サーバー、および現行の CA 更新マスターが表示されます。

```
$ ipa config-show
...
IPA masters: server1.example.com, server2.example.com,
server3.example.com
IPA CA servers: server1.example.com, server2.example.com
IPA NTP servers: server1.example.com, server2.example.com,
server3.example.com
IPA CA renewal master: server1.example.com
```

ipa server-show コマンドを実行すると、特定サーバーで有効になっているロール一覧が表示されます。たとえば、*server.example.com* で有効になっているロールを表示するには、以下を実行します。

```
$ ipa server-show
Server name: server.example.com
...
Enabled server roles: CA server, DNS server, NTP server, KRA server
```

ipa server-find --servrole は、特定のサーバーロールが有効になっているサーバーを検索します。たとえば、CA サーバーを検索するには、以下を実行します。

```
$ ipa server-find --servrole "CA server"
-----
2 IPA servers matched
-----
Server name: server1.example.com
...

Server name: server2.example.com
...
-----
Number of entries returned 2
-----
```

6.5.2. レプリカのマスター CA サーバーへのプロモート



注記

本セクションでは、ドメインレベル 1 の CA 更新マスターの変更について説明しています (7章 [ドメインレベルの表示と引き上げ](#) を参照)。ドメインレベル 0 での CA 更新マスターの変更については、「[レプリカのマスター CA サーバーへのプロモート](#)」を参照してください。

複数のレプリカがあるトポロジーでは、そのうちの 1 つがマスター CA サーバーとして機能し、CA サブシステムの証明書の更新を管理したり、証明書失効リスト (CRL) を生成したりします。デフォルトでは、レプリカを作成する元となる最初のサーバーがマスター CA となります。

マスター CA サーバーをオフラインにする、または使用停止にする場合は、別の CA サーバーをプロモートして、新規 CA 更新マスターとします。

1. レプリカが CA サブシステムの証明書更新を処理するよう設定します。
 - ドメインレベル 1 については、「[現行 CA 更新マスターの変更](#)」を参照してください。
 - ドメインレベル 0 については、「[証明書更新を処理するサーバーの変更](#)」を参照してください。
2. レプリカが CRL を生成するように設定します。「[CRL を生成するサーバーの変更](#)」を参照してください。
3. 今までのマスター CA サーバーの使用を停止する前に、新規マスターが正常に機能することを確認します。「[新規マスター CA サーバーの設定確認](#)」を参照してください。

6.5.2.1. 現行 CA 更新マスターの変更

Web UI: 現行 CA 更新マスターの変更

1. **IPA Server** → **Configuration** を選択します。
2. **IPA CA renewal master** フィールドで、新規 CA 更新マスターを選択します。

コマンドライン: 現行 CA 更新マスターの変更

ipa config-mod --ca-renewal-master-server コマンドを使用します。

```
$ ipa config-mod --ca-renewal-master-server
new_ca_renewal_master.example.com
...
IPA masters: old_ca_renewal_master.example.com,
new_ca_renewal_master.example.com
IPA CA servers: old_ca_renewal_master.example.com,
new_ca_renewal_master.example.com
IPA NTP servers: old_ca_renewal_master.example.com,
new_ca_renewal_master.example.com
IPA CA renewal master: new_ca_renewal_master.example.com
```

出力で更新が成功したことを確認します。

6.5.2.2. CRL を生成するサーバーの変更

CRL を生成するサーバーを変更するには、現行の CRL 生成マスターでの CRL 生成を停止し、それから他のサーバーで生成を有効にします。

現行 CRL 生成マスターの特定

CA がインストールされている各サーバーで `/etc/pki/pki-tomcat/ca/CS.cfg` ファイルをチェックします。

- CRL 生成マスターで、**ca.crl.MasterCRL.enableCRLUpdates** パラメーターを **true** に設定します。

```
# grep ca.crl.MasterCRL.enableCRLUpdates /etc/pki/pki-  
tomcat/ca/CS.cfg  
ca.crl.MasterCRL.enableCRLUpdates=true
```

- CRL 生成クローンでパラメーターを **false** に設定します。

現行 CRL 生成マスターで CRL 生成を停止する

1. CA サービスを停止します。

```
# systemctl stop pki-tomcatd@pki-tomcat.service
```

2. サーバー上での CRL 生成を無効にします。`/etc/pki/pki-tomcat/ca/CS.cfg` ファイルを開いて、**ca.crl.MasterCRL.enableCRLCache** と **ca.crl.MasterCRL.enableCRLUpdates** のパラメーターの値を **false** に設定します。

```
ca.crl.MasterCRL.enableCRLCache=false  
ca.crl.MasterCRL.enableCRLUpdates=false
```

3. CA サービスを起動します。

```
# systemctl start pki-tomcatd@pki-tomcat.service
```

4. CRL リクエストを新規マスターにリダイレクトするように Apache を設定します。`/etc/httpd/conf.d/ipa-pki-proxy.conf` ファイルを開いて、**RewriteRule** 引数をコメント解除します。

```
# Only enable this on servers that are not generating a CRL  
RewriteRule ^/ipa/crl/MasterCRL.bin  
https://server.example.com/ca/ee/ca/getCRL?  
op=getCRL&crlIssuingPoint=MasterCRL [L,R=301,NC]
```

5. Apache を再起動します。

```
# systemctl restart httpd.service
```

この手順の前までは、このサーバーは CRL リクエストに対応していましたが、これですべての CRL リクエストが以前の CA マスターにルーティングされます。

サーバーが CRL を生成するように設定する

1. CA サービスを停止します。

```
# systemctl stop pki-tomcatd@pki-tomcat.service
```

2. サーバー上での CRL 生成を有効にします。**ca.crl.MasterCRL.enableCRLCache** と **ca.crl.MasterCRL.enableCRLUpdates** のパラメーターの値を **true** に設定します。

```
ca.crl.MasterCRL.enableCRLCache=true
ca.crl.MasterCRL.enableCRLUpdates=true
```

3. CA サービスを起動します。

```
# systemctl start pki-tomcatd@pki-tomcat.service
```

4. Apache で CRL リクエストのリダイレクトを無効にします。**/etc/httpd/conf.d/ipa-pki-proxy.conf** ファイルを開いて、**RewriteRule** 引数をコメントアウトします。

```
#RewriteRule ^/ipa/crl/MasterCRL.bin
https://server.example.com/ca/ee/ca/getCRL?
op=getCRL&crlIssuingPoint=MasterCRL [L,R=301,NC]
```

この手順の前までは、全 CRL リクエストは以前の CA マスターにルーティングされていましたが、これでこのサーバーが CRL リクエストに対応するようになりました。

5. Apache を再起動します。

```
# systemctl restart httpd.service
```

6.5.2.3. 新規マスター CA サーバーの設定確認

/var/lib/ipa/pki-ca/publish/MasterCRL.bin ファイルが新規マスター CA サーバーにあることを確認します。

このファイルは、**/etc/pki/pki-tomcat/ca/CS.cfg** ファイルで定義されている間隔を基に、**ca.crl.MasterCRL.autoUpdateInterval** パラメーターを使って生成されます。デフォルト値は、240 分 (4 時間) です。

このファイルが存在すれば、新規マスター CA サーバーは正常に設定されているので、以前の CA マスターシステムを安全に閉鎖できます。

第7章 ドメインレベルの表示と引き上げ

ドメインレベルは、IdM トポロジー内で利用可能な操作と機能を示します。

ドメインレベル 1

利用可能な機能の例

- 簡素化された **ipa-replica-install** ([「レプリカの作成: 概要」](#) を参照)
- 機能強化されたトポロジー管理 ([6章 レプリケーショントポロジーの管理](#) を参照)



重要

ドメインレベル 1 は、Red Hat Enterprise Linux 7.3 の IdM バージョン 4.4 で導入されました。ドメインレベル 1 の機能を使用するには、すべてのレプリカで Red Hat Enterprise Linux 7.3 以降を稼働している必要があります。

最初のサーバーに Red Hat Enterprise Linux 7.3 がインストールされると、ドメインレベルは自動的に 1 に設定されます。

すべてのサーバーを IdM の以前のバージョンから 4.4 にアップグレードしても、ドメインレベルは自動的に上げられません。ドメインレベル 1 の機能を使用するには、[「ドメインレベルの引き上げ」](#) に記載の手順に従ってドメインレベルを手動で上げる必要があります。

ドメインレベル 0

利用可能な機能の例

- **ipa-replica-install** では、初期サーバー上でレプリカ情報ファイルを作成し、これをレプリカにコピーするという複雑なプロセスが必要になります ([「レプリカの作成」](#) を参照)。
- **ipa-replica-manage** および **ipa-csreplica-manage** を使用したより複雑でエラーが発生しやすいトポロジー管理 ([「レプリカとレプリカ合意の管理」](#) を参照)。

7.1. 現行ドメインレベルの表示

コマンドライン: 現行ドメインレベルの表示

1. 管理者としてログインします。

```
$ kinit admin
```

2. **ipa domainlevel-get** コマンドを実行します。

```
$ ipa domainlevel-get
-----
Current domain level: 0
-----
```

Web UI: 現行ドメインレベルの表示

IPA Server → Domain Level を選択します。

7.2. ドメインレベルの引き上げ



重要

この操作は元に戻すことができません。ドメインレベルを **0** から **1** に引き上げると、**1** から **0** にダウングレードすることはできません。

コマンドライン: ドメインレベルの引き上げ

1. 管理者としてログインします。

```
$ kinit admin
```

2. **ipa domainlevel-set** コマンドが必要なレベルを提供して実行します。

```
$ ipa domainlevel-set 1
-----
Current domain level: 1
-----
```

Web UI: ドメインレベルの引き上げ

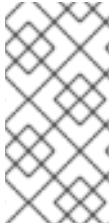
1. **IPA Server** → **Domain Level** を選択します。
2. **Set Domain Level** をクリックします。

第8章 IDENTITY MANAGEMENT の更新と移行

8.1. IDENTITY MANAGEMENT の更新

システム上の Identity Management パッケージの更新には、**yum** ユーティリティを使用します。

また、7.3 などの新規の Red Hat Enterprise Linux マイナーバージョンが利用可能な場合は、**yum** は Identity Management サーバーやクライアントをこのバージョンにアップグレードします。

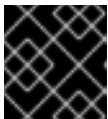


注記

本セクションでは、Red Hat Enterprise Linux 6 から Red Hat Enterprise Linux 7 への Identity Management の移行は説明していません。この移行については、[「Red Hat Enterprise Linux 6 からバージョン 7 への Identity Management の移行」](#)を参照してください。

8.1.1. Identity Management 更新 の注意点

- 少なくとも 1 台のサーバーで Identity Management パッケージを更新したら、トポロジーないの他のサーバーでパッケージを更新していなくても、これらのサーバーは更新されたスキーマを受信します。これにより、新スキーマを使用する新規エントリーを他のサーバー間で複製することが可能になります。
- Identity Management パッケージのダウングレードはサポートされていません。



重要

ipa-* パッケージには **yum downgrade** コマンドを実行しないでください。

- Red Hat では、次のバージョンへのアップグレードのみを推奨しています。たとえば、Red Hat Enterprise Linux 7.4 用の Identity Management にアップグレードする場合には、Red Hat Enterprise Linux 7.3 用の Identity Management からアップグレードすることを推奨します。それ以前のバージョンからのアップグレードでは、問題が発生する可能性があります。

8.1.2. yum を使った Identity Management パッケージの更新

サーバーまたはクライアント上の Identity Management パッケージすべてを更新するには、以下を実行します。

```
# yum update ipa-*
```



警告

複数の Identity Management サーバーをアップグレードする場合は、各アップグレードで少なくとも 10 分間の間隔をあけてください。

複数のサーバーで同時または間隔をあまりあけないでアップグレードを行うと、トポロジー全体でアップグレード後のデータ変更を複製する時間が足りず、複製イベントが競合する可能性があります。

関連情報

- **yum** ユーティリティの使用に関する詳細情報は、『システム管理者のガイド』の [Yum](#) を参照してください。

重要

[CVE-2014-3566](#) のため SSLv3 (Secure Socket Layer version 3) プロトコルは **mod_nss** モジュールで無効にする必要があります。次の手順に従い、無効になっていることを確認してください。

1. **/etc/httpd/conf.d/nss.conf** ファイルを編集して **NSSProtocol** パラメーターを **TLSv1.0** (後方互換用) および **TLSv1.1**、**TLSv1.2** に設定します。

```
NSSProtocol TLSv1.0,TLSv1.1,TLSv1.2
```

2. **httpd** サービスを再起動します。

```
# systemctl restart httpd.service
```

Red Hat Enterprise Linux 7 の Identity Management では、メインパッケージのアップグレードを行うため **yum update ipa-*** コマンドを起動すると上記の手順が自動的に行われます。

8.2. RED HAT ENTERPRISE LINUX 6 からバージョン 7 への IDENTITY MANAGEMENT の移行

本セクションでは、Red Hat Enterprise Linux 6 Identity Management から Red Hat Enterprise Linux 7 サーバーにデータおよび設定を移行する方法について説明します。この移行手順では以下を実行します。

- Red Hat Enterprise Linux 6 ベースの証明局 (CA) マスターサーバーの Red Hat Enterprise Linux 7 への移行。
- 新規 Red Hat Enterprise Linux 7 サーバーへの全サービスの移行。これらのサービスには、CRL と証明書の作成、DNS 管理、Kerberos KDC 管理などがあります。
- 元の Red Hat Enterprise Linux 6 CA マスターの停止。

手順では、以下を前提としています。

- **rhel7.example.com** は、新規 CA マスターとなる Red Hat Enterprise Linux 7 システムです。
- **rhel6.example.com** は、元の Red Hat Enterprise Linux 6 CA マスターです。



注記

どの Red Hat Enterprise Linux 6 サーバーがマスター CA サーバーかを特定するには、どのサーバー上で **certmonger** サービスが **renew_ca_cert** コマンドを追跡しているかを判定します。このコマンドをすべての Red Hat Enterprise Linux 6 サーバーで実行します。

```
[root@rhel6 ~]# getcert list -d /var/lib/pki-ca/alias -n
"subsystemCert cert-pki-ca" | grep post-save
post-save command:
/usr/lib64/ipa/certmonger/renew_ca_cert "subsystemCert
cert-pki-ca"
```

renew_ca_cert を実行する post-save アクションは、CA マスターにのみ定義されています。

8.2.1. Identity Management の Red Hat Enterprise Linux 6 から 7 への移行の前提条件

- **rhel6.example.com** システムを Red Hat Enterprise Linux 6 の最新バージョンに更新します。
- **rhel6.example.com** システムで、ipa-* パッケージをアップグレードします。

```
[root@rhel6 ~]# yum update ipa-*
```

このステップにより、[RHBA-2015:0231-2](#) アドバイザリーの適用も確認されます。これは、bind-dyndb-ldap パッケージの **2.3-6.el6_6** バージョンを提供するもので、Red Hat Enterprise Linux 6.6 Extended Update Support (EUS) と利用できます。



警告

bind-dyndb-ldap の以前のバージョンを使用すると、Red Hat Enterprise Linux 6.6 DNS サーバーと Red Hat Enterprise Linux 7 DNS サーバー間における DNS 正引きゾーンの動作に一貫性がなくなります。

- **rhel7.example.com** システムが「[サーバーインストールの前提条件](#)」および「[レプリカのインストールの前提条件](#)」にある条件を満たしていることを確認してください。
- **rhel7.example.com** システムに必要なパッケージをインストールしてください。詳細は、「[IdM サーバーのインストールに必要なパッケージ](#)」を参照してください。

8.2.2. Red Hat Enterprise Linux 6 上での Identity Management スキーマ

の更新

copy-schema-to-ca.py スキーマ更新スクリプトにより、**rhel6.example.com** での **rhel7.example.com** レプリカインストールを準備することができます。スキーマの更新は、Identity Management のバージョン 3.1 とそれ以降の間におけるスキーマの変更により必要となります。

1. **copy-schema-to-ca.py** スキーマ更新スクリプトを **rhel7.example.com** システムから **rhel6.example.com** システムにコピーします。

```
[root@rhel7 ~]# scp /usr/share/ipa/copy-schema-to-ca.py
root@rhel6:/root/
```

2. 更新された **copy-schema-to-ca.py** スクリプトを **rhel6.example.com** 上で実行します。

```
[root@rhel6 ~]# python copy-schema-to-ca.py
ipa          : INFO      Installed /etc/dirsrv/slapd-PKI-
IPA//schema/60kerberos.ldif
[... output truncated ...]
ipa          : INFO      Schema updated successfully
```

8.2.3. Red Hat Enterprise Linux 7 レプリカのインストール

1. **rhel6.example.com** システム上で、**rhel7.example.com** レプリカのインストールに使用するレプリカファイルを作成します。たとえば、**rhel7.example.com** 用のレプリカファイルを作成します。このシステムの IP アドレスは **192.0.2.1** とします。

```
[root@rhel6 ~]# ipa-replica-prepare rhel7.example.com --ip-address
192.0.2.1

Directory Manager (existing master) password:
Preparing replica for rhel7.example.com from rhel6.example.com
[... output truncated ...]
The ipa-replica-prepare command was successful
```

「[レプリカ情報ファイル](#)」と「[レプリカの作成](#)」も参照してください。

2. レプリカ情報ファイルを **rhel6.example.com** から **rhel7.example.com** にコピーします。

```
[root@rhel6 ~]# scp /var/lib/ipa/replica-info-
replica.example.com.gpg root@rhel7:/var/lib/ipa/
```

3. レプリカファイルを使用して **rhel7.example.com** レプリカをインストールします。たとえば、この例では以下のオプションを使用しています。

- **--setup-ca** は、Certificate System コンポーネントを設定します。
- **--setup-dns** と **--forwarder** は、統合 DNS サーバーとフォワーダーを設定します。
- **--ip-address** は、**rhel7.example.com** システムの IP アドレスを指定します。

```
[root@rhel7 ~]# ipa-replica-install /var/lib/ipa/replica-info-
rhel7.example.com.gpg --setup-ca --ip-address 192.0.2.1 --setup-dns
--forwarder 192.0.2.20
```

```
Directory Manager (existing master) password:
```

```
Checking DNS forwarders, please wait ...
Run connection check to master
[... output truncated ...]
Client configuration complete.
```

関連項目:

- 「[レプリカの作成](#)」では、レプリカ情報ファイルを使用したレプリカ作成について説明しています。
 - 「[統合 DNS 使用の判断](#)」および「[CA 設定の決定](#)」
4. Identity Management サービスが **rhel7.example.com** で稼働していることを確認します。

```
[root@rhel7 ~]# ipactl status
Directory Service: RUNNING
[... output truncated ...]
ipa: INFO: The ipactl command was successful
```

8.2.4. CA サービスの Red Hat Enterprise Linux 7 サーバーへの移行

作業開始前に、以下を実行してください。

- **rhel6.example.com** と **rhel7.example.com** の両方の CA がマスターサーバーとして設定されていることを確認します。

```
[root@rhel7 ~]$ kinit admin
[root@rhel7 ~]$ ipa-csreplica-manage list
rhel6.example.com: master
rhel7.example.com: master
```

レプリカ合意の詳細を表示するには、以下を実行します。

```
[root@rhel7 ~]# ipa-csreplica-manage list --verbose
rhel7.example.com
rhel7.example.com
last init status: None
last init ended: 1970-01-01 00:00:00+00:00
last update status: Error (0) Replica acquired successfully:
Incremental update succeeded
last update ended: 2017-02-13 13:55:13+00:00
```

rhel6.example.com の元のマスター CA で、CA サブシステムの証明書更新を停止します。

1. 元の CA 証明書の追跡を無効にします。

```
[root@rhel6 ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n
"auditSigningCert cert-pki-ca"
Request "20171127184547" removed.
[root@rhel6 ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n
"ocspSigningCert cert-pki-ca"
```

```
Request "20171127184548" removed.
[root@rhel6 ~]# getcert stop-tracking -d /var/lib/pki-ca/alias -n
"subsystemCert cert-pki-ca"
Request "20171127184549" removed.
[root@rhel6 ~]# getcert stop-tracking -d /etc/httpd/alias -n ipaCert
Request "20171127184550" removed.
```

2. **rhel6.example.com** を再設定して、新規マスター CA から更新された証明書を取得します。

- a. **certmonger** サービスディレクトリーに更新ヘルパースクリプトをコピーに、適切なパーミッションを設定します。

```
[root@rhel6 ~]# cp /usr/share/ipa/ca_renewal
/var/lib/certmonger/cas/
[root@rhel6 ~]# chmod 0600 /var/lib/certmonger/cas/ca_renewal
```

- b. SELinux 設定を更新します。

```
[root@rhel6 ~]# restorecon /var/lib/certmonger/cas/ca_renewal
```

- c. **certmonger** を再起動します。

```
[root@rhel6 ~]# service certmonger restart
```

- d. CA が証明書を取得しているかチェックします。

```
[root@rhel6 ~]# getcert list-cas
...
CA 'dogtag-ipa-retrieve-agent-submit':
    is-default: no
    ca-type: EXTERNAL
    helper-location: /usr/libexec/certmonger/dogtag-ipa-retrieve-
agent-submit
```

- e. CA 証明書データベース PIN を取得します。

```
[root@rhel6 ~]# grep internal= /var/lib/pki-ca/conf/password.conf
```

- f. **certmonger** が外部更新用の証明書を追跡するよう設定します。これには、データベース PIN が必要です。

```
[root@rhel6 ~]# getcert start-tracking \
-c dogtag-ipa-retrieve-agent-submit \
-d /var/lib/pki-ca/alias \
-n "auditSigningCert cert-pki-ca" \
-B /usr/lib64/ipa/certmonger/stop_pkicad \
-C '/usr/lib64/ipa/certmonger/restart_pkicad \
"auditSigningCert cert-pki-ca"' \
-T "auditSigningCert cert-pki-ca" \
-P database_pin
New tracking request "20171127184743" added.
[root@rhel6 ~]# getcert start-tracking \
-c dogtag-ipa-retrieve-agent-submit \
```



```

-d /var/lib/pki-ca/alias \
-n "ocspSigningCert cert-pki-ca" \
-B /usr/lib64/ipa/certmonger/stop_pkicad \
-C '/usr/lib64/ipa/certmonger/restart_pkicad \
"ocspSigningCert cert-pki-ca" \
-T "ocspSigningCert cert-pki-ca" \
-P database_pin
New tracking request "20171127184744" added.
[root@rhel6 ~]# getcert start-tracking \
-c dogtag-ipa-retrieve-agent-submit \
-d /var/lib/pki-ca/alias \
-n "subsystemCert cert-pki-ca" \
-B /usr/lib64/ipa/certmonger/stop_pkicad \
-C '/usr/lib64/ipa/certmonger/restart_pkicad \
"subsystemCert cert-pki-ca" \
-T "subsystemCert cert-pki-ca" \
-P database_pin
New tracking request "20171127184745" added.
[root@rhel6 ~]# getcert start-tracking \
-c dogtag-ipa-retrieve-agent-submit \
-d /etc/httpd/alias \
-n ipaCert \
-C /usr/lib64/ipa/certmonger/restart_httpd \
-T ipaCert \
-p /etc/httpd/alias/pwdfile.txt
New tracking request "20171127184746" added.

```

CRL 生成を元の **rhel6.example.com** CA マスターから **rhel7.example.com** に移動します。

1. **rhel6.example.com** で CRL 生成を停止します。

- a. CA サービスを停止します。

```
[root@rhel6 ~]# service pki-cad stop
```

- b. **rhel6.example.com** 上での CRL 生成を無効にします。**/var/lib/pki-ca/conf/CS.cfg** ファイルを開いて、**ca.crl.MasterCRL.enableCRLCache** および **ca.crl.MasterCRL.enableCRLUpdates** のパラメーターの値を **false** に設定します。

```
ca.crl.MasterCRL.enableCRLCache=false
ca.crl.MasterCRL.enableCRLUpdates=false
```

- c. CA サービスを起動します。

```
[root@rhel6 ~]# service pki-cad start
```

2. **rhel6.example.com** で、Apache が CRL リクエストを新しいマスターである **rhel7.example.com** にリダイレクトするよう設定します。

- a. **/etc/httpd/conf.d/ipa-pki-proxy.conf** ファイルを開き、**RewriteRule** 引数のコメントを解除します。サーバーホスト名をサーバー URL にある **rhel7.example.com** ホスト名で置き換えます。

```
RewriteRule ^/ipa/crl/MasterCRL.bin
https://rhel7.example.com/ca/ee/ca/getCRL?
op=getCRL&crlIssuingPoint=MasterCRL [L,R=301,NC]
```

- b. Apache を再起動します。

```
[root@rhel6 ~]# service httpd restart
```

3. **rhel7.example.com** で **rhel7.example.com** を新規 CA マスターとして設定します。

- a. 「[証明書更新を処理するサーバーの変更](#)」にあるように、**rhel7.example.com** が CA サブシステムの証明書更新を処理するように設定します。
- b. 「[サーバーが CRL を生成するように設定する](#)」にあるように、**rhel7.example.com** が証明書失効リスト (CRL) を生成するように設定します。

関連情報

- CA サブシステム証明書更新および CRL についての詳細は、「[レプリカのマスター CA サーバーへのプロモート](#)」を参照してください。

8.2.5. Red Hat Enterprise Linux 6 サーバーの停止

rhel6.example.com 上の全サービスを停止して、新規 **rhel7.example.com** サーバーへのドメイン検索を実施します。

```
[root@rhel6 ~]# ipactl stop
Stopping CA Service
Stopping pki-ca: [ OK ]
Stopping HTTP Service
Stopping httpd: [ OK ]
Stopping MEMCACHE Service
Stopping ipa_memcached: [ OK ]
Stopping DNS Service
Stopping named: . [ OK ]
Stopping KPASSWD Service
Stopping Kerberos 5 Admin Server: [ OK ]
Stopping KDC Service
Stopping Kerberos 5 KDC: [ OK ]
Stopping Directory Service
Shutting down dirsrv:
    EXAMPLE-COM... [ OK ]
    PKI-IPA... [ OK ]
```

この後に **ipa** ユーティリティーを使用すると、Remote Procedure Call (RPC) で新規サーバーに接続します。

8.2.6. マスター CA サーバー移行後のステップ

トポロジー内の各 Red Hat Enterprise Linux 6 サーバーで以下を実行します。

1. **rhel7.example.com** からレプリカファイルを作成します。



注記

Red Hat Enterprise Linux 6 サーバーから Red Hat Enterprise Linux 7 レプリカをインストールした後は、Identity Management ドメインのドメインレベルは自動的に 0 に設定されます。

Red Hat Enterprise Linux 7.3 では、レプリカを容易にインストール、管理する方法が導入されています。これらの機能を使用するには、トポロジーのドメインレベルが 1 である必要があります。詳細は [7章 ドメインレベルの表示と引き上げ](#) を参照してください。

2. レプリカファイルを使用して別の Red Hat Enterprise Linux 7 システムに新規レプリカをインストールします。

[4章 Identity Management のレプリカのインストールとアンインストール](#) を参照してください。

Red Hat Enterprise Linux 6 サーバーの使用を停止するには、

- Red Hat Enterprise Linux 7 サーバー上で削除コマンドを実行して、トポロジーからサーバーを削除します。

[「IdM サーバーのアンインストール」](#) を参照してください。

第9章 IDENTITY MANAGEMENT のバックアップと復元

Red Hat Enterprise Linux Identity Management は、たとえばサーバーが正常に稼働しなくなった場合やデータを損失した場合などのために、IdM システムのバックアップと復元を手動で行うソリューションを提供しています。バックアップ時には、システムは IdM セットアップに関する情報を含むディレクトリーを作成して保存します。復元時には、このバックアップディレクトリーを使って元の IdM セットアップに戻すことができます。

重要

失われたレプリカを残りのサーバーのレプリカとして再インストールして、デプロイメントの残りのサーバーから IdM サーバークループの損失部分を再構築できない場合にのみ、本章に記載のバックアップおよび復元の手順を使用するようにしてください。

["Backup and Restore in IdM/IPA" Knowledgebase solution](#) では、複数のサーバーレプリカを維持することで損失を避ける方法について説明しています。既存のレプリカから同一データを使って再構築する方法が望ましいやり方です。バックアップバージョンには通常古い情報が含まれるので、無効になっている可能性があるからです。

バックアップと復元で回避できる脅威のシナリオには、以下のようなものがあります。

- マシンの上で壊滅的なハードウェア障害が発生し、マシンがそれ以上機能しなくなった場合。このような場合には、オペレーティングシステムを最初から再インストールして、同じ完全修飾ドメイン名 (FQDN) とホスト名でマシンを設定し、IdM パッケージと元のシステムにあった IdM に関連するその他すべてのオプションパッケージをインストールして、IdM サーバーの全バックアップを復元します。
- 分離されているマシンでのアップグレードが失敗した場合。オペレーティングシステムは機能しているものの、IdM データが破損しているため、IdM システムを正常起動時構成に戻す場合です。

重要

上記の 2 例のようにハードウェアの障害やアップグレードで失敗した際に、唯一の証明局 (CA) など、特別なロールが割り当てられたレプリカがなくなった場合や、すべてのレプリカがなくなった場合にのみバックアップから復元するようにしてください。同一データを持つ別のレプリカがまだある場合は、失われたレプリカを削除して、残りのレプリカから再構成することが推奨されます。

- エントリーが削除されてしまい、それらを戻す場合など、LDAP コンテナに望ましくない変更がされた場合。バックアップされた LDAP データを復元すると、IdM システム自体には影響せずに LDAP エントリーを元の状態に戻すことができます。

復元されたサーバーは、IdM の唯一の情報ソースになります。他のマスターサーバーは復元されたサーバーから再度初期化されます。最後のバックアップが実行された後に作成されたデータは失われます。このため、通常のシステムメンテナンスには、バックアップと復元の方法を使用すべきではありません。可能な場合は常に失われたサーバーをレプリカとして再インストールすることで再構成を行ってください。

バックアップおよび復元機能はコマンドラインからのみ操作が可能で、IdM Web UI では操作できません。

9.1. 完全なサーバーバックアップおよびデータのためのバックアップ

IdM は以下の 2 つのバックアップオプションを提供しています。

完全な IdM サーバーバックアップ

完全なサーバーバックアップでは、スタンドアロンバックアップとなる LDAP データのほかに、すべての IdM サーバーファイルのバックアップコピーが作成されます。IdM は数百のファイルに影響を及ぼします。バックアッププロセスでコピーされるファイルはディレクトリー全体と設定ファイルやログファイルなどの特定ファイルを合わせたもので、IdM に直接関連するものと、IdM が依存する様々なサービスに関連します。完全なサーバーバックアップは生ファイルのバックアップなので、これはオフラインで実行されます。完全なサーバーバックアップを実行するスクリプトは、IdM サービスすべてを停止して、バックアッププロセスの安全な実行を確保します。

完全なサーバーバックアップでコピーされるファイルとディレクトリーの全一覧は、[「バックアップ中にコピーされるディレクトリーおよびファイル一覧」](#) を参照してください。

データのためのバックアップ

データのためのバックアップでは、LDAP データのバックアップコピーと changelog のみが作成されます。このプロセスでは **IPA-REALM** インスタンスのバックアップが作成され、さらに複数または単一のバックエンドをバックアップすることができます。バックエンドには、**IPA** バックエンドと **CA Dogtag** バックエンドが含まれます。このタイプのバックエンドは、LDIF (LDAP データ交換形式) で保存されている LDAP コンテンツのレコードもバックアップします。データのためのバックアップは、オフラインでもオンラインでも実行できます。

デフォルトでは、IdM は作成されたバックアップを `/var/lib/ipa/backup/` ディレクトリーに保存します。バックアップを含んでいるサブディレクトリーの命名規則は以下のとおりです。

- 完全なサーバーバックアップの場合は、GMT のタイムゾーンで **ipa-full-YEAR-MM-DD-HH-MM-SS** となります。
- データのためのバックアップの場合は、GMT のタイムゾーンで **ipa-data-YEAR-MM-DD-HH-MM-SS** となります。

9.1.1. バックアップの作成

完全なサーバーバックアップもデータのためのバックアップも、**ipa-backup** ユーティリティーを使用して作成されます。これは常に root で実行する必要があります。

完全なサーバーバックアップを作成するには、**ipa-backup** を実行します。



重要

完全なサーバーバックアップのプロセスはオフラインで実行する必要があるため、すべての IdM サービスが停止されます。バックアップが終了すると、IdM サービスは再開します。

データのためのバックアップを作成するには、**ipa-backup --data** コマンドを実行します。

ipa-backup には以下のオプションを追加することができます。

- **--online** は、オンラインでのバックアップを実行します。これはデータ専用のバックアップでのみ利用可能となります。
- **--logs** は、バックアップに IdM サービスログファイルを含めます。

ipa-backup の使用に関する詳細情報は、ipa-backup(1) man ページを参照してください。

9.1.2. バックアップの暗号化

IdM バックアップは、GNU Privacy Guard (GPG) を使って暗号化することができます。

GPG キーを作成するには、以下を実行します。

1. たとえば、**cat >keygen <<EOF** を実行して、キーの詳細を含む **keygen** ファイルを作成します。そして、コマンドラインから必要となる暗号化詳細をファイルに提供します。

```
[root@server ~]# cat >keygen <<EOF
> %echo Generating a standard key
> Key-Type: RSA
> Key-Length:2048
> Name-Real: IPA Backup
> Name-Comment: IPA Backup
> Name-Email: root@example.com
> Expire-Date: 0
> %pubring /root/backup.pub
> %secring /root/backup.sec
> %commit
> %echo done
> EOF
[root@server ~]#
```

2. **backup** という名前のキーペアを新規生成し、コマンド **keygen** のコンテンツを提供します。以下のコマンドでは、**/root/backup.sec** および **/root/backup.pub** というパス名のキーペアが生成されます。

```
[root@server ~]# gpg --batch --gen-key keygen
[root@server ~]# gpg --no-default-keyring --secret-keyring
/root/backup.sec \
--keyring /root/backup.pub --list-secret-keys
```

GPG 暗号化されたバックアップを作成するには、以下のオプションを使って生成された**backup** キーを **ipa-backup** に渡します。

- **--gpg** は、**ipa-backup** に暗号化バックアップを実行するよう指示します。
- **--gpg-keyring=GPG_KEYRING** は、ファイル拡張子なしで GPG keyring への完全パスを提供します。

例を示します。

```
[root@server ~]# ipa-backup --gpg --gpg-keyring=/root/backup
```

注記

使用中のシステムが **gpg2** ユーティリティーを使って GPG キーを生成している場合、問題が発生する可能性があります。これは、**gpg2** が機能するには外部のプログラムを必要とするためです。この状況でコンソールのみからキーを生成するには、キーの生成前に **pinentry-program /usr/bin/pinentry-curses** の行を **.gnupg/gpg-agent.conf** ファイルに追加します。

9.1.3. バックアップ中にコピーされるディレクトリーおよびファイル一覧

ディレクトリー

```
/usr/share/ipa/html  
/root/.pki  
/etc/pki-ca  
/etc/pki/pki-tomcat  
/etc/sysconfig/pki  
/etc/httpd/alias  
/var/lib/pki  
/var/lib/pki-ca  
/var/lib/ipa/sysrestore  
/var/lib/ipa-client/sysrestore  
/var/lib/ipa/dnssec  
/var/lib/sss/pubconf/krb5.include.d/  
/var/lib/authconfig/last  
/var/lib/certmonger  
/var/lib/ipa  
/var/run/dirsrv  
/var/lock/dirsrv
```

ファイル

```
/etc/named.conf  
/etc/named.keytab  
/etc/resolv.conf  
/etc/sysconfig/pki-ca  
/etc/sysconfig/pki-tomcat  
/etc/sysconfig/dirsrv  
/etc/sysconfig/ntpd  
/etc/sysconfig/krb5kdc  
/etc/sysconfig/pki/ca/pki-ca  
/etc/sysconfig/ipa-dnskeysyncd  
/etc/sysconfig/ipa-ods-exporter  
/etc/sysconfig/named  
/etc/sysconfig/ods  
/etc/sysconfig/authconfig  
/etc/ipa/nssdb/pwdfile.txt  
/etc/pki/ca-trust/source/ipa.p11-kit  
/etc/pki/ca-trust/source/anchors/ipa-ca.crt  
/etc/nsswitch.conf  
/etc/krb5.keytab  
/etc/sss/sss.conf  
/etc/openldap/ldap.conf  
/etc/security/limits.conf  
/etc/httpd/conf/password.conf  
/etc/httpd/conf/ipa.keytab  
/etc/httpd/conf.d/ipa-pki-proxy.conf  
/etc/httpd/conf.d/ipa-rewrite.conf  
/etc/httpd/conf.d/nss.conf  
/etc/httpd/conf.d/ipa.conf  
/etc/ssh/sshd_config  
/etc/ssh/ssh_config  
/etc/krb5.conf  
/etc/ipa/ca.crt  
/etc/ipa/default.conf  
/etc/dirsrv/ds.keytab
```

```

/etc/ntp.conf
/etc/samba/smb.conf
/etc/samba/samba.keytab
/root/ca-agent.p12
/root/cacert.p12
/var/kerberos/krb5kdc/kdc.conf
/etc/systemd/system/multi-user.target.wants/ipa.service
/etc/systemd/system/multi-user.target.wants/sss.service
/etc/systemd/system/multi-user.target.wants/certmonger.service
/etc/systemd/system/pki-tomcatd.target.wants/pki-tomcatd@pki-
tomcat.service
/var/run/ipa/services.list
/etc/openssl/conf.xml
/etc/openssl/kasp.xml
/etc/ipa/dnssec/softsm2.conf
/etc/ipa/dnssec/softsm_pin_so
/etc/ipa/dnssec/ipa-ods-exporter.keytab
/etc/ipa/dnssec/ipa-dnskeysyncd.keytab
/etc/idm/nssdb/cert8.db
/etc/idm/nssdb/key3.db
/etc/idm/nssdb/secmod.db
/etc/ipa/nssdb/cert8.db
/etc/ipa/nssdb/key3.db
/etc/ipa/nssdb/secmod.db

```

ログファイルおよびディレクトリー

```

/var/log/pki-ca
/var/log/pki/
/var/log/dirsrv/slapd-PKI-IPA
/var/log/httpd
/var/log/ipaserver-install.log
/var/log/kadmind.log
/var/log/pki-ca-install.log
/var/log/messages
/var/log/ipaclient-install.log
/var/log/secure
/var/log/ipaserver-uninstall.log
/var/log/pki-ca-uninstall.log
/var/log/ipaclient-uninstall.log
/var/named/data/named.run

```

9.2. バックアップの復元

ipa-backup を使って作成されたバックアップのディレクトリーがあれば、IdM サーバーまたは LDAP コンテンツをバックアップ実行時の状態に復元することができます。バックアップが作成された元のホストとは異なるホスト上では、バックアップの復元はできません。



注記

IdM サーバーをアンインストールしても、自動的にこのサーバーのバックアップは削除されません。

9.2.1. 完全なサーバーバックアップまたはデータのためのバックアップからの復元



重要

完全なサーバーの復元前にサーバーをアンインストールすることが推奨されます。

完全なサーバーバックアップもデータ専用のバックアップも、**ipa-restore** ユーティリティーを使って復元されます。これは常に **root** で実行する必要があります。バックアップをコマンドに渡します。

- バックアップがデフォルトの **/var/lib/ipa/backup/** ディレクトリーにある場合は、ディレクトリー名のみを渡します。
- バックアップを含んでいるディレクトリーがデフォルト以外の場所にある場合は、完全パスを渡します。例を示します。

```
[root@server ~]# ipa-restore /path/to/backup
```

ipa-restore ユーティリティーはディレクトリーに含まれているバックアップのタイプを自動的に検出し、デフォルトで同一タイプの復元を実行します。

ipa-restore には以下のオプションを追加できます。

- **--data** は、完全なサーバーバックアップからデータのみを復元を実行します。つまり、完全なサーバーバックアップを含むディレクトリーから LDAP データのコンポーネントのみを復元します。
- **--online** は、データのみで LDAP データをオンラインで復元します。
- **--instance** は、どの 389 DS インスタンスを復元するか指定します。Red Hat Enterprise Linux 7 の IdM は **IPA-REALM** インスタンスしか使用しませんが、たとえば、別のインスタンスを持つシステム上でバックアップを作成することは可能です。この場合、**--instance** を使うと、**IPA-REALM** の復元のみが許可されます。例を示します。

```
[root@server ~]# ipa-restore --instance=IPA-REALM /path/to/backup
```

このオプションは、データのみを復元実行時にのみ使用できます。

- **--backend** は復元するバックエンドを指定します。このオプションがない場合は、**ipa-restore** は発見したすべてのバックエンドを復元します。可能性のあるバックエンドを定義する引数は **userRoot** でこれは IPA データバックエンドを復元し、**ipaca** の場合は CA バックエンドを復元します。

このオプションは、データのみを復元実行時にのみ使用できます。

- **--no-logs** は、ログファイルなしでバックアップを復元します。

IdM master で認証の問題を回避するには、リストア後に SSSD のキャッシュを消去してください。

1. SSSD サービスを停止します。

```
[root@server ~]# systemctl stop sssd
```

2. SSSD からキャッシュされたコンテンツをすべて削除します。

```
[root@server ~]# find /var/lib/sss/ ! -type d | xargs rm -f
```

3. SSSD サービスを起動します。

```
[root@server ~]# systemctl start sssd
```



注記

バックアップから復元した後は、システムを再起動することが推奨されます。

ipa-restore 使用に関する詳細情報は、ipa-restore(1) man ページを参照してください。

9.2.2. 複数マスターサーバーでの復元

バックアップから復元すると復元されたサーバーが新規のデータマスターとして設定され、他のマスターすべてを再度初期化する必要があります。この初期化を行うには **ipa-replica-manage** コマンドを実行し、CA がインストールされているマスター上で **ipa-csreplica-manage** コマンドを実行します。

```
[root@server ~]# ipa-replica-manage re-initialize --  
from=restored_master_FQDN
```

復元時のレプリケーションと他のマシン上での復元に関する情報は、ipa-restore(1) man ページを参照してください。

9.2.3. 暗号化バックアップからの復元

GPG で暗号化されたバックアップからの復元を行うには、**--gpg-keyring** オプションを使って秘密鍵および公開鍵への完全パスを提供します。例を示します。

```
[root@server ~]# ipa-restore --gpg-keyring=/root/backup /path/to/backup
```

第10章 IDM ユーザーのアクセス制御の定義

アクセス制御は、誰がマシンやサービスまたはエントリーなどの特定のリソースにアクセスできるかや、どのような種類の操作の実行を許可されるかななどを定義する、セキュリティー機能セットです。Identity Management はいくつかのアクセス制御エリアを提供し、どのような種類のアクセスが許可されているか、誰に許可されているか、を明確にします。この一部として Identity Management は、ドメイン内のリソースに対するアクセス制御と、IdM 設定自体へのアクセス制御を区別します。

本章では、IdM サーバーおよび他の IdM ユーザーに対する IdM 内のユーザーに利用可能な異なる内部アクセス制御メカニズムを説明しています。

10.1. IDM エントリーのアクセス制御

アクセス制御は、他のユーザーやオブジェクトに対してユーザーが許可された操作についての権限やパーミッションを定義します。

Identity Management のアクセス制御構造は、標準の LDAP アクセス制御に基づいています。IdM サーバー内のアクセスは、バックエンドの Directory Server インスタンスに保存されている IdM ユーザーをベースとしており、このユーザーは、Directory Server インスタンスに LDAP エントリーとして保存されている他の IdM エンティティーにアクセスが許可されています。

アクセス制御指示 (ACI) には、以下の 3 つの部分があります。

Actor

これは、動作のパーミッションを付与されているエンティティーです。これはユーザーが誰かを定義し、1 日のある時間帯や特定のマシンに試行を制限するなど、オプションでバインドの試行に対して他の制限を必須とすることが可能なため、LDAP アクセス制御モデルでは**バインドルール**と呼ばれます。

Target

これは、Actor が許可されている操作を実行する対象のエントリーを定義します。

Operation type

最後の部分では、ユーザーが実行を許可されるアクションの種類を決定します。最も一般的な操作は、追加、削除、書き込み、読み取り、および検索です。Identity Management では、すべてのユーザーが暗示的に IdM ドメイン内のすべてのエントリーに対する読み取りおよび検索権限を付与されています。制限されるのは、パスワードや Kerberos キーなどの重要な属性のみです。匿名ユーザーは、**sudo** ルールやホストベースのアクセス制御など、セキュリティー関連の設定は読み取ることができません。

いかなる操作でもそれが試行されると、IdM クライアントはまずバインド操作の一部としてユーザーの認証情報を送信します。バックエンドの Directory Server はまずユーザー認証情報を、次にユーザーアカウントをチェックして、ユーザーが要求された操作を実行するパーミッションを持っているかどうかを確認します。

10.1.1. Identity Management におけるアクセス制御方法

アクセス制御ルールの実装をシンプルかつ明確にするために、Identity Management はアクセス制御の定義を以下の 3 つのカテゴリーに分けています。

セルフサービスルール

これは、ユーザーが自分のパーソナルエントリーで実行可能な操作を定義します。アクセス制御タイプは、エントリー内での属性への書き込みパーミッションのみを許可します。エントリー自体の

追加もしくは削除操作は許可されません。

委任ルール

委任ルールでは、特定のユーザーグループが別のユーザーグループ内のユーザーの特定属性に関して書き込み (編集) 操作を許可されます。セルフサービスルールのように、この形式のアクセス制御は特定の属性値の編集に制限されており、エントリー全体を追加したり削除する権限や特定されていない属性に対する制御を付与するものではありません。

ロールベースのアクセス制御

ロールベースのアクセス制御では特別のアクセス制御グループが作成され、このグループに IdM ドメイン内での全タイプのエントリーに対するより幅広い権限が付与されます。ロールには編集、追加、および削除の権限が付与されるので、選択された属性だけでなくエントリー全体に対する完全な制御が付与されます。

ロールのなかには既に作成され、Identity Management 内で使用可能なものもあります。ホストや automount 設定、netgroup、DNS 設定、および IdM 設定など、すべてのタイプのエントリーを特別な方法で管理するために、特別なロールを作成することもできます。

10.2. セルフサービス設定の定義

セルフサービスのアクセス制御ルールでは、エントリーがそれ自体で実行可能な操作を定義します。このルールでは、ユーザー (または他の IdM エンティティ) が自身のパーソナルエントリーで編集可能な属性のみを定義します。

デフォルトでは、以下の 3 つのセルフサービスルールがあります。

- パーソナルエントリー内の一般的な属性を編集するルール。これには、姓、名、電話番号、および住所などが含まれます。
- パーソナルパスワードを編集するルール。これには 2 つの Samba パスワード、Kerberos パスワード、および一般的なユーザーパスワードが含まれます。
- パーソナル SSH キーを管理するルール。

10.2.1. Web UI でのセルフサービスルールの作成

1. トップメニューで **IPA Server** タブを開き、**Self Service Permissions** サブタブを選択します。
2. セルフサービス ACI リストのトップにある **Add** をクリックします。

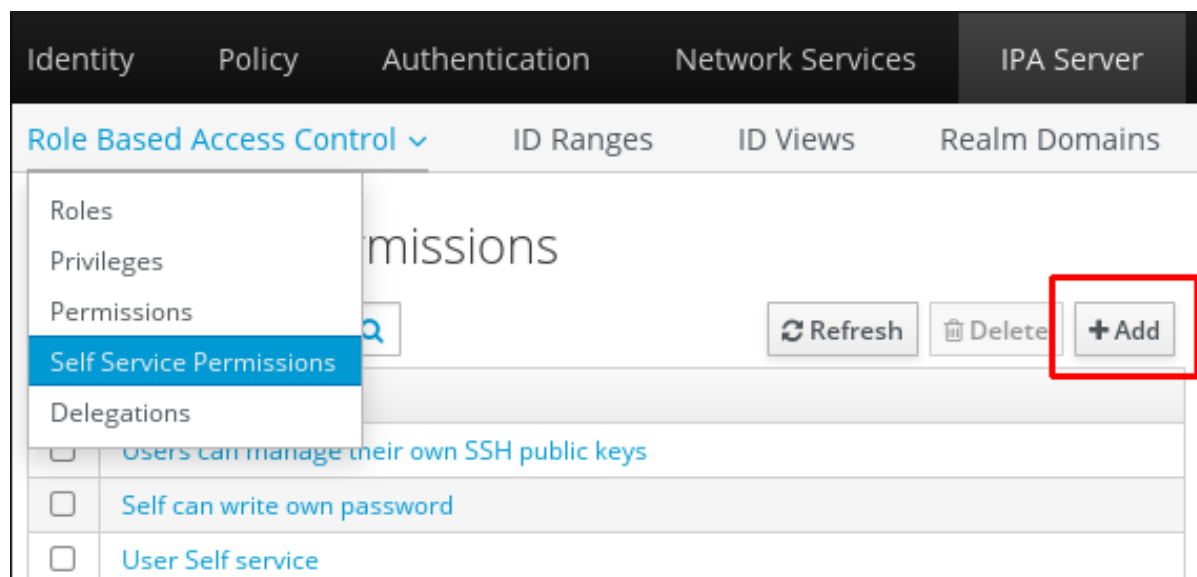


図10.1 新規セルフサービスルールの追加

3. ポップアップウィンドウでルール名を入力します。空白を使用することもできます。

図10.2 セルフサービス追加のフォーム

4. この ACI でユーザーによる編集を許可する属性のチェックボックスを選択します。
5. **Add** をクリックして新規セルフサービス ACI を保存します。

10.2.2. コマンドライン でのセルフサービスルールの作成

新規セルフサービスルールは、**selfservice-add** コマンドで追加できます。以下の 2 つのオプションが必須になります。

- **--permissions** は、ACI が付与する書き込み、追加、または削除などのパーミッションを設定します。
- **--attrs** では、この ACI がパーミッションを付与する属性の完全一覧を提供します。

```
[jsmith@server ~]$ ipa selfservice-add "Users can manage their own name
details" --permissions=write --attrs=givenname --attrs=displayname --
attrs=title --attrs=initials
-----
Added selfservice "Users can manage their own name details"
-----
    Self-service name: Users can manage their own name details
    Permissions: write
    Attributes: givenname, displayname, title, initials
```

10.2.3. セルフサービスルールの編集

ウェブ UI のセルフサービスエントリーでは、ACI に含まれている属性の一覧のみが編集可能な要素です。チェックボックスで選択または選択解除ができます。

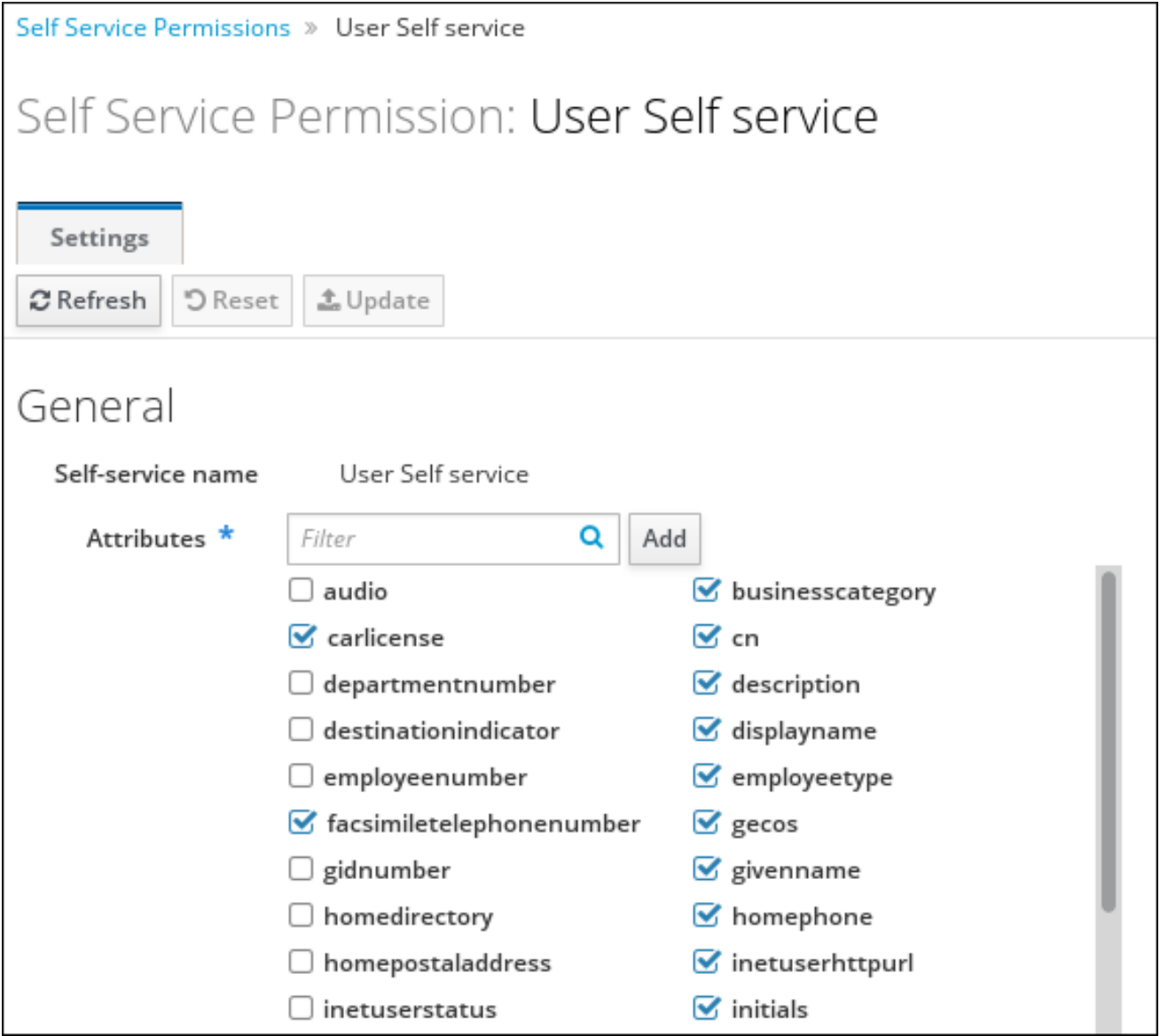


図10.3 セルフサービス編集ページ

コマンドラインでは、**ipa selfservice-mod** コマンドでセルフサービスルールを編集します。**--attrs** オプションはそれまでにサポートされていた属性をすべて上書きするので、新規属性の他に属性の完全一覧を常に含めるようにしてください。

```
[jsmith@server ~]$ ipa selfservice-mod "Users can manage their own name
details" --attrs=givenname --attrs=displayname --attrs=title --
attrs=initials --attrs=surname
-----
Modified selfservice "Users can manage their own name details"
-----
Self-service name: Users can manage their own name details
Permissions: write
Attributes: givenname, displayname, title, initials
```



重要

セルフサービスルールを修正する際は、既存の属性も含め、すべての属性を含めるようにしてください。

10.3. ユーザーへのパーミッションの委任

ユーザーのあるグループが別のユーザーのグループのエントリーを管理するパーミッションを割り当てられるという意味で、委任はロールにとってもよく似ています。ただし、付与される完全なアクセスがエントリー全体に対してではなく、特定のユーザー属性のみに対してであるという意味で、委任される権限はセルフサービスルールにより似ています。また、委任された権限内のグループは、アクセス制御のために特別に作成されたロールではなく、既存の IdM ユーザーグループになります。

10.3.1. Web UI でのユーザーグループへのアクセス委任

1. トップメニューで **IPA Server** タブを開き、**Delegations** サブタブを選択します。
2. 委任 ACI 一覧の上部にある **Add** をクリックします。

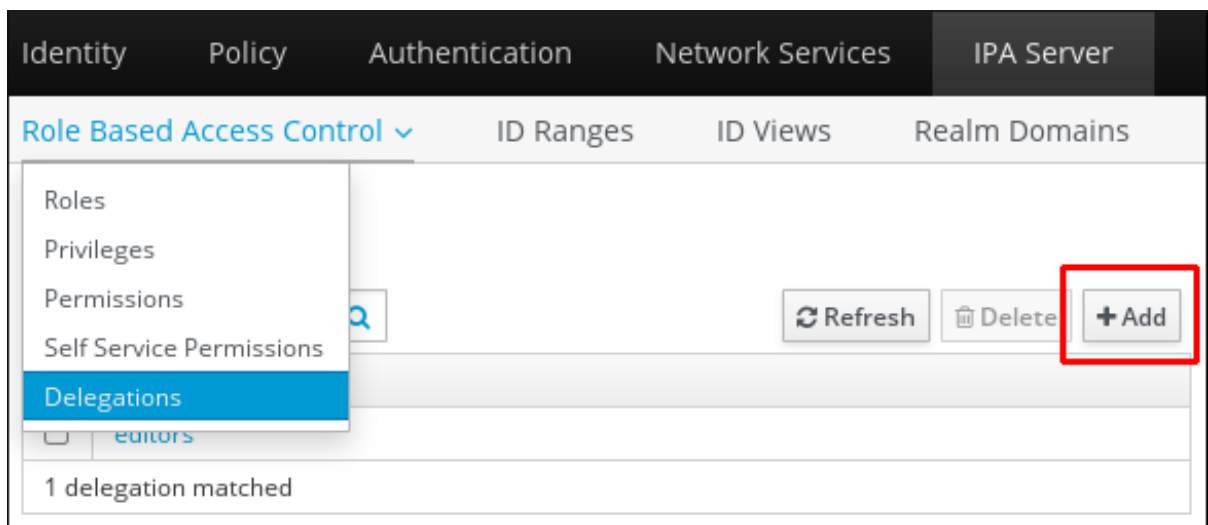


図10.4 新規委任の追加

3. 新規委任に名前を付けます。
4. ユーザーが特定の属性を閲覧する権限を持つ (read) かその属性を追加または変更する権限を持つ (write) かをチェックボックスで選択して、パーミッションを設定します。

ユーザーによっては情報を閲覧する必要はあるものの、編集可能にすべきでないユーザーもいます。

5. **User group** ドロップダウンメニューで、ユーザーグループのユーザーエントリーに **パーミッションを付与される グループ** を選択します。

Add Delegation

Delegation name *

Permissions ☒ read ☒ write

User group *

Member user group *

Attributes *

<input type="checkbox"/> audio	<input checked="" type="checkbox"/> businesscategory
<input checked="" type="checkbox"/> carlicense	<input type="checkbox"/> cn
<input checked="" type="checkbox"/> departmentnumber	<input type="checkbox"/> description
<input type="checkbox"/> destinationindicator	<input type="checkbox"/> displayname
<input type="checkbox"/> employeenumber	<input checked="" type="checkbox"/> employeeetype
<input checked="" type="checkbox"/> facsimiletelephonenumber	<input type="checkbox"/> gecos
<input type="checkbox"/> gidnumber	<input checked="" type="checkbox"/> givenname
<input type="checkbox"/> homedirectory	<input type="checkbox"/> homephone
<input type="checkbox"/> homepostaladdress	<input type="checkbox"/> inetuserhttpurl

* Required field

図10.5 委任追加のフォーム

6. **Member user group** ドロップダウンメニューで、委任グループのメンバーが編集する対象エントリーのグループを選択します。
7. 属性ボックスでは、メンバーのユーザーグループがパーミッションを付与される属性を選択します。
8. **Add** をクリックして新規委任 ACI を保存します。

10.3.2. コマンドラインでのユーザーグループへのアクセス委任

新規の委任アクセス制御ルールは、**delegation-add** コマンドで追加できます。以下の 3 つのオプションが必須になります。

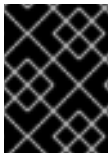
- **--group** は、ユーザーグループのユーザーエントリーにパーミッションを付与されるグループです。
- **--membergroup** は、委任グループのメンバーが編集する対象エントリーのグループです。
- **--attrs** は、メンバーグループのユーザーが編集を許可される属性です。

例を示します。

```
$ ipa delegation-add "basic manager attrs" --attrs=manager --attrs=title -
--attrs=employeetype --attrs=employeeenumber --group=engineering_managers --
membergroup=engineering
-----
Added delegation "basic manager attrs"
-----
Delegation name: basic manager attrs
Permissions: write
Attributes: manager, title, employeetype, employeeenumber
Member user group: engineering
User group: engineering_managers
```

委任ルールは、**delegation-mod** コマンドで編集します。**--attrs** オプションはそれまでにサポートされていた属性をすべて上書きするので、新規属性に加えて属性の完全一覧を常に含めるようにしてください。

```
[jsmith@server ~]$ ipa delegation-mod "basic manager attrs" --
attrs=manager --attrs=title --attrs=employeetype --attrs=employeeenumber --
attrs=displayname
-----
Modified delegation "basic manager attrs"
-----
Delegation name: basic manager attrs
Permissions: write
Attributes: manager, title, employeetype, employeeenumber, displayname
Member user group: engineering
User group: engineering_managers
```



重要

委任ルールを修正する際は、既存の属性も含め、すべての属性を含めるようにしてください。

10.4. ロールベースのアクセス制御の定義

ロールベースのアクセス制御では、セルフサービスおよび委任アクセス制御の場合とは非常に異なる種類の権限をユーザーに付与します。ロールベースのアクセス制御は基本的には管理業務用で、たとえばエントリーの追加、削除、または大幅な修正などができます。

ロールベースのアクセス制御は、以下の 3 つの部分で構成されます。

- **パーミッション**。これは、(読み取り、書き込み、追加、または削除などの) 特定の操作と、これらの操作が適用される IdM LDAP ディレクトリー内のターゲットエントリーを定義します。パーミッションはビルディングブロックで、必要に応じて複数の権限に割り当てることができます。

IdM パーミッションを使用すると、どのユーザーがどのオブジェクトにアクセスできるかや、それらのオブジェクトのどの属性にアクセスできるかという制御すら可能になります。IdM を使うと、個別の属性をブラックリスト化したりホワイトリスト化したりすることができるほか、全匿名ユーザー、全認証ユーザー、または権限のあるユーザーの特定グループのみに対するユーザー、グループ、または sudo といった特定の IdM 機能の視認性全体を変更することもできます。パーミッションに対するアプローチがこのように柔軟なことで、たとえば、管理者がユーザーやグループのアクセスをこれらのユーザーやグループが必要とする特定のセクションのみに限定し、他のセクションを完全に見えないようにする場合などに便利になります。

- **ロールで利用可能な 権限。**権限は、パーミッショングループに必須のものです。パーミッションは、ロールには直接適用されません。パーミッションは権限に追加され、そうすることで権限で完全かつ一貫性のあるアクセス制御ルールが作成されます。たとえば、automount の場所を追加、編集、削除するパーミッションを作成します。そしてそのパーミッションを FTP サービスを管理する別のパーミッションと組み合わせることで、ファイルシステム管理に関連する単一の権限を作成することができます。
- **ロール。**これは、権限で定義されたアクションの実行が可能な IdM ユーザーのリストになります。

完全に新しいパーミッションを作成したり、既存または新規のパーミッションをベースにして新たな権限を作成したりすることができます。

10.4.1. ロール

10.4.1.1. Web UI でのロールの作成

1. トップメニューで **IPA Server** タブを開き、**Role Based Access Control** サブタブを選択します。
2. Role Based ACI 一覧の上部にある **Add** をクリックします。

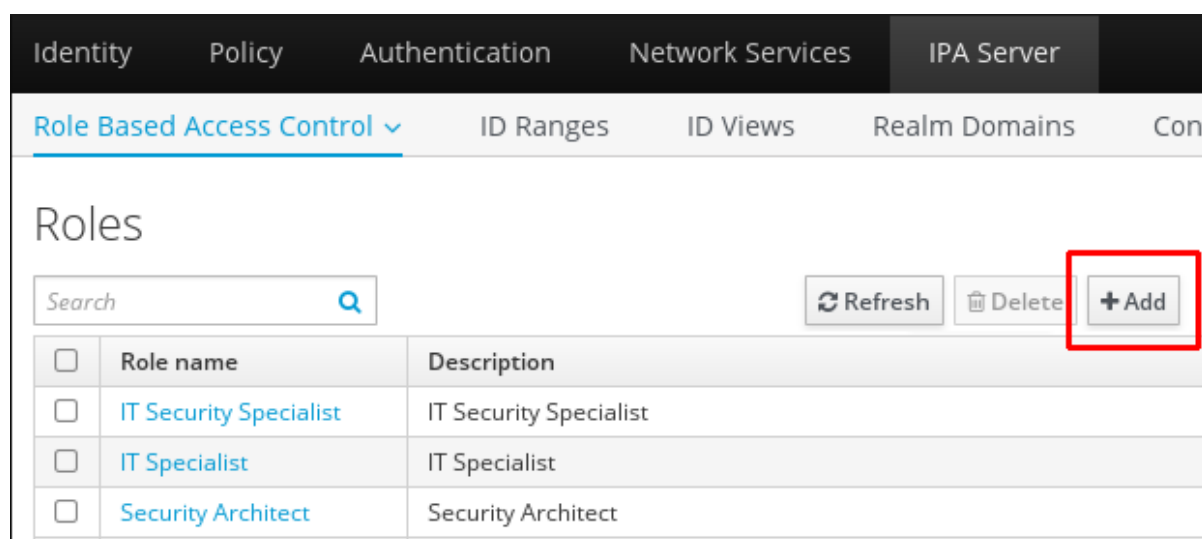
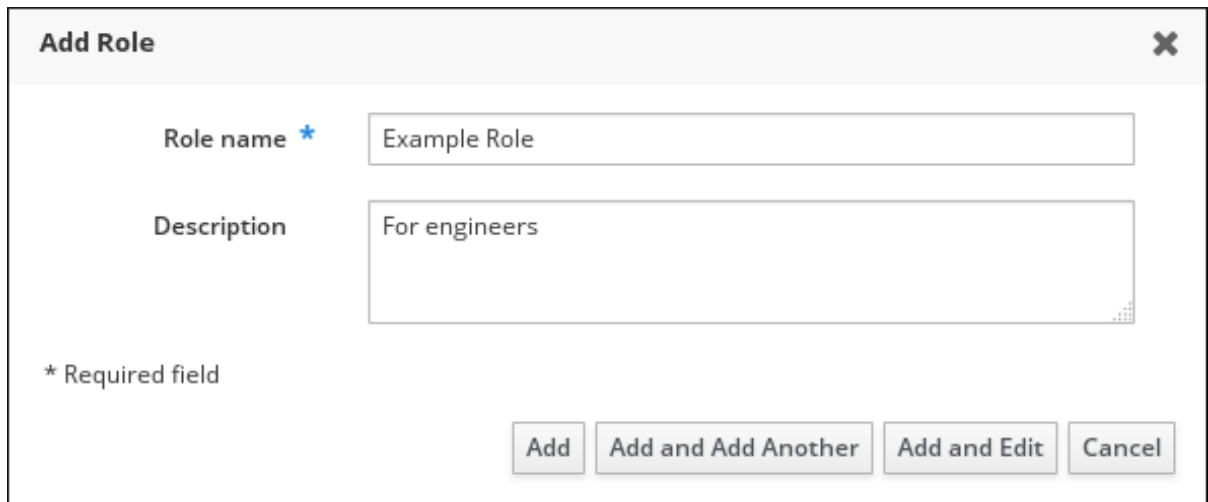


図10.6 新規ロールの追加

3. ロール名と説明を入力します。



Add Role [X]

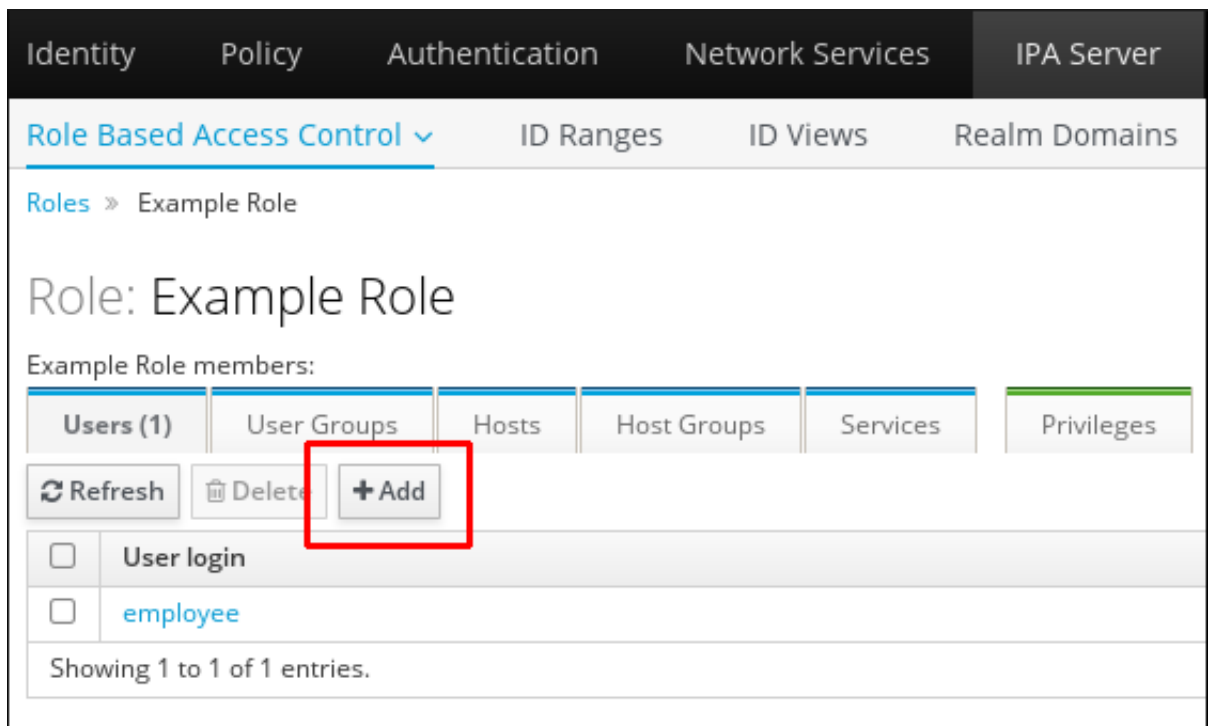
Role name *

Description

* Required field

図10.7 ロール追加のフォーム

4. **Add and Edit** をクリックして新規ロールを保存し、設定ページに移動します。
5. **Users** タブの上部で、グループ追加の場合は **Users Groups** タブで、**Add** をクリックします。



Identity Policy Authentication Network Services **IPA Server**

Role Based Access Control ▾ ID Ranges ID Views Realm Domains

Roles » Example Role

Role: Example Role

Example Role members:

Users (1)	User Groups	Hosts	Host Groups	Services	Privileges
<input type="button" value="Refresh"/> <input type="button" value="Delete"/> <input type="button" value="+Add"/>					
<input type="checkbox"/>	User login				
<input type="checkbox"/>	employee				

Showing 1 to 1 of 1 entries.

図10.8 ユーザーの追加

6. 左側のユーザーを選択し、右矢印 > を使って **Prospective** コラムに移動させます。

Add Users into Role Example Role

Filter available Users Filter

Available

<input type="checkbox"/>	User login
<input checked="" type="checkbox"/>	admin
<input type="checkbox"/>	helpdesk

Prospective

<input type="checkbox"/>	User login
<input type="checkbox"/>	manager

Add Cancel

図10.9 ユーザーの選択

7. **Privileges** タブの上部で **Add** をクリックします。

Role: Example Role

Example Role members:

Users (1) User Groups Hosts Host Groups Services **Privileges (4)** Settings

Refresh Delete **Add**

<input type="checkbox"/>	Privilege name
<input type="checkbox"/>	Automount Administrators
<input type="checkbox"/>	Certificate Administrators
<input type="checkbox"/>	DNS Administrators

図10.10 権限の追加

8. 左側の権限を選択し、右矢印 **>** を使って **Prospective** コラムに移動させます。

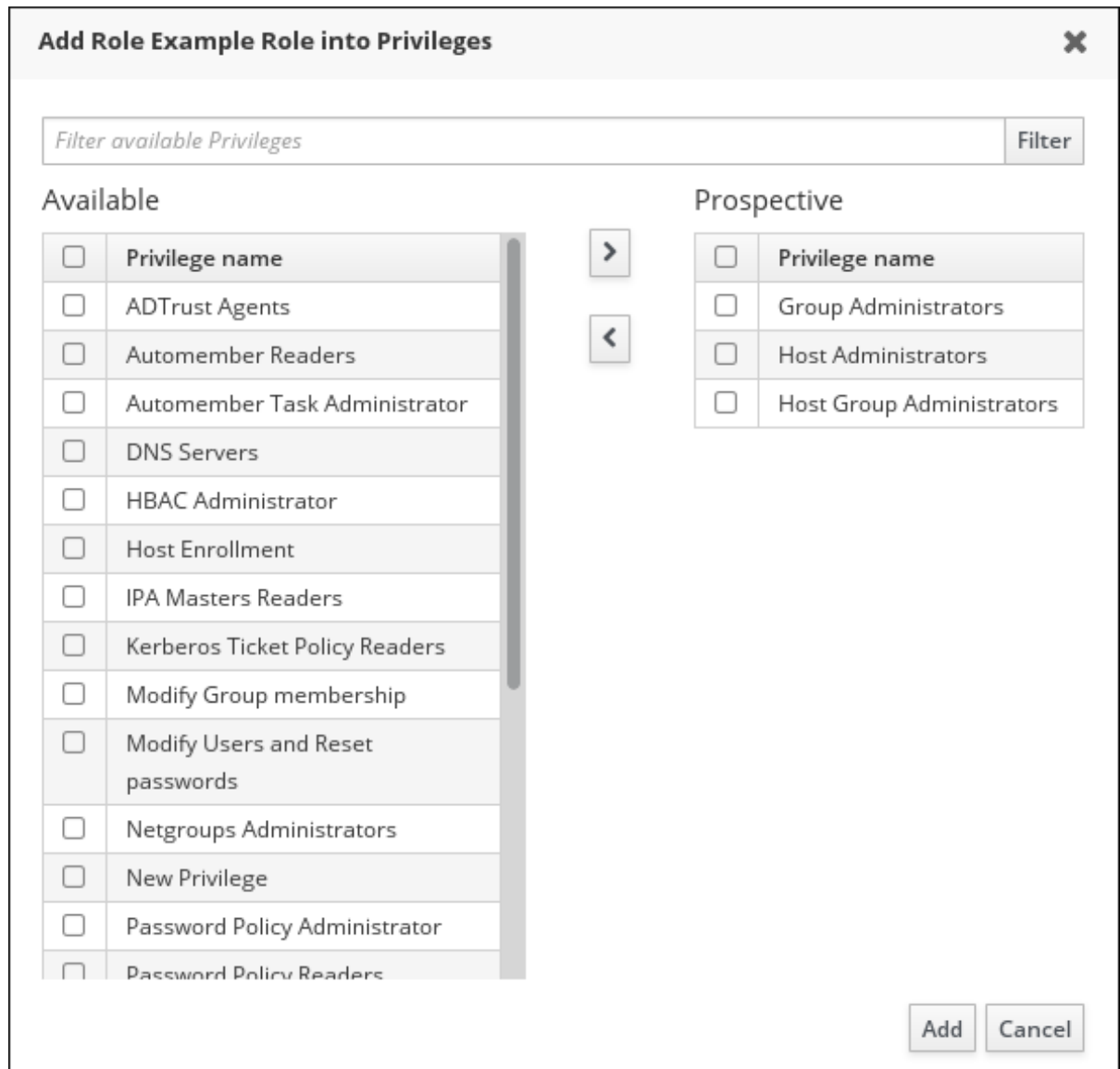


図10.11 権限の選択

9. **Add** をクリックして保存します。

10.4.1.2. コマンドライン でのロールの作成

1. 新規ロールを追加します。

```
[root@server ~]# kinit admin
[root@server ~]# ipa role-add --desc="User Administrator" useradmin
-----
Added role "useradmin"
-----
Role name: useradmin
Description: User Administrator
```

2. 必要な権限をロールに追加します。

```
[root@server ~]# ipa role-add-privilege --privileges="User
Administrators" useradmin
Role name: useradmin
Description: User Administrator
```

```
Privileges: user administrators
```

```
-----  
Number of privileges added 1  
-----
```

3. 必要なグループを追加します。このケースでは、既存の単一グループ **useradmin** を追加しています。

```
[root@server ~]# ipa role-add-member --groups=useradmins useradmin  
Role name: useradmin  
Description: User Administrator  
Member groups: useradmins  
Privileges: user administrators  
-----  
Number of members added 1  
-----
```

10.4.2. パーミッション

10.4.2.1. Web UI での新規パーミッションの作成

1. トップメニューで **IPA Server** タブを開き、**Role Based Access Control** サブタブを選択します。
2. **Permissions** タスクリンクを選択します。

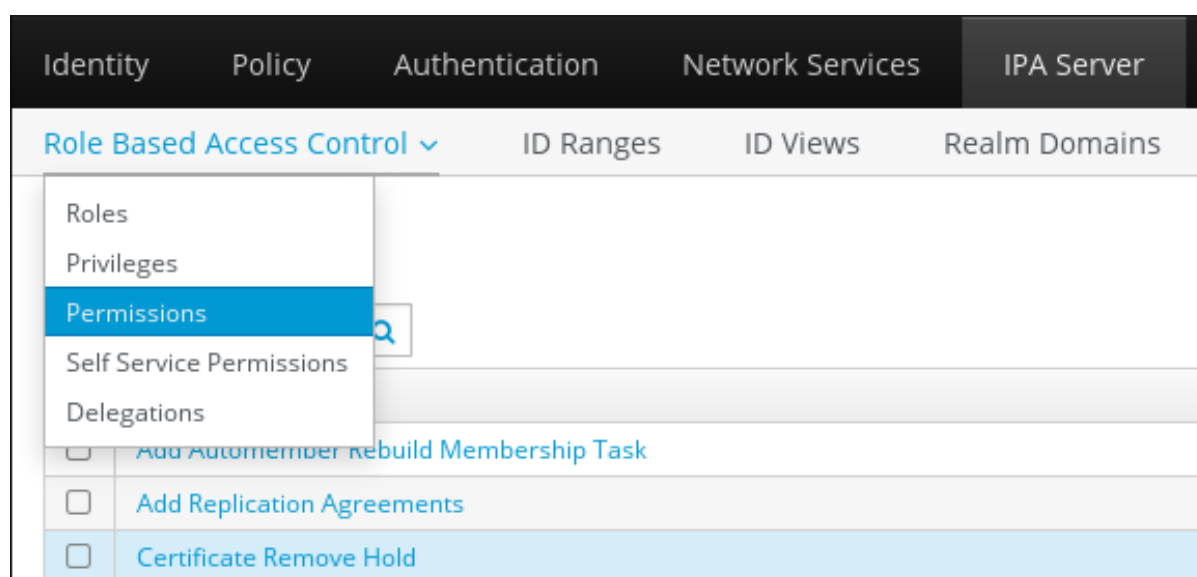


図10.12 パーミッションタスク

3. パーミッション一覧の上部にある **Add** をクリックします。

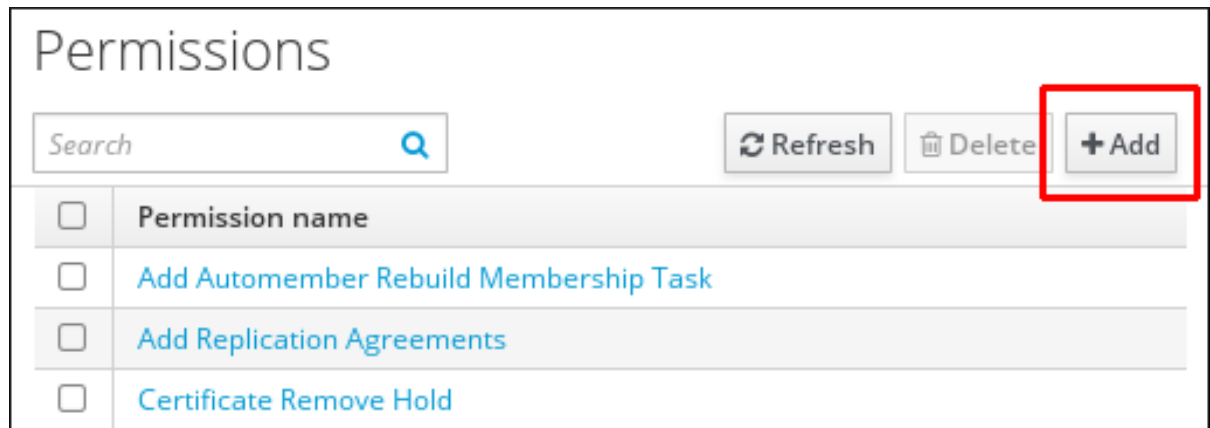


図10.13 新規パーミッションの追加

4. 表示されるフォームで新規パーミッションのプロパティを定義します。

Add Permission

Permission name *

New Permission

Bind rule type

☒ permission ☐ all ☐ anonymous

Granted rights *

☒ read ☐ search ☐ compare

☐ write ☐ add ☐ delete

☐ all

Type

Subtree *

dc=demo1,dc=freeipa,dc=org

Extra target filter

Add

Target DN

Member of group

employees

Undo

Add

Effective attributes

uid

Undo

loginshell

Undo

Add

* Required field

Add

Add and Add Another

Add and Edit

Cancel

図10.14 パーミッション追加のフォーム

5. フォームの下にある **Add** をクリックしてパーミッションを保存します。

以下のパーミッションのプロパティを指定することができます。

1. 新規パーミッションの名前を入力します。
2. 適切な **Bind rule type** を選択します。
 - **permission** がデフォルトのパーミッションタイプになります。これでアクセスが権限とロールによって付与されます。

- **all** を選択すると、パーミッションがすべての認証ユーザーに適用されます。
- **anonymous** を選択すると、非認証ユーザーを含むすべてのユーザーにパーミッションが適用されます。



注記

デフォルト以外の bind rule type のパーミッションを権限に追加することはできません。また、権限に既にあるパーミッションをデフォルト以外の bind rule type に設定することもできません。

3. パーミッションが付与する権限を **Granted rights** で選択します。
4. パーミッションのターゲットエントリーを識別する方法を定義します。
 - **Type** では、ユーザー、ホスト、またはサービスなどのエントリータイプを指定します。**Type** で値を設定すると、そのエントリータイプの ACI でアクセス可能な全属性が **Effective Attributes** に表示されます。

Type を定義すると、**Subtree** と **Target DN** が定義済みのいずれかの値に設定されます。

- **Subtree** は、サブツリーエントリーを指定します。このサブツリーエントリー以下のすべてのエントリーがターゲットになります。**Subtree** はワイルドカードや存在しないドメイン名 (DN) を受け付けませんので、既存のサブツリーエントリーを提供してください。例を示します。

```
cn=automount,dc=example,dc=com
```

- **Extra target filter** では、LDAP フィルターを使ってパーミッションが適用されるエントリーを特定します。このフィルターは有効な LDAP フィルターであればどれでも構いません。例を示します。

```
(!(objectclass=posixgroup))
```

IdM は、提供されたフィルターの有効性を自動的にチェックします。無効なフィルターが入力された場合は、パーミッションを保存使用とすると IdM が警告します。

- **Target DN** はドメイン名 (DN) を指定し、ワイルドカードを受け付けます。例を示します。

```
uid=*,cn=users,cn=accounts,dc=com
```

- **Member of group** は、特定のグループのメンバーにターゲットフィルターを設定します。

フィルター設定を記入して **Add** をクリックすると、IdM がフィルターの有効性を確認します。すべてのパーミッション設定が適切であれば、IdM は検索を実行します。設定が適切でない場合、IdM は該当設定についてのメッセージを表示します。

5. **Type** を設定する場合は、**Effective attributes** で利用可能な ACI 属性一覧から選択します。**Type** を使用しない場合は、**Effective attributes** フィールドに手動で属性を追加します。属性は 1 つずつ追加します。複数の属性を追加するには、**Add** をクリックして別の入力フィールドを追加します。



重要

パーミッションの属性を設定しない場合は、デフォルトで全属性が含まれることになります。

10.4.2.2. コマンドライン での新規パーミッションの作成

新規パーミッションを追加するには、**ipa permission-add** コマンドを発行します。対応するオプションを提供して、パーミッションのプロパティを指定します。

- プロパティ名を提供します。例を示します。

```
[root@server ~]# ipa permission-add "dns admin permission"
```

- **--bindtype** では、bind rule type を指定します。このオプションは**all**、**anonymous**、および**permission** の引数を受け付けます。例を示します。

```
--bindtype=all
```

--bindtype を使用しない場合は、タイプは自動的にデフォルトの**permission** の値に設定されます。



注記

デフォルト以外の bind rule type のパーミッションを権限に追加することはできません。また、権限に既にあるパーミッションをデフォルト以外の bind rule type に設定することもできません。

- **--permissions** には、パーミッションで付与される権限を記載します。複数の属性を設定するには、**--permissions** オプションを複数回使用するか、オプションを中括弧内にコンマ区切りの一覧で記載します。例を示します。

```
--permissions=read --permissions=write
--permissions={read,write}
```

- **--attrs** には、パーミッションが付与される属性を記載します。複数の属性を設定するには、**--attrs** オプションを複数回使用するか、オプションを中括弧内にコンマ区切りの一覧で記載します。例を示します。

```
--attrs=description --attrs=automountKey
--attrs={description,automountKey}
```

--attrs に提供する属性は既存のもので、該当のオブジェクトタイプに許可されている必要があります。そうでない場合は、コマンドはスキーマ構文エラーで失敗します。

- **--type** では、ユーザー、ホスト、またはサービスなどのエン트리タイプを定義します。各タイプには、許可される属性セットがあります。例を示します。

```
[root@server ~]# ipa permission-add "manage service" --
permissions=all --type=service --attrs=krbprincipalkey --
attrs=krbprincipalname --attrs=managedby
```

- **--subtree** では、サブツリーエントリーを提供します。これでフィルターがこのサブツリー

エントリー以下にあるすべてのエントリーをターゲットとします。 **--subtree** はワイルドカードや存在しないドメイン名 (DN) を受け付けません。ディレクトリー内の DN を使用してください。

IdM は簡素化された平坦なディレクトリーツリー構造を使用しているので、 **--subtree** を使って、automount の場所のような、他の設定のコンテナや親エントリーである、一定タイプのエントリーをターゲットにすることができます。例を示します。

```
[root@server ~]# ipa permission-add "manage automount locations" --
subtree="ldap://ldap.example.com:389/cn=automount,dc=example,dc=com"
--permissions=write --attrs=automountmapname --attrs=automountkey --
attrs=automountInformation
```

--type と **--subtree** のオプションは、相互排他的になります。

- **--filter** は、パーミッションが適用されるエントリーを特定する LDAP フィルターを使用します。IdM は提供されたフィルターの有効性を自動的にチェックします。このフィルターは有効な LDAP フィルターにします。例を示します。

```
[root@server ~]# ipa permission-add "manage Windows groups" --
filter="(!(objectclass=posixgroup))" --permissions=write --
attrs=description
```

- **--memberof** では、特定のグループが存在することを確認した後、そのグループのメンバーにターゲットフィルターを設定します。例を示します。

```
[root@server ~]# ipa permission-add ManageHost --permissions="write"
--subtree=cn=computers,cn=accounts,dc=testrelm,dc=com --
attr=nshostlocation --memberof=admins
```

- **--targetgroup** では、特定のユーザーグループが存在することを確認した後、そのグループにターゲットを設定します。

Target DN 設定は Web UI では利用可能ですが、コマンドラインでは使用できません。



注記

パーミッションの修正および削除に関する情報については、**ipa permission-mod --help** と **ipa permission-del --help** のコマンドを実行してください。

10.4.2.3. デフォルトの管理パーミッション

管理 (Managed) パーミッションは、Identity Management でインストール済みとして提供されるパーミッションです。これらはユーザーが作成した通常のパーミッションのように動作しますが、以下の点が異なります。

- 名前、場所、およびターゲット属性が修正できません。
- 削除することができません。
- 以下の 3 つの属性セットがあります。
 - **default** 属性は IdM が管理し、ユーザーは修正することができません。

- *included* 属性は、ユーザーが追加する属性です。included 属性を管理パーミッションに追加するには、**ipa permission-mod** コマンドで **--includedattrs** オプションを使用して属性を指定します。
- *excluded* 属性は、ユーザーが削除する属性です。excluded 属性を管理パーミッションに追加するには、**ipa permission-mod** コマンドで **--excludedattrs** オプションを使用して属性を指定します。

管理パーミッションは、default および included 属性セットに表示される全属性に適用されますが、excluded セットの属性には適用されません。

管理パーミッション修正時に **--attrs** オプションを使用すると、included および excluded 属性セットは自動的に調整され、**--attrs** で提供された属性のみが有効になります。



注記

管理パーミッションは削除できませんが、その bind type を **permission** に設定し、該当管理パーミッションを全権限から削除すると、実質的に無効となります。

すべての管理パーミッションの名前は **System:** で始まります。たとえば、*System: Add Sudo rule* や *System: Modify Services* などです。

IdM の以前のバージョンでは、デフォルトのパーミッションに異なるスキームを使用しており、ユーザーはデフォルトのパーミッションを修正できず、パーミッションを権限に割り当てることのみが可能でした。これらデフォルトのパーミッションはほとんど管理パーミッションとなっていますが、以下のパーミッションはまだ以前のスキームを使用しています。

- Add Automember Rebuild Membership Task
- Add Replication Agreements
- Certificate Remove Hold
- Get Certificates status from the CA
- Modify DNA Range
- Modify Replication Agreements
- Remove Replication Agreements
- Request Certificate
- Request Certificates from a different host
- Retrieve Certificates from the CA
- Revoke Certificate
- Write IPA Configuration

管理パーミッションを Web UI から修正する場合は、修正できない属性は灰色表示となります。

Permission: System: Modify Users

Settings Privileges (2)

Refresh Reset Update

Permission settings

Permission name

System: Modify Users

Bind rule type

☒ permission ☐ all ☐ anonymous

Granted rights

<input type="checkbox"/> read	<input type="checkbox"/> search	<input type="checkbox"/> compare	<input checked="" type="checkbox"/> write
<input type="checkbox"/> add	<input type="checkbox"/> delete	<input type="checkbox"/> all	

図10.15 灰色表示となった属性

コマンドラインから管理パーミッションを修正する場合、修正不可能な属性の変更はシステムで許可されません。たとえば、デフォルトの **System: Modify Users** パーミッションをグループに適用しようとするとう失敗します。

```
$ ipa permission-mod 'System: Modify Users' --type=group
ipa: ERROR: invalid 'ipapermlocation': not modifiable on managed
permissions
```

ただし、**System: Modify Users** パーミッションを **GECOS** 属性に適用しないようにすることは可能です。

```
$ ipa permission-mod 'System: Modify Users' --excludedattrs=gecos
-----
Modified permission "System: Modify Users"
```

10.4.2.4. Identity Management の以前のバージョンのパーミッション

Identity Management の以前のバージョンでは、パーミッションは別の方法で処理されていました。たとえば、

- 書き込み、追加、および削除パーミッションタイプのみが利用可能でした。
- パーミッション設定のオプションは詳細設定ができませんでした。たとえば、同一のパーミッションにフィルターとサブツリーの両方を追加することはできませんでした。

- グローバルの IdM ACI は、ログインしていない匿名ユーザーを含めた、サーバーの全ユーザーに読み取りアクセスを付与していました。

パーミッションの処理方法が新しくなったことで、IdM ではユーザーまたはグループアクセスの制御が大幅に改善しました。一方で、以前のバージョンとの後方互換性も維持しています。IdM の以前のバージョンからアップグレードすると、全サーバー上のグローバルの IdM ACI が削除され、管理パーミッションで置き換えられます。

以前の方法で作成されたパーミッションは、修正時に自動的に新しいスタイルに変換されます。変更されない場合は、以前のスタイルのパーミッションは変換されずに残ります。パーミッションは一旦新しいスタイルを使用すると、元のスタイルに戻すことはできません。



注記

IdM の以前のバージョンを実行しているサーバー上でパーミッションを権限に割り当てることは、引き続き可能です。

`ipa permission-show` と `ipa permission-find` のコマンドは、新スタイルと以前のスタイルの両方のパーミッションを認識します。これらコマンドの出力は新スタイルになりますが、パーミッション自体を変更するものではありません。パーミッションエントリーはデータを出力する前にメモリーでのみアップグレードされ、変更は LDAP にコミットされません。

新スタイルおよび旧スタイルのパーミッションは両方とも、以前のバージョンの IdM および現行バージョンの IdM を実行している全サーバーに効力が及びます。ただし、以前のバージョンの IdM を実行しているサーバーでは新スタイルのパーミッションの作成や変更はできません。

10.4.3. 権限

10.4.3.1. Web UI での新規権限の作成

1. トップメニューで **IPA Server** タブを開き、**Role Based Access Control** サブタブを選択します。
2. **Privileges** タスクリンクを選択します。

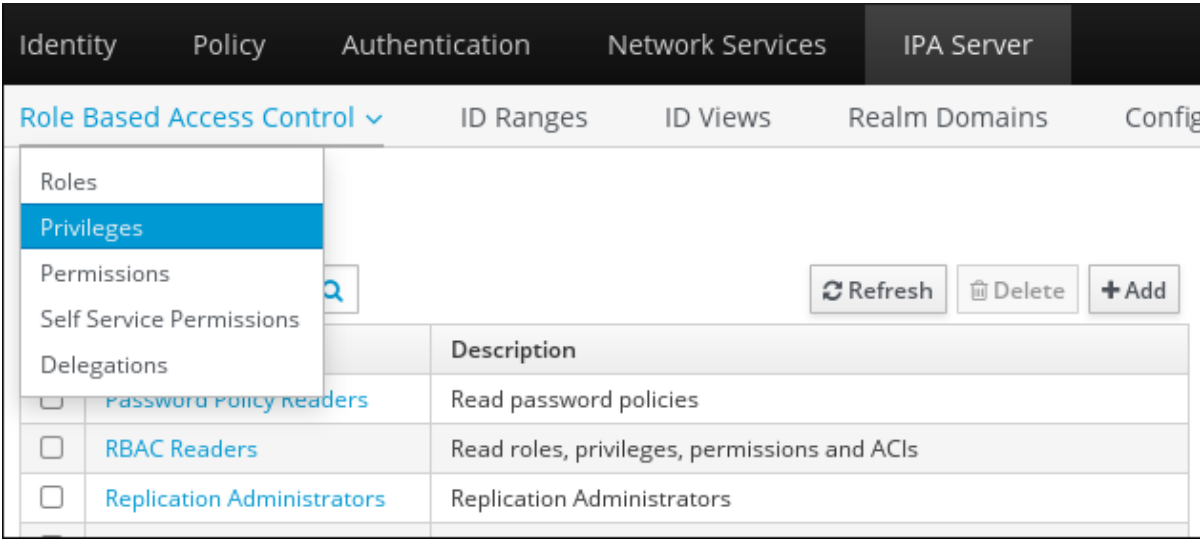


図10.16 権限タスク

3. 権限一覧の上部にある **Add** をクリックします。

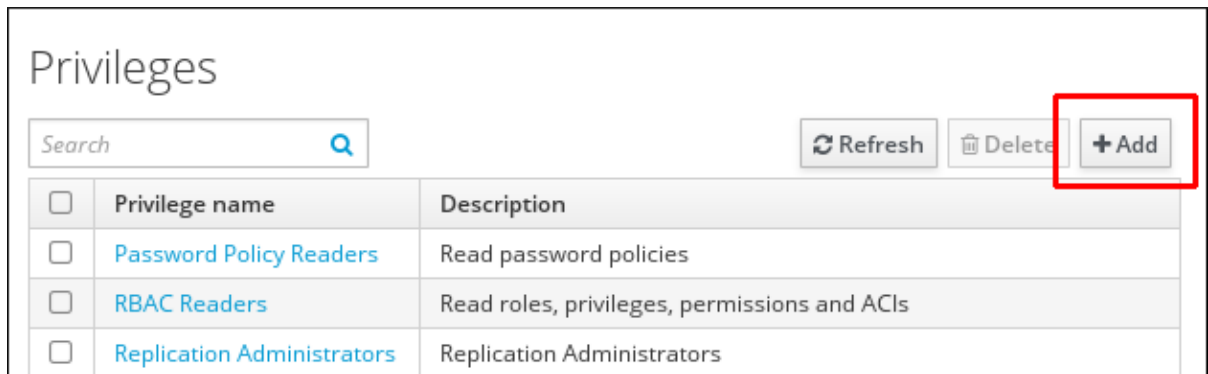


図10.17 新規権限の追加

4. 権限の名前と説明を入力します。

図10.18 権限追加のフォーム

5. **Add and Edit** をクリックして、権限設定ページに移動し、パーミッションを追加します。
6. **Permissions** タブを選択します。
7. 権限に追加するパーミッション一覧のトップにある **Add** をクリックします。

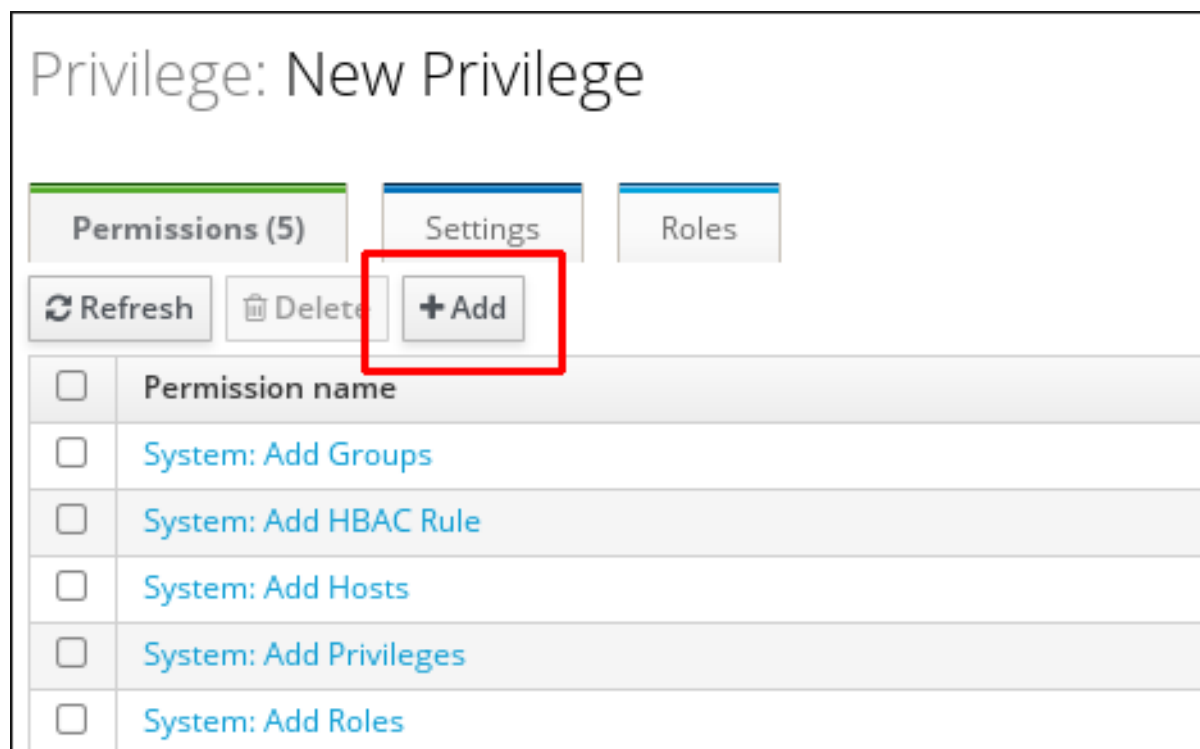


図10.19 パーミッションの追加

8. 追加するパーミッションの名前の横にあるチェックボックスを選択し、右矢印 > を使って **Prospective** コラムに移動させます。

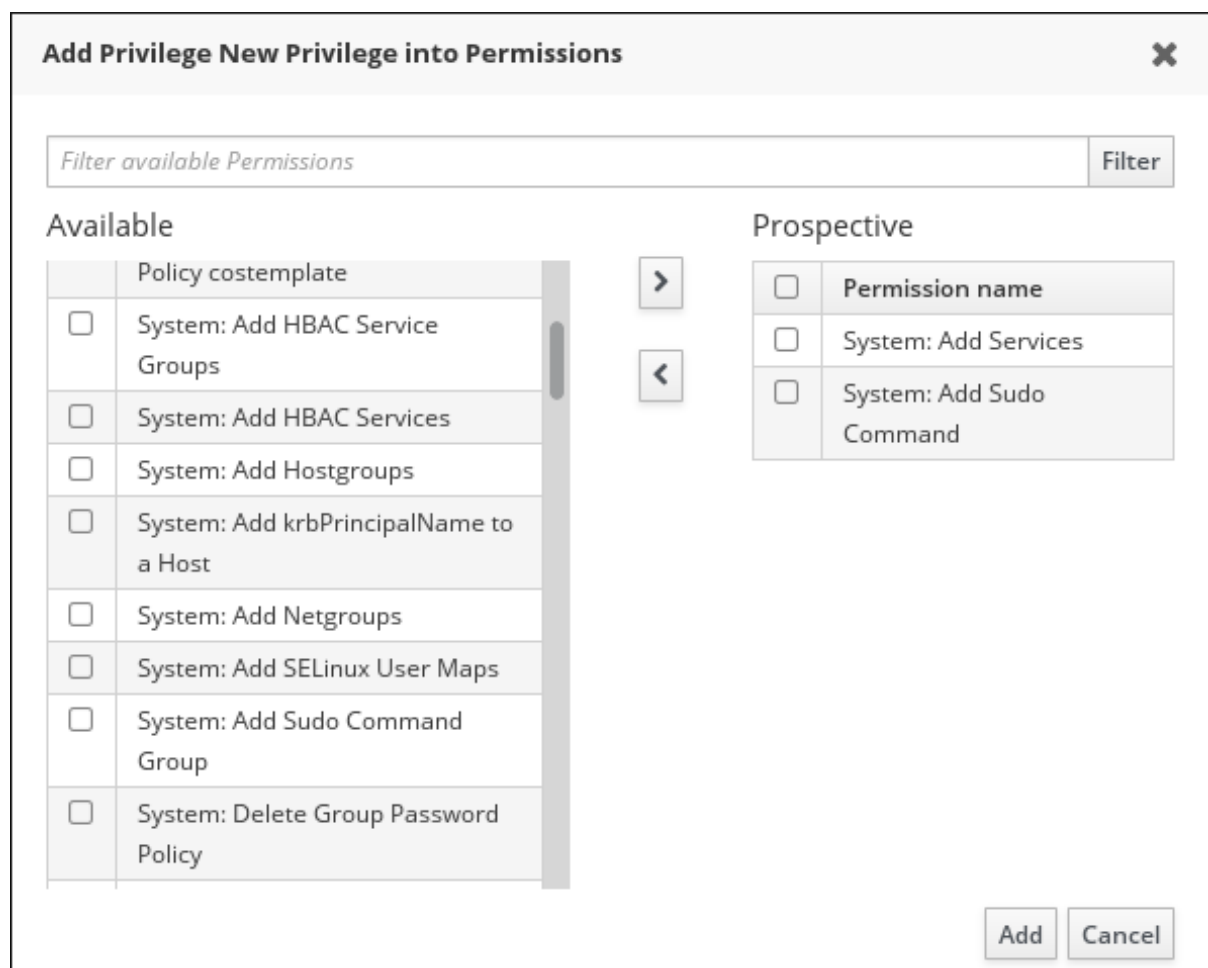


図10.20 パーミッションの選択

9. **Add** をクリックして保存します。

10.4.3.2. コマンドライン での新規権限の作成

権限のエントリは **privilege-add** コマンドで作成されます。次に、**privilege-add-permission** コマンドを使ってパーミッションを権限グループに追加します。

1. 権限エントリを作成します。

```
[jsmith@server ~]$ ipa privilege-add "managing filesystems" --  
desc="for filesystems"
```

2. 希望するパーミッションを割り当てます。

```
[jsmith@server ~]$ ipa privilege-add-permission "managing  
filesystems" --permissions="managing automount" --  
permissions="managing ftp services"
```

パート **IV.** アイデンティティの管理

第11章 ユーザーアカウントの管理

本章は、ユーザーアカウントの一般的な管理および設定について説明します。

11.1. ユーザーホームディレクトリーの設定

ユーザーにはすべて、ホームディレクトリーを設定することを推奨します。デフォルトでは、ユーザーのホームディレクトリーの場所は、**/home/** ディレクトリーが想定されます。たとえば、IdM では、**user_login** ログインのユーザーは、**/home/user_login** にホームディレクトリーを設定する必要があります。



注記

デフォルトのホームディレクトリーの場所を変更するには、**ipa config-mod** コマンドを使用してください。

IdM では、ユーザーのホームディレクトリーは自動的に作成されませんが、ユーザーがログインした時に、自動的にホームディレクトリーが作成されるように、PAM ホームディレクトリーモジュールを設定できます。または、NFS 共有と **automount** ユーティリティーを使用して手動でホームディレクトリーを追加することもできます。

11.1.1. PAM ホームディレクトリーモジュールを使用して自動的にホームディレクトリーをマウントする手順

サポートされる PAM ホームディレクトリーモジュール

IdM ドメインにユーザーがログインした時点で自動的にホームディレクトリーが作成されるように、PAM ホームディレクトリーモジュールを設定するには、以下の PAM モジュールの 1 つを使用します。

- **pam_oddjob_mkhomedir**
- **pam_mkhomedir**

IdM はまず、**pam_oddjob_mkhomedir** の使用を試行します。このモジュールがインストールされていない場合は、IdM は代わりに **pam_mkhomedir** の使用を試みます。

PAM ホームディレクトリーモジュールの設定

PAM ホームディレクトリーを有効化すると、ローカル環境に影響があります。そのため、各クライアントやサーバーなど、必要とされる場所上で個別にモジュールを有効化する必要があります。

サーバーまたはクライアントのインストール時にはモジュールを設定するには、マシンのインストール時に、**--mkhomedir** オプションを指定して **ipa-server-install** または **ipa-client-install** ユーティリティーを実行します。

すでにインストール済みのサーバーやクライアントでモジュールを設定するには **authconfig** ユーティリティーを使用します。以下に例を示します。

```
# authconfig --enablemkhomedir --update
```

authconfig を使用したホームディレクトリーの作成に関する情報は、[「authconfig を使用したカスタムホームディレクトリーの有効化」](#)を参照してください。

11.1.2. ホームディレクトリーを手動でマウントする手順

IdM ドメインの全マシンで利用できるように **/home/** ディレクトリーを提供し、**automount** ユーティリティーで IdM マシンにディレクトリーをマウントするには、NFS ファイルサーバーを使用します。

NFS 使用時に発生する可能性のある問題

NFS は、パフォーマンスおよびセキュリティに悪影響を与える可能性があります。たとえば、NFS では、NFS ユーザーに root のアクセス権限を割り当てるとセキュリティの脆弱性が発生したり、**/home/** ディレクトリーツリー全体を読み込む際にパフォーマンスの問題が発生したり、ホームディレクトリーにリモートサーバーを使用する際にネットワークパフォーマンスの問題が発生したりする可能性があります。

これらの問題の影響を軽減するために、以下のガイドラインに従うことが推奨されます。

- **automount** を使用して、ユーザーがログインした時のみ、ユーザーのホームディレクトリーのみをマウントします。これを使用して、**/home/** ツリー全体を読み込ませないようにしてください。
- 限定的なパーミッションを割り当てたりリモートユーザーを使用してホームディレクトリーを作成し、このユーザーとして IdM サーバーに共有をマウントします。IdM サーバーは **httpd** プロセスとして実行されるので、**sudo** または同様のプログラムを使用して IdM サーバーへの限定的なパーミッションを許可して、NFS サーバーにホームディレクトリーを作成することができます。

NFS および automount を使用したホームディレクトリーの設定

NFS 共有および **automount** を使用して別の場所から IdM サーバーにホームディレクトリーを手動で追加するには以下を実行します。

1. ユーザーディレクトリーマップ用に新しい場所を作成します。

```
$ ipa automountlocation-add userdirs
Location: userdirs
```

2. 新たに作成した場所の **auto.direct** ファイルにダイレクトマップを追加します。**auto.direct** ファイルは、**ipa-server-install** ユーティリティーで自動的に作成された **automount** マップです。以下の例では、マウントポイントは **/share** です。

```
$ ipa automountkey-add userdirs auto.direct --key=/share --info="-ro,soft, server.example.com:/home/share"

Key: /share
Mount information: -ro,soft, server.example.com:/home/share
```

IdM での **automount** の使用に関する詳細は [33章Automount の使用](#) を参照してください。

11.2. ユーザーのライフサイクル

Identity Management (IdM) では *stage*、*active* および *preserved* の 3 つのユーザーアカウントの状態をサポートします。

- **Stage** ユーザーは認証ができません。これは最初の状態です。アクティブなユーザーが必要とする、ユーザーアカウントのプロパティーの一部は、まだ設定されていません。
- **Active** ユーザーは認証が可能です。この状態では、必要とされるユーザーアカウントのプロパティーはすべて設定されています。
- **Preserved** ユーザーは、過去に **active** であったユーザーです。このユーザーは非アクティ

ブとみなされ、IdM への認証はできません。Preserved ユーザーは、Active ユーザーに設定されたていたアカウントプロパティの大半が保持されますが、どのユーザーグループにも所属しません。



注記

preserved 状態のユーザー一覧は、以前のユーザーアカウントの履歴を提供することができます。

ユーザーエントリーも IdM データベースから完全に削除することができます。ユーザーエントリーを完全に削除すると、エントリー自体とグループメンバーシップやパスワードなど、そのユーザーの情報をすべて IdM から削除します。システムアカウントやホームディレクトリーなどのユーザーの外部設定は削除されませんが、IdM でアクセスできなくなります。



重要

削除したユーザーアカウントは復元することができます。ユーザーアカウントを削除すると、そのアカウントに関連付けられたすべての情報が完全に失われます。

新規管理者ユーザーは、デフォルトの **admin** ユーザーなど、別の管理者でしか作成できません。すべての管理者ユーザーを誤って削除してしまった場合は、Directory Manager が手動で新規の管理者を Directory Server に作成する必要があります。

ユーザーのライフサイクル管理に関する操作

ユーザーのプロビジョニングを管理するには、管理者はユーザーアカウントの状態を変更します。新規ユーザーは、**active** または **stage** のいずれかの状態で追加できますが、**preserved** ではできません。

IdM は、ユーザーライフサイクル管理の以下の操作をサポートします。

stage → active

stage の状態のアカウントが正しくアクティベートされる準備ができると、管理者は **active** の状態に移動します。

active → preserved

ユーザーが企業を退社した後は、管理者はアカウントを **preserved** の状態に移動します。

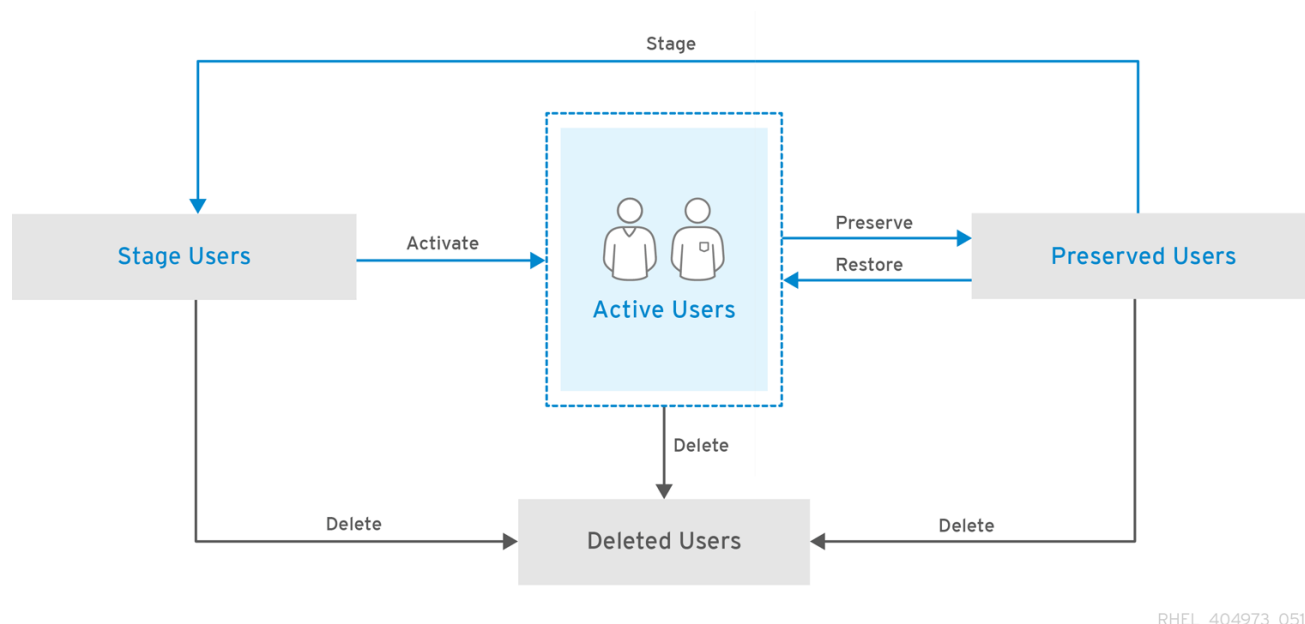
preserved → active

以前働いていたユーザーが再度企業に復帰した場合に、管理者は、ユーザーアカウントを **preserved** から **active** の状態に移動して復帰させます。

preserved → stage

以前のユーザーが企業に復帰する予定の場合に、管理者はアカウントの状態を **preserved** から **stage** に移動して、後ほど再アクティベートできるようにアカウントの準備をします。

IdM から active、stage および preserved ユーザーを完全に削除することもできます。stage ユーザーは **preserved** の状態に移動できず、完全に削除することしかできない点に注意してください。



RHEL_404973_0516

図11.1 ユーザーライフサイクルの操作

11.2.1. stage または **Active** ユーザーの追加

Web UI でのユーザーの追加

1. **Identity** → **Users** タブを選択します。
2. ユーザーを **active** または **stage** のいずれの状態を追加するかによって、**Active users** または **Stage users** カテゴリーを選択します。

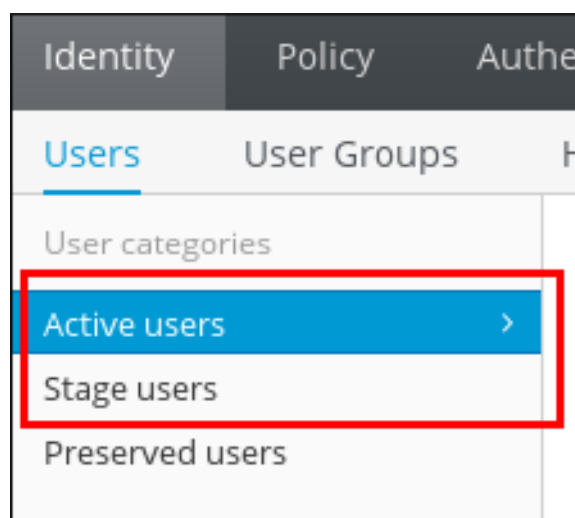


図11.2 ユーザーカテゴリーの選択

active または **stage** ユーザーのライフサイクルに関する詳しい情報は、「[ユーザーのライフサイクル](#)」を参照してください。

3. ユーザー一覧上部にある **Add** をクリックします。

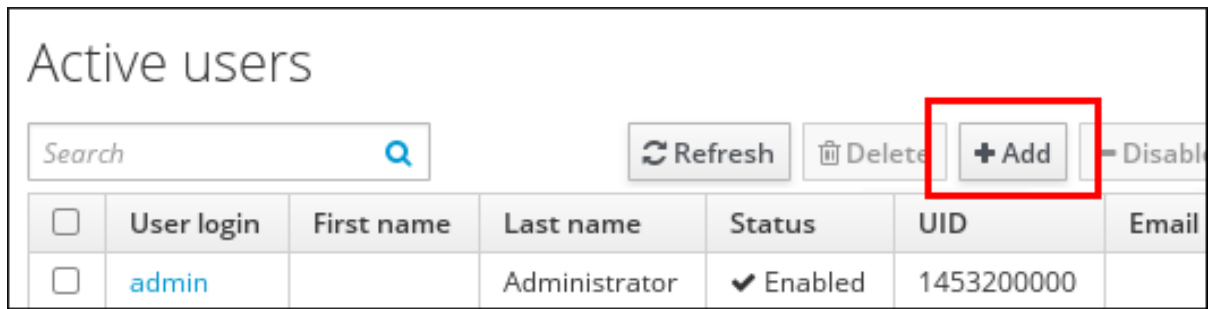


図11.3 ユーザーの追加

4. Add User のフォームに入力します。

ユーザーログインを手動で設定しない場合には IdM は指定の名前に基づいて自動的にログイン ID が生成されます。

5. Add をクリックします。

または、**Add and Add Another** をクリックして、別のユーザーを追加するか、**Add and Edit** をクリックして新規ユーザーエントリーの編集を開始します。ユーザーエントリーの編集に関する情報は「[ユーザーの編集](#)」を参照してください。

コマンドラインからのユーザーの追加

active の状態の新規ユーザーを追加するには、**ipa user-add** コマンドを使用します。**stage** の状態の新規ユーザーを追加するには、**ipa stageuser-add** コマンドを使用します。



注記

active または **stage** ユーザーのライフサイクルに関する詳しい情報は、「[ユーザーのライフサイクル](#)」を参照してください。

ipa user-add および **ipa stageuser-add** をオプションなしで実行すると、最小限必要なユーザー属性の入力が求められ、他の属性についてはデフォルト値が使用されます。または、オプションを追加してコマンドに直接さまざまな属性を指定することができます。

対話型のセッションでは、オプションなしでコマンドを実行すると、IdM は指定したフルネームをもとに自動的にユーザーログインを提案し、カッコ内 ([]) に表示します。デフォルトのログイン ID を受け入れるには、**Enter** を押して確定します。カスタムログインを指定するには、デフォルトを確定せずに、カスタムのログインを指定します。

```
$ ipa user-add
First name: first_name
Last name: last_name
User login [default_login]: custom_login
```

ipa user-add および **ipa stageuser-add** にオプションを追加すると、多くのユーザー属性にカスタムの値を定義できるようになります。これを使用すると、対話型セッションよりも多くの情報を指定することができます。たとえば、stage ユーザーを追加するには以下を実行します。

```
$ ipa stageuser-add stage_user_login --first=first_name --last=last_name -
-email=email_address
```

ipa user-add および **ipa stageuser-add** で利用可能なオプションの完全な一覧については、コマンドに **--help** オプションを指定して実行します。

11.2.1.1. ユーザー名の要件

IdM は、以下の正規表現で記述可能なユーザー名をサポートします。

```
[a-zA-Z0-9_][a-zA-Z0-9_-.]{0,252}[a-zA-Z0-9_.$-]?
```



注記

Samba 3.x マシンのサポートを有効にするために、ユーザー名の最後にドル記号 (\$) を指定できるようになっています。

名前に大文字を含むユーザーを追加した場合には、IdM は自動的に小文字に変換して保存します。そのため、IdM はログイン時にユーザー名はすべて小文字で入力する必要があります。さらに、**user** と **User** のように、ユーザー名の違いが大文字と小文字のみのユーザーは追加できません。

デフォルトでは、ユーザー名の最大文字数は 32 文字となっています。これを変更するには、**ipa config-mod --maxusername** コマンドを使用します。たとえば、ユーザー名の最大長を 64 文字に変更するには、以下を実行します。

```
$ ipa config-mod --maxusername=64
Maximum username length: 64
...
```

11.2.1.2. カスタムの UID または GID 番号の定義

カスタムの UID または GID 番号を指定せずにユーザーエントリーを追加する場合には、IdM は、ID 範囲内で次に利用可能な ID 番号が自動的に割り当てます。つまり、ユーザーには常に一意の ID 番号が指定されることになります。ID 範囲に関する詳しい情報は[14章 一意の UID および GID 番号の割り当て](#)を参照してください。

カスタム ID 番号を指定すると、カスタム ID 番号が一意であるかどうか、サーバーによる検証はありません。そのため、複数のユーザーエントリーに同じ ID 番号が割り当てられる可能性があります。Red Hat は、複数のエントリーに同じ ID 番号が割り当てられないようにすることを推奨します。

11.2.2. ユーザーの一覧表示およびユーザーの検索

Web UI でのユーザーの表示

1. **Identity → Users** タブを選択します。
2. **Active users**、**Stage users** または **Preserved users** カテゴリーを選択します。

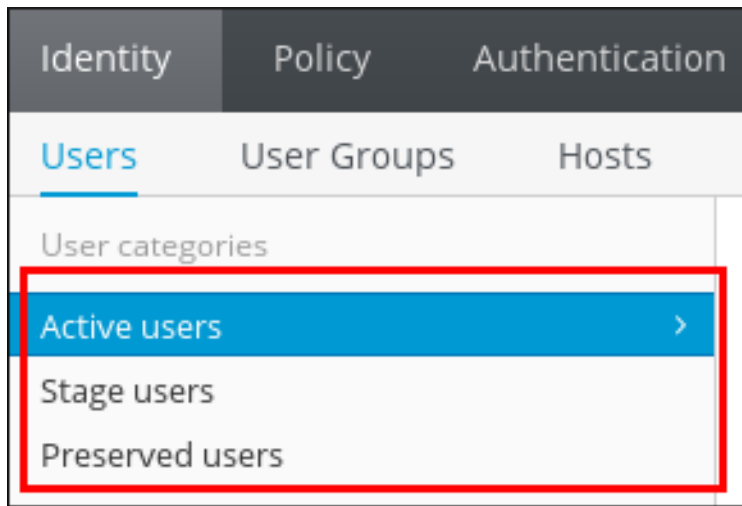


図11.4 ユーザーの一覧表示

Web UI でのユーザー情報の表示

ユーザーの詳細情報を表示するには、ユーザーリストでユーザー名をクリックします。

Active users

Search

Refresh

<input type="checkbox"/>	User login	First name	Last name	Status	UID	Email address
<input type="checkbox"/>	admin		Administrator	✓ Enabled	1453200000	
<input type="checkbox"/>	user	User	User	✓ Enabled	1453200006	user1@example.com
<input type="checkbox"/>	user2	User2	User2	✓ Enabled	1453200007	user2@abc.idm.l
<input type="checkbox"/>	user3	User3	User3	✓ Enabled	1453200008	user3@abc.idm.l

図11.5 ユーザー情報の表示

コマンドラインからのユーザーの表示

active ユーザーすべてを表示するには、**ipa user-find** コマンドを実行します。stage ユーザーをすべてを表示するには、**ipa stageuser-find** コマンドを使用します。preserved ユーザーを表示するには、**ipa user-find --preserved=true** コマンドを実行します。

例を示します。

```
$ ipa user-find
-----
23 users matched
-----
User login: admin
Last name: Administrator
Home directory: /home/admin
Login shell: /bin/bash
UID: 1453200000
GID: 1453200000
Account disabled: False
Password: True
Kerberos keys available: True
```

```
User login: user
...
```

ipa user-find および **ipa stageuser-find** にオプションおよび引数を追加して、検索条件を定義して検索結果を絞り込むことができます。たとえば、特定のタイトルが定義された active ユーザーをすべてを表示するには、以下を実行します。

```
$ ipa user-find --title=user_title
-----
2 users matched
-----
    User login: user
    ...
    Job Title: Title
    ...

    User login: user2
    ...
    Job Title: Title
    ...
```

同様に、ログインに **user** が含まれる stage ユーザーすべてを表示するには、以下を実行します。

```
$ ipa user-find user
-----
3 users matched
-----
User login: user
...

User login: user2
...

User login: user3
...
```

ipa user-find および **ipa stageuser-find** で利用可能なオプションの完全な一覧については、コマンドに **--help** オプションを指定して実行します。

コマンドラインからのユーザー情報の表示

active または preserved ユーザーに関する情報を表示するには **ipa user-show** コマンドを使用します。

```
$ ipa user-show user_login
    User login: user_login
    First name: first_name
    Last name: last_name
    ...
```

stage ユーザーの情報を表示するには、**ipa stageuser-show** コマンドを使用します。

11.2.3. ユーザーの有効化、保存、削除、復元

以下のセクションでは、異なるユーザーライフサイクルの状態の間でユーザーのアカウントを移動する

方法について説明します。IdM のライフサイクルの状態に関する詳しい情報は、「[ユーザーのライフサイクル](#)」を参照してください。

Web UI でのユーザーライフサイクルの管理

stage ユーザーをアクティベートするには、以下を実行してください。

- **Stage users** 一覧で、アクティベートユーザーを選択して **Activate** をクリックします。

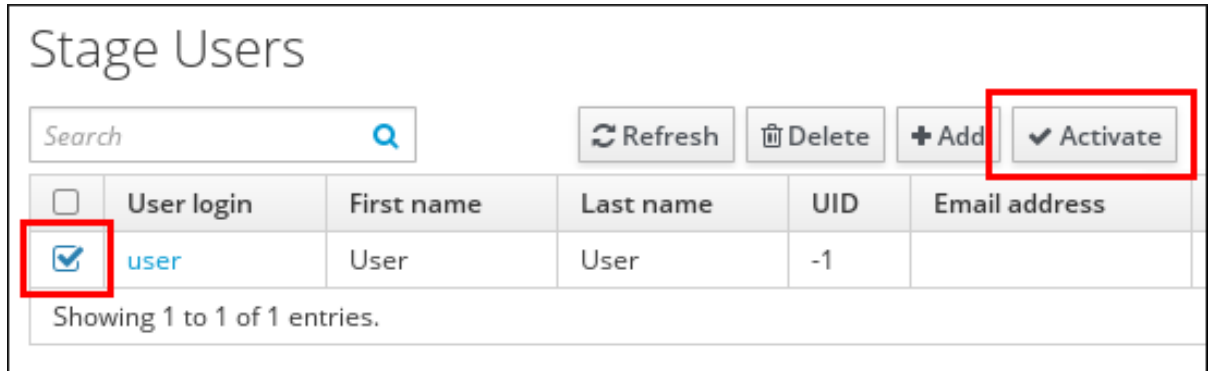


図11.6 ユーザーの有効化

ユーザーを保存または削除するには、以下を実行します。

1. **Active users** または **Stage users** リストで対象のユーザーを選択して **Delete** をクリックします。

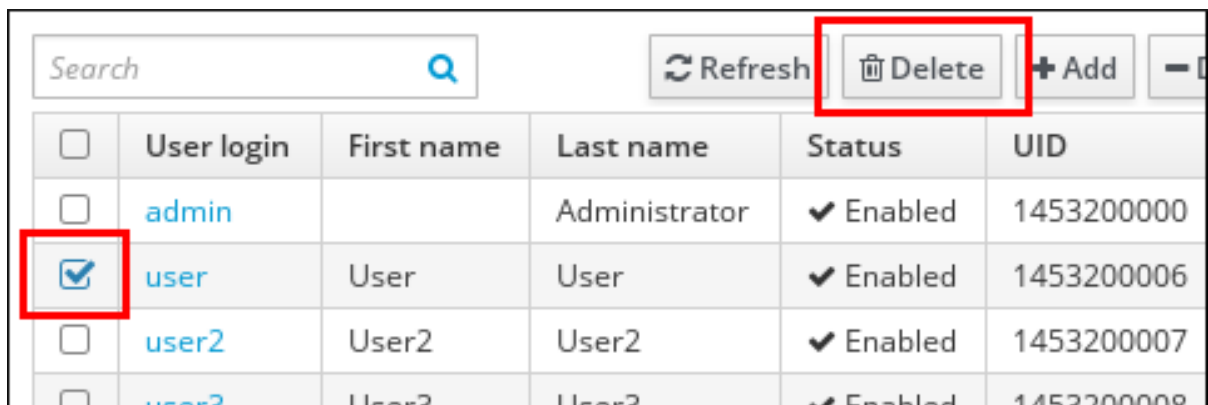


図11.7 ユーザーの削除

2. active ユーザーを選択した場合には **delete** または **preserve** を選択します。また、stage ユーザーを選択した場合には、そのユーザーは削除しかできません。デフォルトの UI オプションは **delete** です。

たとえば、active ユーザーを保存するには、以下を行います。

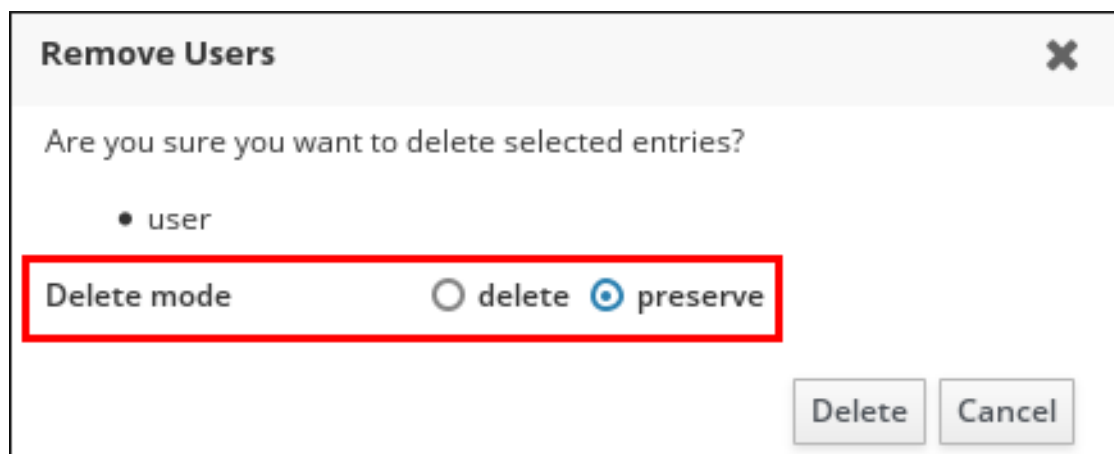


図11.8 Web UI での削除モードの選択

Delete ボタンをクリックして確定します。

preserved ユーザーを復元するには以下を行います。

- **Preserved users** 一覧で、復元するユーザーを選択し、**Restore** をクリックします。



図11.9 ユーザーの復元



注記

ユーザーを復元しても、そのアカウントに指定されている以前の属性すべてが復元されるわけではありません。たとえば、ユーザーのパスワードは復元されないので、再度定義する必要があります。

Web UI で、ユーザーを **preserved** から **stage** の状態に移動できません。

コマンドラインからのユーザーライフサイクルの管理

stage から **active** に移動してユーザーアカウントをアクティベートするには、**ipa stageuser-activate** コマンドを使用します。

```
$ ipa stageuser-activate user_login
-----
Stage user user_login activated
-----
...
```

ユーザーアカウントを保存または削除するには **ipa user-del** または **ipa stageuser-del** コマンドを使用します。

- IdM データベースから active ユーザーを完全に削除するには、オプションの指定なしに **ipa user-del** を実行します。

```
$ ipa user-del user_login
-----
Deleted user "user3"
-----
```

- active ユーザーを保存するには、**--preserve** オプションを指定して **ipa user-del** を実行します。

```
$ ipa user-del --preserve user_login
-----
Deleted user "user_login"
-----
```

- IdM データベースから stage ユーザーを完全に削除するには **ipa stageuser-del** を実行します。

```
$ ipa stageuser-del user_login
-----
Deleted stage user "user_login"
-----
```

注記

複数のユーザーを削除する場合には、**--continue** オプションを使用するとエラーが発生しても強制的にコマンドを続行できます。コマンドの完了時に、操作結果のサマリーが **stdout** の標準出力ストリームに表示されます。

```
$ ipa user-del --continue user1 user2 user3
```

--continue が指定されていない場合には、コマンドはエラーが発生するまでユーザーの削除を続け、エラーが発生すると操作を停止して終了します。

preserved から **active** に移動して preserved ユーザーアカウントを復元するには、**ipa user-undel** コマンドを使用します。

```
$ ipa user-undel user_login
-----
Undeleted user account "user_login"
-----
```

preserved から **stage** に移動して preserved ユーザーアカウントを復元するには、**ipa user-stage** コマンドを使用します。

```
$ ipa user-stage user_login
-----
Staged user account "user_login"
-----
```



注記

ユーザーアカウントを復元しても、そのアカウントに指定されている以前の属性すべてが復元されるわけではありません。たとえば、ユーザーのパスワードは復元されないの
で、再度定義する必要があります。

これらのコマンドや対応のオプションに関する情報は、コマンドに **--help** オプションを追加して実行してください。

11.3. ユーザーの編集

Web UI でのユーザーの編集

1. **Identity** → **Users** タブを選択します。
2. **Active users**、**Stage users** または **Preserved users** カテゴリを検索して編集するユーザーを特定します。
3. 編集するユーザー名をクリックします。

User categories

Active users >

Stage users

Preserved users

Active users

Search

<input type="checkbox"/>	User login	First name	Last name	Status	UID
<input type="checkbox"/>	admin		Administrator	✓ Enabled	14532
<input type="checkbox"/>	user	User	User	✓ Enabled	14532

図11.10 編集するユーザーの選択

4. 必要に応じてユーザーを属性フィールドを編集します。
5. ページ上部にある **Save** をクリックします。

✓ User: user

user is a member of:

Settings

User Groups

Netgroups

Roles

HBAC Rules

Sudo R

Refresh

Revert

Save

Actions ▼

Identity Settings

Job Title

First name

User

図11.11 変更したユーザー属性の保存

Web UI でユーザー情報を更新した後は、新しい値はすぐに同期されません。クライアントシステムで新しい値が反映されるまで約 5 分程度かかる可能性があります。

コマンドラインからのユーザーの編集

active または **preserved** の状態のユーザーを変更するには、**ipa user-mod** コマンドを使用します。**stage** の状態のユーザーを変更するには、**ipa stageuser-mod** コマンドを使用します。

ipa user-mod および **ipa stageuser-mod** コマンドでは、以下のオプションを利用できます。

- 変更するユーザーアカウントを特定するユーザーログイン
- 新規属性の値を指定するオプション

コマンドラインから変更可能なユーザーエントリーの属性に関する完全一覧は、**ipa user-mod** および **ipa stageuser-mod** で利用可能なオプション一覧を参照してください。オプション一覧を表示するには、コマンドに **--help** オプションを追加して実行してください。

ipa user-mod または **ipa stageuser-mod** に属性オプションを追加するだけで、現在の属性値を上書きします。たとえば、以下を実行するとユーザーのタイトルが変更されるか、ユーザーのタイトルが指定されていない場合には新規タイトルが追加されます。

```
$ ipa user-mod user_login --title=new_title
```

LDAP 属性に複数の値を指定可能な場合は、IdM でも複数の値を指定できます。たとえば、ユーザーアカウントに 2 つの電子メールを保存できます。既存の値を上書きすることなしに別の属性値を追加するには、新規属性値を指定するオプションと **--addattr** オプションを使用します。たとえば、すでにメールが指定されているユーザーアカウントに新規のメールアドレスを追加するには以下を実行します。

```
$ ipa user-mod user --addattr=mobile=new_mobile_number
-----
Modified user "user"
-----
  User login: user
...
  Mobile Telephone Number: mobile_number, new_mobile_number
...
```

同時に 2 つの属性値を設定するには、**--addattr** オプションを 2 度使用します。

```
$ ipa user-mod user --addattr=mobile=mobile_number_1 --
addattr=mobile=mobile_number_2
```

ipa user-mod コマンドは、属性値の設定に **--setattr** オプションを、属性値の削除に **--delattr** オプションを指定することができます。これらのオプションは、**--addattr** と同じように使用されます。詳細は **ipa user-mod --help** コマンドの出力を参照してください。



注記

ユーザーの現在のメールアドレスを上書きするには **--email** オプションを使用しますが、別のメールアドレスを追加するには、**--email** オプションと **--addattr** オプションを使用します。

```
$ ipa user-mod user --email=email@example.com
```

```
$ ipa user-mod user --addattr=mail=another_email@example.com
```

11.4. ユーザーアカウントの有効化、無効化

管理者は、active ユーザーアカウントを無効化および有効化できます。ユーザーアカウントのアクティベートを解除すると、アカウントが無効になり、無効化されたユーザーアカウントを使用して認証できません。無効化されたアカウントを使用するユーザーは、IdM にログインできず、Kerberos などの IdM サービスを使用できずタスクも実行できません。

無効化されたユーザーアカウント自体は、IdM に存在しており、関連の情報はすべて変更されません。preserved ユーザーアカウントとは異なり、無効化されたユーザーアカウントは **active** の状態のまま保持されるので、**ipa user-find** コマンドを実行すると、出力に表示されます。以下に例を示します。

```
$ ipa user-find
...
  User login: user
  First name: User
  Last name: User
  Home directory: /home/user
  Login shell: /bin/sh
  UID: 1453200009
  GID: 1453200009
  Account disabled: True
  Password: False
  Kerberos keys available: False
...
```

無効化されたユーザーアカウントは再度有効にすることができます。



注記

ユーザーアカウントを無効化した後に、Kerberos TGT や他のチケットの期限が切れるまで既存の接続は有効な状態のままです。チケットの期限が切れると、ユーザーはチケットを更新することができません。

Web UI でのユーザーアカウントの有効化および無効化

1. **Identity → Users** タブを選択します。
2. **Active users** 一覧から対象のユーザーを選択して、**Disable** または **Enable** をクリックします。

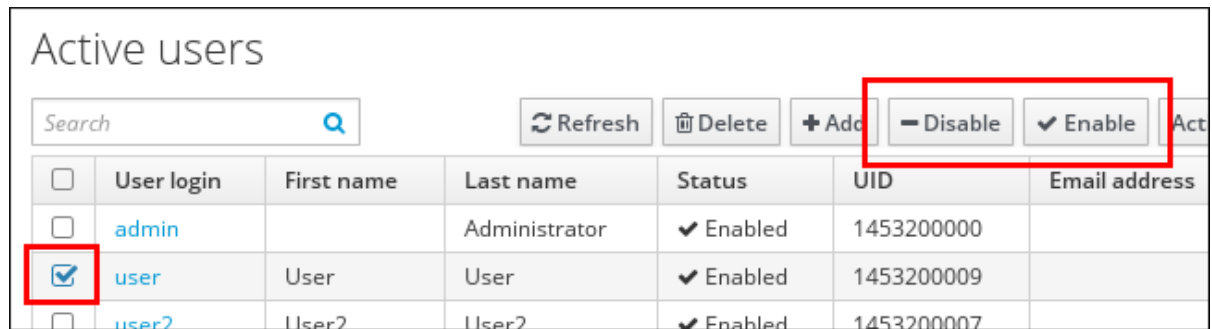


図11.12 ユーザーアカウントの無効化または有効化

コマンドラインからのユーザーアカウントの無効化および有効化

ユーザーアカウントを無効化するには、**ipa user-disable** コマンドを使用します。

```
$ ipa user-disable user_login
-----
Disabled user account "user_login"
-----
```

ユーザーアカウントを有効化するには、**ipa user-enable** コマンドを使用します。

```
$ ipa user-enable user_login
-----
Enabled user account "user_login"
-----
```

11.5. 管理者以外のユーザーによるユーザーエントリーの管理許可

デフォルトでは **admin** ユーザーのみがユーザーライフサイクルの管理やユーザーアカウントの有効化/無効化が可能です。管理者以外の別ユーザがこの操作をできるようにするには、新規ロールを作成して、このロールに適切なパーミッションを追加し、管理者以外のユーザーにこのロールを割り当てます。

デフォルトでは、IdM にはユーザーアカウントの管理に関する以下の権限が含まれます。

Modify Users and Reset passwords

この権限には、さまざまなユーザー属性を変更するパーミッションが含まれます。

User Administrators

この権限は、active ユーザーの追加、active 以外のユーザーのアクティベート、ユーザーの削除、ユーザー属性やその他のパーミッションの変更が含まれます。

Stage User Provisioning

この権限には、stage ユーザーを追加するパーミッションが含まれます。

Stage User Administrator

以下の権限には、stage ユーザーの追加、ライフサイクルの状態の間におけるユーザーの移動などライフサイクルでの操作を実行するパーミッションが含まれます。ただし、active の状態にユーザーを移動するパーミッションは含まれません。

ロール、パーミッション、権限の定義に関する情報は「[ロールベースのアクセス制御の定義](#)」を参照してください。

異なるユーザーに対するさまざまなユーザー管理操作の実行許可

ユーザーアカウントの管理に関するさまざまな権限を各種ユーザーに追加することができます。たとえば、以下のように従業員のアカウントエントリとアクティベーションの権限を区別することができます。

- *stage user administrator* ユーザーを 1 つ設定する。このユーザーは、IdM に 今後入社する従業員を stage ユーザーとして追加できるが、アクティベートはできません。
- *security administrator* として別のユーザーを 1 つ設定する。このユーザーは、就業日初日に従業員の認証情報を確認後に、stage ユーザーをアクティベートすることができます。

ユーザーが特定のユーザー管理の操作を実行できるようにするには、必要な権限を指定した新規ロールを作成し、そのロールを対象のユーザーに割り当てます。

例11.1 管理者以外のユーザーによる **stage** ユーザーの追加許可

以下の例では、新規の stage ユーザーの追加のみが可能で、他の stage ユーザー管理操作を実行できないユーザーを作成する方法を説明します。

1. ロールベースのアクセス制御を管理可能な **admin** ユーザーまたは別のユーザーとしてログインします。

```
$ kinit admin
```

2. 新規カスタムロールを作成して、stage ユーザーの追加を管理します。

- a. **System Provisioning** ロールを作成します。

```
$ ipa role-add --desc "Responsible for provisioning stage
users" "System Provisioning"
-----
Added role "System Provisioning"
-----
Role name: System Provisioning
Description: Responsible for provisioning stage users
```

- b. **Stage User Provisioning** の権限をロールに割り当てます。この権限により、stage ユーザーが追加できるようになります。

```
$ ipa role-add-privilege "System Provisioning" --
privileges="Stage User Provisioning"
Role name: System Provisioning
Description: Responsible for provisioning stage users
Privileges: Stage User Provisioning
-----
Number of privileges added 1
-----
```

3. 管理者以外のユーザーに stage ユーザーを追加する権限を割り当てます。

- a. 管理者以外のユーザーが存在しない場合には、新規ユーザーを作成します。以下の例では **stage_user_admin** という名前のユーザーです。

```
$ ipa user-add stage_user_admin --password
First name: first_name
Last name: last_name
Password:
Enter password again to verify:
...
```

- b. **stage_user_admin** ユーザーに **System Provisioning** ロールを割り当てます。

```
$ ipa role-add-member "System Provisioning" --
users=stage_user_admin
Role name: System Provisioning
Description: Responsible for provisioning stage users
Member users: stage_user_admin
Privileges: Stage User Provisioning
-----
Number of members added 1
-----
```

- c. **System Provisioning** ロールが正しく設定されていることを確認するには、**ipa role-show** コマンドを使用してロールの設定を表示します。

```
$ ipa role-show "System Provisioning"
-----
1 role matched
-----
Role name: System provisioning
Description: Responsible for provisioning stage users
Member users: stage_user_admin
Privileges: Stage User Provisioning
-----
Number of entries returned 1
-----
```

4. **stage_user_admin** ユーザーとして新しい stage ユーザーが追加されているかをテストします。

- a. **stage_user_admin** としてログインします。以前の手順で新規ユーザーとして **stage_user_admin** を作成した場合は、IdM により **admin** が設定した初期パスワードを変更するように求められます。

```
$ kinit stage_user_admin
Password for stage_user_admin@EXAMPLE.COM:
Password expired. You must change it now.
Enter new password:
Enter it again:
```

- b. **admin** の Kerberos チケットが **stage_user_admin** の Kerberos チケットに置き換えられていることを確認するには **klist** ユーティリティーを使用します。

```
$ klist
Ticket cache: KEYRING:persistent:0:krb_ccache_xIlCQDW
Default principal: stage_user_admin@EXAMPLE.COM
```

Valid starting	Expires	Service principal
02/25/2016 11:42:20	02/26/2016 11:42:20	krbtgt/EXAMPLE.COM

c. 新規 stage ユーザーを追加します。

```
$ ipa stageuser-add stage_user
First name: first_name
Last name: last_name
ipa: ERROR: stage_user: stage user not found
```



注記

stage ユーザーの追加後に IdM からエラーが報告されるのは想定内です。**stage_user_admin** は、stage ユーザーの追加のみが可能で、stage ユーザーの情報の表示はできません。そのため、新たに追加された **stage_user** の設定サマリーを表示する代わりに、IdM はエラーを表示します。

stage_user_admin ユーザーは、stage ユーザーの情報を表示できません。そのため、**stage_user_admin** でログインして、新しい **stage_user** ユーザーに関する情報を表示しようとすると、失敗してしまいます。

```
$ ipa stageuser-show stage_user
ipa: ERROR: stage_user: stage user not found
```

stage_user に関する情報を表示するには **admin** でログインしてください。

```
$ kinit admin
Password for admin@EXAMPLE.COM:
$ ipa stageuser-show stage_user
User login: stage_user
First name: Stage
Last name: User
...
```

11.6. ユーザーおよびグループへの外部プロビジョニングシステムの使用

Identity Management (IdM) は、ID 管理用の外部ソリューションを使用して IdM でユーザーおよびグループ ID がプロビジョニングされるように、お使いの環境を設定するサポートをします。以下のセクションは、このような設定の例について説明します。例には以下が含まれます。

- 「外部プロビジョニングシステムで利用されるようにユーザーアカウントを設定する手順」
- 「IdM が自動的に stage ユーザーアカウントをアクティベートするように設定する手順」
- 「外部プロビジョニングシステムの LDAP プロバイダーが IdM のアイデンティティーを管理するように設定する手順」

11.6.1. 外部プロビジョニングシステムで利用されるようにユーザーアカウントを設定する手順

以下の手順では、外部プロビジョニングシステムが IdM ユーザーアカウントを使用するように設定する方法を説明します。適切なパスワードポリシーが指定されたグループにアカウントを追加すると、外部プロビジョニングシステムが IdM でユーザーのプロビジョニングを管理できるようになります。

1. stage ユーザーの追加権限のあるユーザー **provisionator** を作成します。外部プロビジョニングシステムがこのユーザーアカウントを使用して新しい stage ユーザーを追加します。

- a. **provisionator** ユーザーアカウントを追加します。

```
$ ipa user-add provisionator --first=provisioning --last=account
--password
```

- b. **provisionator** ユーザーに必要な権限を割り当てます。

stage ユーザーの追加を管理する **System Provisioning** というカスタムロールを作成します。

```
$ ipa role-add --desc "Responsible for provisioning stage users"
"System Provisioning"
```

Stage User Provisioning の権限をロールに割り当てます。この権限により、stage ユーザーが追加できるようになります。

```
$ ipa role-add-privilege "System Provisioning" --
privileges="Stage User Provisioning"
```

provisionator ユーザーをロールに追加します。

```
$ ipa role-add-member --users=provisionator "System Provisioning"
```

2. ユーザーアカウントの管理権限を持つ **activator** ユーザーを作成します。このユーザーアカウントを使用すると、外部プロビジョニングシステムにより追加された stage ユーザーが自動的にアクティベートされます。

- a. **activator** ユーザーアカウントを追加します。

```
$ ipa user-add activator --first=activation --last=account --
password
```

- b. **activator** ユーザーに必要な権限を割り当てます。

デフォルトの **User Administrator** ロールにこのユーザーを追加します。

```
$ ipa role-add-member --users=activator "User Administrator"
```

3. サービスおよびアプリケーションアカウントのユーザーグループを作成します。

```
$ ipa group-add service-accounts
```

4. グループのパスワードポリシーを更新します。以下のポリシーは、アカウントのパスワードの期限切れやロックアウトがされないようにしますが、複雑なパスワードを求めることでリスクの可能性を低減します。

```
$ ipa pwpolicy-add service-accounts --maxlife=10000 --minlife=0 --
history=0 --minclasses=4 --minlength=20 --priority=1 --maxfail=0 --
failinterval=1 --lockouttime=0
```

5. サービスおよびアプリケーションアカウントのグループにプロビジョニングアカウントとアクティベーションアカウントを追加します。

```
$ ipa group-add-member service-accounts --users=
{provisionator,activator}
```

6. ユーザーアカウントのパスワードを変更します。

```
$ kpasswd provisionator
$ kpasswd activator
```

新規の IdM ユーザーのパスワードはすぐに失効するため、パスワードを変更する必要があります。

追加リソース:

- 新規ユーザーの追加に関する情報は、[「stage または Active ユーザーの追加」](#)を参照してください。
- 他のユーザーアカウントの管理に必要な権限をユーザーに割り当てる方法については [「管理者以外のユーザーによるユーザーエントリーの管理許可」](#)を参照してください。
- IdM パスワードポリシーの管理に関する詳しい情報は[27章 パスワードポリシーの定義](#)を参照してください。

11.6.2. IdM が自動的に **stage** ユーザーアカウントをアクティベートするように設定する手順

以下の手順では、stage ユーザーをアクティベートするスクリプトを作成する方法を説明します。システムは指定した間隔で自動的にスクリプトを実行します。これにより、新規ユーザーアカウントを自動的にアクティベートし、作成後すぐに利用できるようにします。



重要

以下の手順では、スクリプトを IdM に追加する前に、新規ユーザーアカウントの検証が必要ないとの想定です。たとえば、外部プロビジョニングシステムの所有者によりユーザーがすでに検証済みの場合には、検証は必要ありません。

IdM サーバーの 1 つで、アクティベーションプロセスを有効化するだけで十分です。

1. アカウントのアクティベート用に keytab ファイルを生成します。

```
# ipa-getkeytab -s example.com -p "activator" -k /etc/krb5.ipa-
activation.keytab
```

複数の IdM サーバーでアクティベーションプロセスを有効にするには、1 つのサーバーでのみ keytab ファイルを作成し、他のサーバーにその keytab ファイルをコピーします。

2. 以下の内容を含む **/usr/local/sbin/ipa-activate-all** スクリプトを作成して全ユーザーをアクティベートします。

```
#!/bin/bash

kinit -k -i activator

ipa stageuser-find --all --raw | grep " uid:" | cut -d ":" -f 2 |
while read uid; do ipa stageuser-activate ${uid}; done
```

3. **ipa-activate-all** スクリプトのパーミッションと所有者を編集して、実行可能なファイルに変更します。

```
# chmod 755 /usr/local/sbin/ipa-activate-all
# chown root:root /usr/local/sbin/ipa-activate-all
```

4. **systemd** のユニットファイルである **/etc/systemd/system/ipa-activate-all.service** を作成して以下の内容を追加します。

```
[Unit]
Description=Scan IdM every minute for any stage users that must be
activated

[Service]
Environment=KRB5_CLIENT_KTNAME=/etc/krb5.ipa-activation.keytab
Environment=KRB5CCNAME=FILE:/tmp/krb5cc_ipa-activate-all
ExecStart=/usr/local/sbin/ipa-activate-all
```

5. **systemd** タイマーである **/etc/systemd/system/ipa-activate-all.timer** を作成して以下の内容を追加します。

```
[Unit]
Description=Scan IdM every minute for any stage users that must be
activated

[Timer]
OnBootSec=15min
OnUnitActiveSec=1min

[Install]
WantedBy=multi-user.target
```

6. **ipa-activate-all.timer** を有効化します。

```
# systemctl enable ipa-activate-all.timer
```

追加リソース:

- **systemd** ユニットファイルに関する詳しい情報は『システム管理者のガイド』の「[システムのユニットファイルの作成および変更](#)」の章を参照してください。

11.6.3. 外部プロビジョニングシステムの **LDAP** プロバイダーが **IdM** のアイデンティティーを管理するように設定する手順

以下の章では、さまざまなユーザーおよびグループの管理オプションのテンプレートを紹介します。これらのテンプレートを使用すると、プロビジョニングシステムの LDAP プロバイダーが IdM ユーザーアカウントを管理できるように設定できます。たとえば、従業員が退職した後にユーザーアカウントを無効にするようにシステムを設定することができます。

LDAP を使用したユーザーアカウントの管理

下層の Directory Server データベースを編集して、新規ユーザーエントリーの追加、既存のエントリーの変更、異なるライフサイクルの状態間でのユーザーの移動、ユーザーの削除ができます。データベースを編集するには **ldapmodify** ユーティリティを使用します。

以下の LDIF 形式のテンプレートには、**ldapmodify** を使用して変更する属性に関する情報が含まれます。例に関する詳しい手順は [例11.2「**ldapmodify** での stage ユーザーへの追加](#)」 および [例11.3「**ldapmodify** でのユーザーの保存](#)」を参照してください。

新規 stage ユーザーの追加

UID および GID が自動的に割り当てられたユーザーの追加

```
dn: uid=user_login,cn=staged
users,cn=accounts,cn=provisioning,dc=example,dc=com
changetype: add
objectClass: top
objectClass: inetorgperson
uid: user_login
sn: surname
givenName: first_name
cn: full_name
```

UID および GID が静的に割り当てられたユーザーの追加

```
dn: uid=user_login,cn=staged
users,cn=accounts,cn=provisioning,dc=example,dc=com
changetype: add
objectClass: top
objectClass: person
objectClass: inetorgperson
objectClass: organizationalperson
objectClass: posixaccount
uid: user_login
uidNumber: UID_number
gidNumber: GID_number
sn: surname
givenName: first_name
cn: full_name
homeDirectory: /home/user_login
```

stage ユーザーの追加時に IdM オブジェクトクラスを指定する必要はありません。IdM は、ユーザーのアクティベート後に、これらのクラスを自動的に追加します。

作成したエントリーの識別名 (DN) は **uid=user_login** で開始する必要がある点に注意してください。

既存ユーザーの変更

ユーザーを変更する前に、ユーザーの識別名 (DN) をユーザーのログインで検索して取得します。以下の例では、*user_allowed_to_read* ユーザーは、ユーザーおよびグループの情報を読み込む権限があり、*password* はユーザーのパスワードとなっています。

```
# ldapsearch -LLL -x -D
"uid=user_allowed_to_read,cn=users,cn=accounts,dc=example, dc=com" -w
"password" -H ldap://server.example.com -b "cn=users, cn=accounts,
dc=example, dc=com" uid=user_login
```

ユーザーの属性を変更する方法:

```
dn: distinguished_name
changetype: modify
replace: attribute_to_modify
attribute_to_modify: new_value
```

ユーザーを無効化する方法:

```
dn: distinguished_name
changetype: modify
replace: nsAccountLock
nsAccountLock: TRUE
```

ユーザーを有効化する方法:

```
dn: distinguished_name
changetype: modify
replace: nsAccountLock
nsAccountLock: FALSE
```

ユーザーを保存する方法:

```
dn: distinguished_name
changetype: modrdn
newrdn: uid=user_login
deleteoldrdn: 0
newsuperior: cn=deleted users,cn=accounts,cn=provisioning,dc=example
```

nssAccountLock 属性を更新しても、stage および preserved ユーザーには影響はありません。更新操作が正常に完了しても、属性の値は **nssAccountLock: TRUE** のままです。

新規グループの作成

新規グループの作成方法:

```
dn: cn=group_distinguished_name,cn=groups,cn=accounts,dc=example,dc=com
changetype: add
objectClass: top
objectClass: ipaobject
objectClass: ipausergroup
objectClass: groupofnames
objectClass: nestedgroup
objectClass: posixgroup
cn: group_name
gidNumber: GID_number
```

グループの変更

グループを変更する前に、グループ名で検索してグループの識別名 (DN) を取得します。

```
# ldapsearch -Y GSSAPI -H ldap://server.example.com -b
"cn=groups,cn=accounts,dc=example,dc=com" "cn=group_name"
```

既存のグループの削除方法:

```
dn: group_distinguished_name
changetype: delete
```

グループにメンバーを追加する方法:

```
dn: group_distinguished_name
changetype: modify
add: member
member: uid=user_login,cn=users,cn=accounts,dc=example,dc=com
```

グループからメンバーを削除する方法:

```
dn: distinguished_name
changetype: modify
delete: member
member: uid=user_login,cn=users,cn=accounts,dc=example,dc=com
```

stage または preserved ユーザーをグループには追加しないでください。更新が正常に完了した場合でも、ユーザーはグループのメンバーとして更新されません。active ユーザーだけがグループに所属します。

例11.2 ldapmodify での stage ユーザーへの追加

標準の **interorgperson** オブジェクトクラスを使用して **stageuser** ユーザーを追加する方法:

1. **ldapmodify** を使用してユーザーを追加します。

```
# ldapmodify -Y GSSAPI
SASL/GSSAPI authentication started
SASL username: admin@EXAMPLE
SASL SSF: 56
SASL data security layer installed.
dn: uid=stageuser,cn=staged
users,cn=accounts,cn=provisioning,dc=example
changetype: add
objectClass: top
objectClass: inetorgperson
cn: Stage
sn: User

adding new entry "uid=stageuser,cn=staged
users,cn=accounts,cn=provisioning,dc=example"
```

2. ステージエントリーの内容を検証し、必要とされる POSIX 属性がプロビジョニングシステムに追加され、ステージエントリーのアクティベートの準備ができていることを確認します。新規 stage ユーザーの LDAP 属性を表示するには、**ipa stageuser-show --all -**

-raw コマンドを使用します。ユーザーは **nsaccountlock** 属性で明示的に無効化されていることに注意してください。

```
$ ipa stageuser-show stageuser --all --raw
dn: uid=stageuser,cn=staged
users,cn=accounts,cn=provisioning,dc=example
uid: stageuser
sn: User
cn: Stage
has_password: FALSE
has_keytab: FALSE
nsaccountlock: TRUE
objectClass: top
objectClass: inetorgperson
objectClass: organizationalPerson
objectClass: person
```

例11.3 ldapmodify でのユーザーの保存

LDAP **modrdn** 操作を使用して **user** を保存する方法:

1. **ldapmodify** ユーティリティを使用してユーザーエントリーを変更します。

```
$ ldapmodify -Y GSSAPI
SASL/GSSAPI authentication started
SASL username: admin@EXAMPLE
SASL SSF: 56
SASL data security layer installed.
dn: uid=user1,cn=users,cn=accounts,dc=example
changetype: modrdn
newrdn: uid=user1
deleteoldrdn: 0
newsuperior: cn=deleted
users,cn=accounts,cn=provisioning,dc=example

modifying rdn of entry "uid=user1,cn=users,cn=accounts,dc=example"
```

2. オプションで、すべての preserved ユーザーを表示して、このユーザーが保存されたことを確認します。

```
$ ipa user-find --preserved=true
-----
1 user matched
-----
  User login: user1
  First name: first_name
  Last name: last_name
  ...
-----
Number of entries returned 1
-----
```

第12章 ホストの管理

DNS と Kerberos は両方とも、初期クライアント設定の一部として設定されています。DNS と Kerberos は、マシンを IdM ドメイン内に配備し、接続先の IdM サーバーを識別できるようにするサービスなので、この設定が必要になります。初期設定後 IdM には、ドメインサービスの変更や IT 環境の変更、クライアントホスト名の変更など、Kerberos や証明書、および DNS サービスに影響するマシン自体の変更に対応するために DNS と Kerberos サービスの両方を管理するツールがあります。

本章では、クライアントマシンに直接関連する以下の ID サービスの管理方法について説明します。

- DNS エントリーおよび設定
- マシン認証
- (ドメインサービスに影響する) ホスト名の変更

12.1. ホスト、サービス、およびマシン ID と認証

登録プロセスの基本的な役割は、IdM ディレクトリー内でクライアントマシン用の **ホスト** エントリーを作成することです。このホストエントリーは、他のホストとドメイン内のサービスの関係を確立するために使用されます。この関係は、権限と管理をドメイン内のホストへ**委任**するプロセスの一部です。

ホストエントリーには、IdM 内のクライアントについて以下のような情報のすべてが含まれます。

- ホストに関連付けられたサービスエントリー
- ホストおよびサービスプリンシパル
- アクセス制御ルール
- 物理的位置やオペレーティングシステムなどのマシンについての情報

ホスト上で実行されるサービスには、IdM ドメインに属するものもあります。Kerberos プリンシパルまたは SSL 証明書のいずれか (またはこれら両方) を保存することができるサービスは、IdM サービスとして設定することができます。IdM ドメインにサービスを追加すると、そのサービスはドメインから SSL 証明書や keytab を要求することができます。(証明書の公開鍵のみがサービスレコードに保存されます。秘密鍵はサービスのローカルになります。)

An IdM ドメインは、共通の ID 情報、ポリシー、共有 サービスで、マシン間に共通性を確立します。あるドメインに所属しているマシンは、そのドメインのクライアントとして機能するので、そのドメインが提供するサービスを使用できます。(1章 [Red Hat Identity Management について](#) の説明にあるように) An IdM ドメインは特に、以下の 3 つの主要サービスをマシンに提供します。

- DNS
- Kerberos
- 証明書管理

マシンは、IdM が管理する別の ID として扱われます。クライアントは DNS を使って IdM サーバー、サービス、およびドメインメンバーを識別します。これらはユーザー ID のように、IdM サーバーの 389 Directory Server インスタンスに保存されます。マシンはユーザーのように、Kerberos または証明書を使ってマシンの ID を確認して、ドメインに対して認証することができます。

マシン側からは、これらのドメインサービスにアクセスする以下のようなタスクが実行可能です。

- DNS ドメインへの参加 (マシン登録)

- DNS エントリーおよびゾーンの管理
- マシン認証の管理

IdM での認証には、ユーザーのほかにマシンも含まれます。マシン認証は、IdM がマシンを信頼して、そのマシン上にインストールされているクライアントソフトウェアからの IdM 接続を受け付けるために必要なものです。クライアントを認証すると、IdM サーバーはクライアントの要求に対応できるようになります。IdM は、マシン認証で以下の 3 つのアプローチに対応しています。

- SSH 鍵。ホスト用の SSH 公開鍵が作成され、ホストエントリーにアップロードされます。そこから System Security Services Daemon (SSSD) は、IdM を ID プロバイダーとして使用し、OpenSSH や他のサービスと連携して Identity Management に一元化して設置されている公開鍵を参照します。これは「[ホストの公開 SSH キーの管理](#)」で説明しています。
- キーテーブル (または *keytabs*。対称キーで、多少ユーザーパスワードに類似) およびマシン証明書。Kerberos チケットは Kerberos サービスの一部として生成され、ポリシーはサーバーが定義します。初期の Kerberos チケットの提供、Kerberos 証明書の更新、Kerberos セッションの破棄はすべて IdM サービスによって処理されます。Kerberos の管理は [28 章 Kerberos ドメインの管理](#) で説明されています。
- マシンの証明書。この場合、IdM サーバーの認証局が発行し、IdM の Directory Server に保存されている SSL 証明書をマシンは使用します。証明書はマシンに送信され、サーバーに対して認証する際に提示されます。クライアントでは、証明書は **certmonger** と呼ばれるサービスが管理します。

12.2. ホストエントリー設定のプロパティー

ホストエントリーには、ホストの物理的な場所や MAC アドレス、鍵および証明書など、システム設定以外の情報を含めることができます。

ホストエントリーを手動で作成する場合は、これらの情報は設定可能です。手動作成でない場合は、ホストをドメインに登録した後で、情報を追加する必要があります。

表12.1 ホスト設定のプロパティー

UI フィールド	コマンドラインオプション	Description
Description	--desc=description	ホストの説明。
Locality	--locality=locality	ホストの位置情報
Location	--location=location	データセンターラックなど、ホストの位置情報
Platform	--platform=string	ホストのハードウェアまたはアーキテクチャー
Operating system	--os=string	ホストのオペレーティングシステムおよびバージョン

UI フィールド	コマンドラインオプション	Description
MAC address	--macaddress=address	ホストの MAC アドレス。これは複数値の属性です。MAC アドレスは、NIS プラグインがホスト用の NIS ethers マップを作成するために使用します。
SSH public keys	--sshpubkey=string	ホスト用の SSH 公開鍵。これは複数値の属性なので、複数の鍵を設定できます。
Principal name (not editable)	--principalname=principal	ホストの Kerberos プリンシパル名。これは -p オプションで明示的に別のプリンシパルを設定しなければ、クライアントのインストール中にホスト名でデフォルト設定されます。これはコマンドラインツールを使用すると変更可能ですが、UI では変更できません。
Set One-Time Password	--password=string	一括登録で使用可能なホストのパスワードを設定します。
-	--random	一括登録で使用するランダムなパスワードを生成します。
-	--certificate=string	ホストの証明書プロブ。
-	--updatedns	これは IP アドレス変更時にホストが DNS エントリーを動的に更新できるかどうかを設定します。

12.3. ホストエントリーの追加

12.3.1. Web UI でのホストエントリーの追加

1. **Identity** タブを開き、**Hosts** サブタブを選択します。
2. ホスト一覧上部にある **Add** をクリックします。

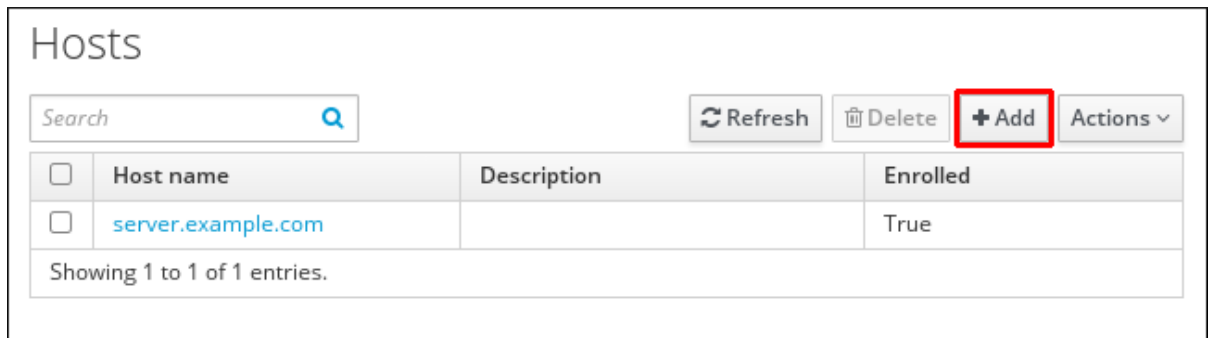


図12.1 ホストエントリーの追加

- マシン名を記入し、ドロップダウンリストの設定済みゾーンからドメインを選択します。ホストに既に静的 IP アドレスが割り当てられている場合は、ホストエントリーにそのアドレスを含めることで DNS エントリーが完全に作成されます。

図12.2 ホスト追加ウィザード

DNS ゾーンは IdM で作成可能で、これは「[Master DNS ゾーンの追加と削除](#)」で説明しています。IdM サーバーが DNS サーバーを管理しない場合は、ゾーンは通常のテキストフィールドのようにメニューエリアで手動で入力することができます。



注記

ホスト名が解決できない場合でも、**Force** チェックボックスを選択してホスト DNS レコードを追加してください。

これは DHCP を使用し、静的 IP アドレスを持たないホストで便利なものです。これにより、IdM DNS サービスにプレースホルダーエントリーが作成されます。DNS サービスが動的にレコードを更新すると、ホストの現行の IP アドレスが削除され、DNS レコードが更新されます。

- Add and Edit** をクリックして、拡張エントリーページに移動し、属性情報をさらに入力します。ホストエントリーには、ホストのハードウェアや物理的な場所に関する情報を含めることができます。

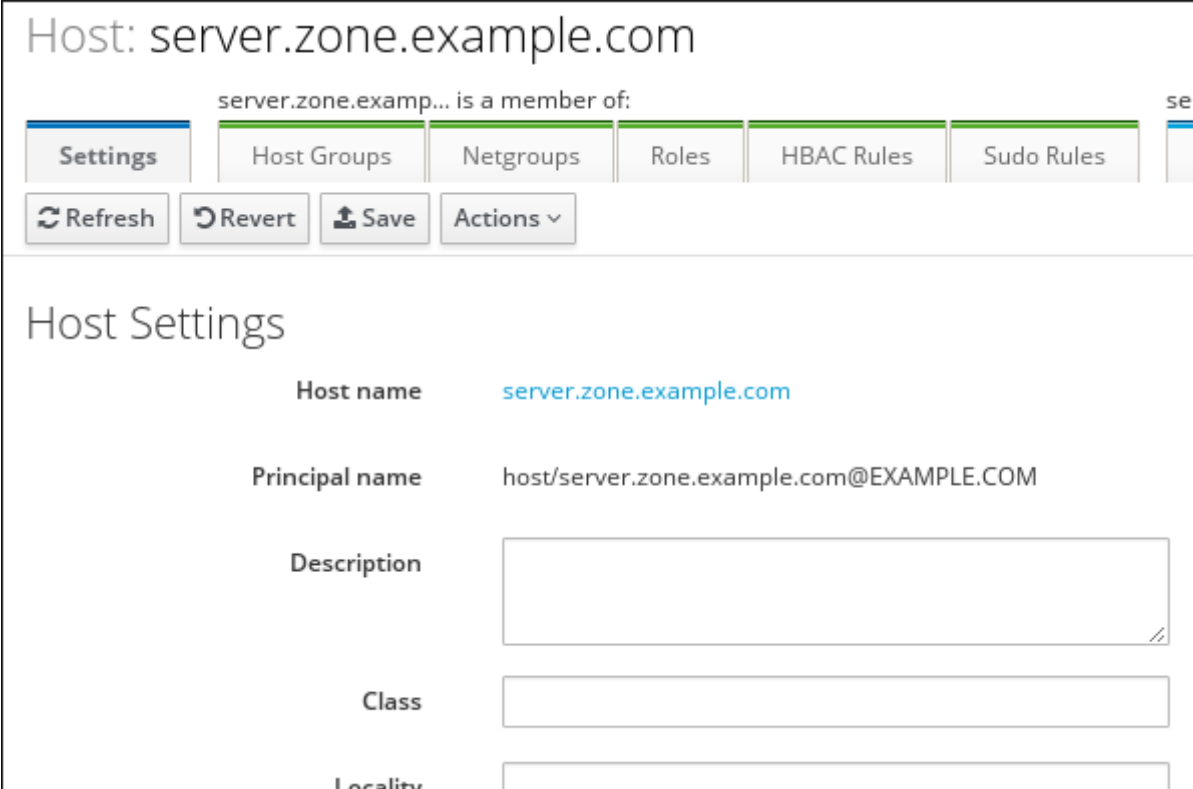


図12.3 展開されたエントリーページ

12.3.2. コマンドラインでのホストエントリーの追加

ホストエントリーは **host-add** コマンドを使って作成されます。このコマンドは、ホストエントリーを IdM Directory Server に追加します。**host-add** の全オプションは、**ipa host man** ページに記載されています。このコマンドの最も基本的な操作では、クライアントを Kerberos レalmに追加し、IdM LDAP サーバーにエントリーを作成するために、クライアントのホスト名のみが必要となります。

```
$ ipa host-add client1.example.com
```

IdM サーバーが DNS を管理するよう設定されている場合は、**--ip-address** および **--force** のオプションを使用してホストを DNS リソースレコードに追加することができます。

例12.1 静的 IP アドレスのホストエントリーの作成

```
$ ipa host-add --force --ip-address=192.168.166.31 client1.example.com
```

ホストに静的 IP アドレスがない、またはクライアントの設定時に IP アドレスが分からないことがよくあります。たとえば、ノートパソコンが Identity Management クライアントとして設定されていても、その設定時には IP アドレスが分からない場合などです。その場合でも DHCP を使用するホストは、**--force** を使用して DNS エントリーを設定することができます。これにより、IdM DNS サービスにプレースホルダーエントリーが作成されます。DNS サービスが動的にレコードを更新すると、ホストの現行の IP アドレスが削除され、DNS レコードが更新されます。

例12.2 DHCP のホストエントリーの作成

```
$ ipa host-add --force client1.example.com
```

host-del コマンドを使用するとホストレコードが削除されます。IdM ドメインが DNS を使用している場合は、**--updatedns** オプションはホスト関連のレコードも DNS から削除します。

```
$ ipa host-del --updatedns client1.example.com
```

12.4. ホストエントリーの無効化および再有効化

アクティブなホストは、ドメイン内の他のサービスやホスト、ユーザーからアクセス可能です。アクティビティーからホストを削除する必要がある場合もあります。ただし、ホストを削除するとエントリーや関連する設定もすべて完全に削除されてしまいます。

12.4.1. ホストエントリーの無効化

ホストを無効にすると、ホストをドメインから永久に削除することなくドメインユーザーがホストにアクセスすることを防ぎます。これは **host-disable** コマンドを使用することで実行できます。

例を示します。

```
[jsmith@ipaserver ~]$ kinit admin
[jsmith@ipaserver ~]$ ipa host-disable server.example.com
```



重要

ホストエントリーを無効にすると、ホストだけでなくそのホスト上で設定されているすべてのサービスが無効になります。

12.4.2. ホストの再有効化

ホストを無効にすると、実質的に現行のアクティブな keytab を強制終了します。keytab を削除すると、ホストの設定エントリーを変更せずにホストを IdM ドメインから削除することになります。

ホストを再度有効にするには、**ipa-getkeytab** コマンドを使用するだけです。**-s** オプションはどの IdM サーバーに keytab を要求するかを設定し、**-p** はプリンシパル名を提示し、**-k** では keytab を保存するファイルを提供します。

新規のホスト keytab を要求する場合は、以下のようになります。

```
[jsmith@ipaserver ~]$ ipa-getkeytab -s ipaserver.example.com -p
host/server.example.com -k /etc/krb5.keytab -D
fqdn=server.example.com,cn=computers,cn=accounts,dc=example,dc=com -w
password
```

ipa-getkeytab コマンドをアクティブな IdM クライアントまたはサーバーで実行する場合は、LDAP 認証情報 (**-D** および **-w**) なしで実行可能です。IdM ユーザーは、Kerberos 認証情報を使ってドメインに認証します。無効となっているホスト上で直接コマンドを実行するには、LDAP 認証情報を提供して IdM サーバーに認証を行います。認証情報は、再有効化を行なっているホストまたはサーバーに対応するものにしてください。

12.5. ホストの公開 SSH キーの管理

OpenSSH は、公開キーを使ってホストに対して認証を行います。あるマシンが別のマシンに対してア

クセスを試みると、キーのペアを提示します。ホストが最初に認証する際は、ターゲットマシンの管理者は、この要求を手動で認証する必要があります。するとマシンはホストの公開キーを **known_hosts** ファイルに保存します。リモートのマシンがターゲットマシンにアクセスを再度試みると、ターゲットマシンは **known_hosts** ファイルをチェックして、認証済みホストに自動的にアクセスを許可します。

このシステムには、以下のような問題があります。

- **known_hosts** ファイルは、ホストエントリーをホスト IP アドレス、ホスト名、およびキーという 3 項目で保存します。IP アドレスが変更されたり (仮想環境やデータセンターでは一般的)、キーが更新されたりすると、このファイルはすぐに無効になってしまいます。
- SSH キーは、環境内の全マシンに手動かつ個別に配布する必要があります。
- 管理者は設定に追加するホストキーを許可する必要がありますが、ホストまたはキー発行者を適切に検証することが困難なことから、セキュリティ問題が発生する可能性があります。

Red Hat Enterprise Linux では、System Security Services Daemon (SSSD) が SSH キーをキャッシュ、取得するよう設定して、アプリケーションやサービスがホストキーを 1 か所で探せるようにできます。SSSD は Identity Management を ID 情報プロバイダーとして使用できるので、Identity Management をキーの汎用かつ集中化リポジトリとすることができます。このため管理者は、ホスト SSH キーの配布や更新、検証を心配する必要がありません。

12.5.1. SSH 鍵の形式

キーを IdM エントリーにアップロードする際には、キーの形式は [OpenSSH-style key](#) か生の [RFC 4253-style blob](#) にすることができます。RFC 4253-style key は、IdM LDAP サーバーにインポート、保存される前に、自動的に OpenSSH-style key に変換されます。

IdM サーバーは、アップロードされたキーブロッブから、RSA または DSA キーといったキーのタイプを識別することができます。しかし、`~/.ssh/known_hosts` のようなキーファイルでは、キーのエントリーはサーバーのホスト名および IP アドレス、キーのタイプ、最後にキー自体で識別されます。例を示します。

```
host.example.com,1.2.3.4 ssh-rsa AAA...ZZZ==
```

これはユーザーの公開キーエントリーとは多少異なります。ユーザーの公開キーエントリーの要素は、**type key== comment** という順序になります。

```
"ssh-rsa ABCD1234...== ipaclient.example.com"
```

キーファイルからの 3 要素はすべて、ホストエントリーにアップロードし、表示することができます。その場合、`~/.ssh/known_hosts` ファイルからのホスト公開キーエントリーがユーザーキーの形式 **type key== comment** に一致するように順序を変える必要があります。

```
ssh-rsa AAA...ZZZ== host.example.com,1.2.3.4
```

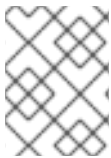
キータイプは公開キーのコンテンツから自動的に判断されます。個別のキーの識別を容易にするコメントはオプションになります。必須要素は、公開キーブロッブ自体のみとなります。

12.5.2. ipa-client-install および OpenSSH

ipa-client-install スクリプトはデフォルトで、OpenSSH サーバーと IdM クライアントマシン上のクライアントを設定します。また SSSD がホストおよびユーザーキーのキャッシングを実行するように設定します。実質的には、クライアントを設定するだけで、ホストがキーキャッシングおよび取得

のために SSSD、OpenSSH、および Identity Management を使用するすべての必須設定が実行されます。

SSH サービスがクライアントインストール時に有効にされている場合 (これがデフォルト)、**ssh** サービスの初回起動時に RSA キーが作成されます。



注記

ipa-client-install を使用して IdM クライアントとしてマシンを追加する場合、クライアントには RSA および DSS という 2 つの SSH キーが作成されます。

ipa-client-install コマンドには **--ssh-trust-dns** という設定オプションもあり、これを一緒に実行すると、OpenSSH がキーフィンガープリントを保存している IdM DNS レコードを信頼するように自動設定します。

別の方法では、**--no-sshd** を使ってクライアントインストール時に OpenSSH を無効にすることができます。これにより、インストールスクリプトは OpenSSH サーバーを設定できなくなります。

--no-dns-sshfp というもうひとつのオプションは、ホストが自身の DNS エントリーで DNS SSHFP レコードを作成できないようにします。これは **--no-sshd** オプションとの併用も可能です。

12.5.3. ホスト SSH 鍵の Web UI でのアップロード

1. ホストのキーは、`~/.ssh/known_hosts` から取得できます。例を示します。

```
server.example.com,1.2.3.4 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEApxjBvSFskTU0WQW4e0weeo0DZZ08F9Ud21xLLy6F
0hzwpxFGIyxvXZ52+siHBHbbqGL5+14N7UvElruyslIHx9LYUR/pPKSMXCGyboLy5aTN
l50Q5EHwrhVnFDIKXkvp45945R7SKYCUtRumm0Iw6wq0XD4o+ILeVbV3wmcB1bXs36Zv
C/M6riefn9PcJmh6vNCvIsbMY6S+FhkWUTTi0XJjUDYRLlwM273FfWhzHK+SSQXeBp/z
InlgFvJhSZMRi9HZpDoqxLbBB9QIdIw6U4MIjNmKsSI/ASpkFm2GuQ7ZK9KuMItY2AoC
uIRmRADf8iYNHBTXNfFurGogXwRDjQ==
```

必要に応じて、ホストキーを生成します。OpenSSH ツールを使用の場合は、空白のパスフレーズを使用し、キーをユーザーの `~/.ssh/` ディレクトリー以外の場所に保存して、既存のキーを上書きしないようにします。

```
[jsmith@server ~]$ ssh-keygen -t rsa -C "server.example.com,1.2.3.4"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/jsmith/.ssh/id_rsa):
/home/jsmith/.ssh/host_keys
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/jsmith/.ssh/host_keys.
Your public key has been saved in /home/jsmith/.ssh/host_keys.pub.
The key fingerprint is:
SHA256:GAUIDVVEgly7rs1lTWP6oguHz8BKvyZkpqCqVSsmi7c
server.example.com
The key's randomart image is:
+--[ RSA 2048 ]-----+
|           .. |
|           .+ |
|          o  .* |
|         o .  .* |
```

```
|      S + . 0+|
|      E . . . |
|      . = . 0  |
|      0 .  .0  |
|      . . . . |
|      +-----+
```

- 2. 公開キーをキーファイルからコピーします。完全なキーエントリーは、**hostname,IP type key==** の形式で、必要なのは **key==** の部分だけですが、エントリー全体を保存することもできます。エントリー内の全要素を使用するには、エントリーを配列しなおして、**type key== [hostname,IP]** の順序にします。

```
[jsmith@server ~]$ cat /home/jsmith/.ssh/host_keys.pub
ssh-rsa AAAAB3NzaC1yc2E...tJG1PK2Mq++wQ== server.example.com,1.2.3.4
```

- 3. **Identity** タブを開き、**Hosts** サブタブを選択します。
- 4. 編集するホスト名をクリックします。

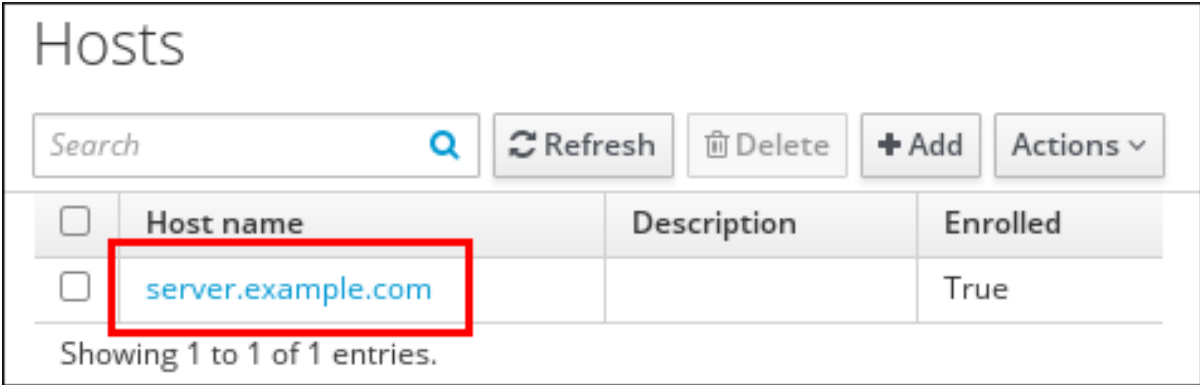
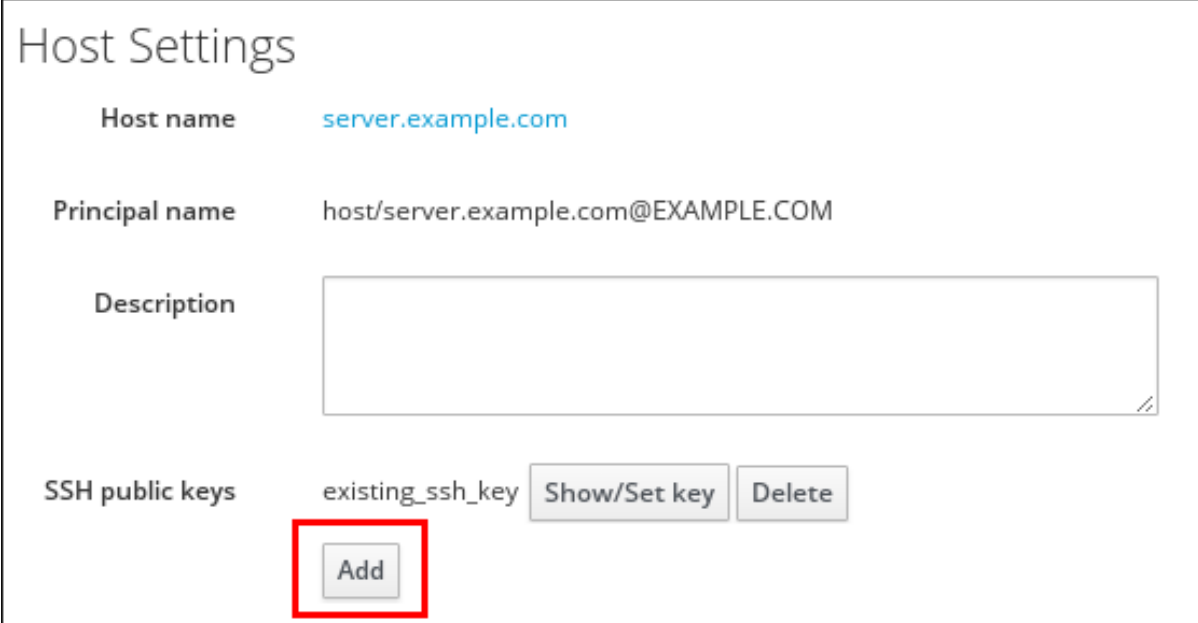


図12.4 ホストの一覧

- 5. **Settings** タブの **Host Settings** エリアで、**SSH public keys** の横にある **Add** をクリックします。



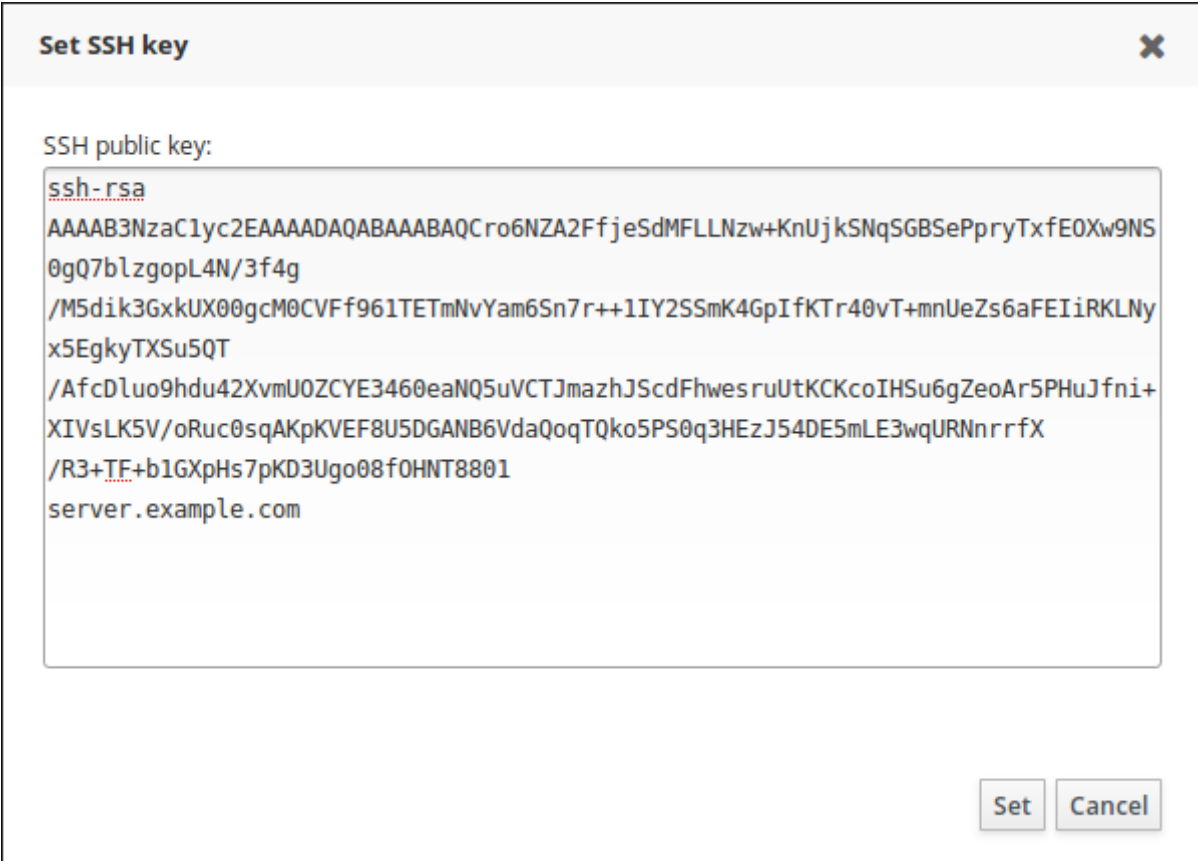
The 'Host Settings' dialog box displays configuration for a host named 'server.example.com'. It includes fields for 'Host name', 'Principal name' (host/server.example.com@EXAMPLE.COM), and a 'Description' text area. Under the 'SSH public keys' section, there is a table with one entry 'existing_ssh_key' and buttons for 'Show/Set key' and 'Delete'. A red rectangle highlights an 'Add' button located below the table.

SSH public keys
existing_ssh_key

Buttons: Show/Set key, Delete, Add (highlighted)

図12.5 SSH キーの追加

6. ホストの公開キーを貼り付けて、**Set** をクリックします。



The 'Set SSH key' dialog box shows a text area for pasting an SSH public key. The key is labeled 'ssh-rsa' and consists of a long alphanumeric string followed by the host name 'server.example.com'. At the bottom right, there are 'Set' and 'Cancel' buttons.

SSH public key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACro6NZA2FfjeSdMFLLNzw+KnUjkSNqSGBSePpryTxfe0Xw9NS
0gQ7blzgopL4N/3f4g
/M5dik3GxkUX00gcM0CVFf961TETmNvYam6Sn7r++1IY2SSmK4GpIfKTr40vT+mnUeZs6aFEIiRKLNy
x5EgkyTXSu5QT
/AfcDluo9hdu42XvmU0ZCYE3460eaNQ5uVCTJmazhJScdFhwesruUtKCKcoIHSu6gZeoAr5PHuJfni+
XIVsLK5V/oRuc0sqAKpKVEF8U5DGANB6VdaQoqTQko5PS0q3HEzJ54DE5mLE3wqURNnrrfX
/R3+TF+b1GXpHs7pKD3Ugo08f0HNT8801
server.example.com
```

Buttons: Set, Cancel

図12.6 SSH キーの設定

これで **SSH public keys** フィールドに、新しいキーが表示されます。**Show/Set key** をクリックすると、追加したキーが表示されます。

7. 複数のキーをアップロードするには、公開キーリストの下にある **Add** をクリックして、他のキーをアップロードします。

- すべてのキーが追加されたら、ホストページ上部の **Save** をクリックして、変更を保存します。

公開キーが保存されると、エントリーにはキーの指紋、コメント (ある場合)、キーのタイプが表示されます [2]。

ホストキーをアップロードしたら、Identity Management を ID ドメインの 1 つとして使用するよう SSSD を設定し、OpenSSH がホストキー管理に SSSD ツールを使用するよう設定します。これは、[the "Configuring Services: OpenSSH and Cached Keys" section in the System-Level Authentication Guide](#) で説明しています。

12.5.4. コマンドライン からホストキーを追加する

ホスト SSH キーが IdM のホストエントリーに追加されるのは、**host-add** を使ってホストを作成する際か、エントリーを後で修正する際になります。



注記

インストールスクリプトで SSH サービスが明示的に無効にされなければ、**ipa-client-install** コマンドで RSA と DSS ホストキーが作成されます。

- host-mod** コマンドを **--sshpubkey** オプションと実行して、base64 暗号化公開キーをユーザーエントリーにアップロードします。

ホストキーを追加するとホストの DNS SSHFP エントリーも変更されるので、**--updatedns** オプションも使ってホストの DNS エントリーも更新します。

例を示します。

```
[jsmith@server ~]$ ipa host-mod --sshpubkey="ssh-rsa RjlzYQo==" --updatedns host1.example.com
```

キーは通常はイコール記号 (=) で終わりますが、実際にはもっと長いものになります。

複数のキーをアップロードするには、複数の **--sshpubkey** コマンドラインパラメーターを入力します。

```
--sshpubkey="RjlzYQo==" --sshpubkey="ZEt0TAo=="
```



注記

ホストは複数の公開キーを持つことが可能です。

- ホストキーをアップロードしたら、Identity Management を ID ドメインの 1 つとして使用するよう SSSD を設定し、OpenSSH がホストキー管理に SSSD ツールを使用するよう設定します。これは、[the "Configuring Services: OpenSSH and Cached Keys" section in the System-Level Authentication Guide](#) で説明しています。

12.5.5. ホストキーの削除

ホストキーは、期限が切れるか有効でなくなると、削除することができます。

個別のホストキーを削除する一番簡単な方法は、Web UI によるものです。

1. **Identity** タブを開き、**Hosts** サブタブを選択します。
2. 編集するホスト名をクリックします。

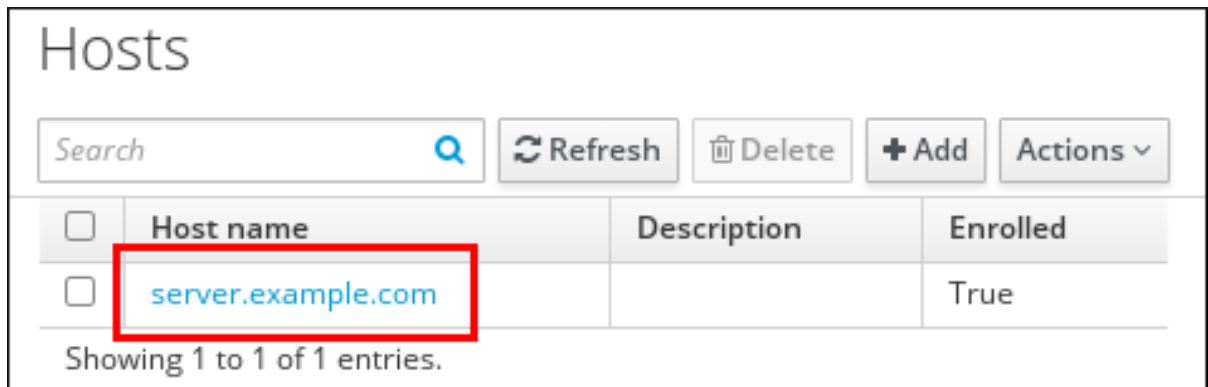


図12.7 ホストの一覧

3. **SSH public keys** エリアで、削除するキーの指紋の横にある **Delete** をクリックします。

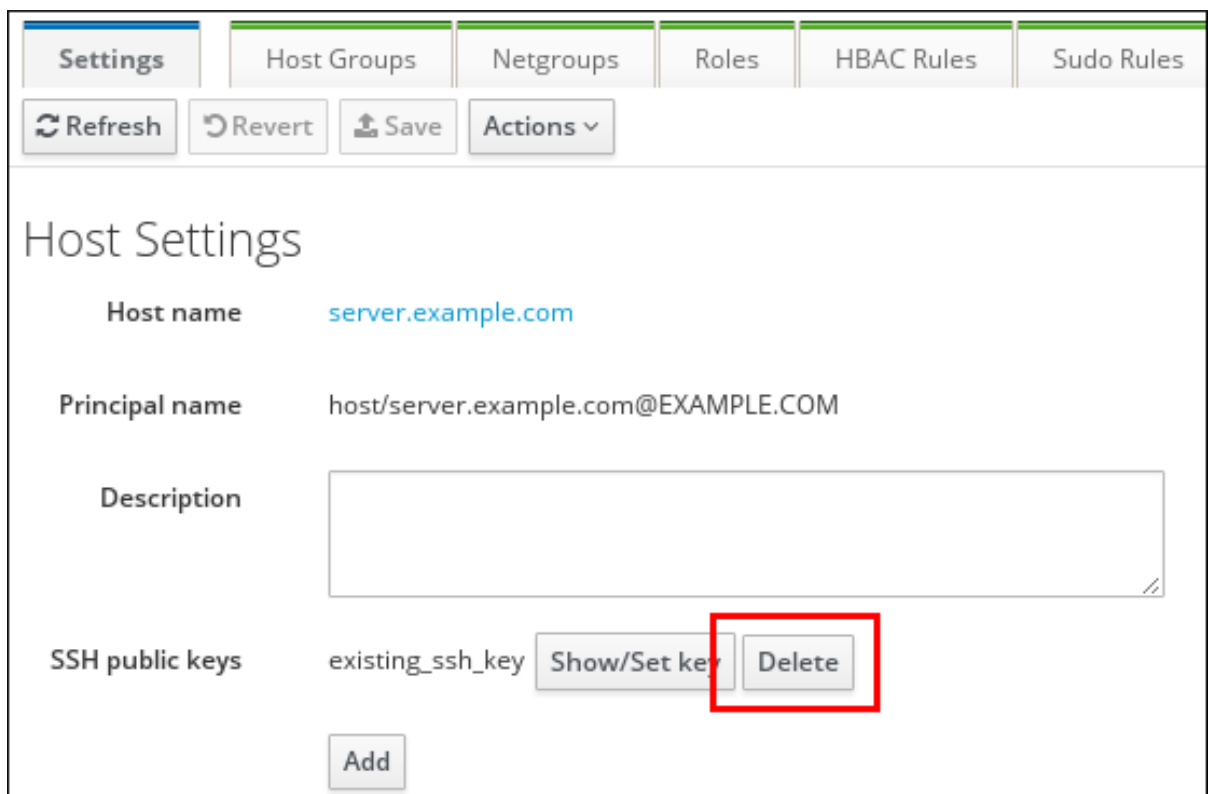


図12.8 公開キーの削除

4. ホストのページの上部にある **Save** をクリックして変更を保存します。

コマンドラインツールを使ってすべてのキーを削除することもできます。**ipa host-mod** を **--sshpubkey=** の値を空白にして実行します。これでホストのすべての公開キーが削除されます。また、**--updatedns** オプションを使うと、ホストの DNS エントリが更新されます。例を示します。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa host-mod --sshpubkey= --updatedns host1.example.com
```

12.6. ホストの ETHERS 情報の設定

NIS は **ethers** テーブルをホストすることができます。これを使うと、システムのプラットフォームやオペレーティングシステム、DNS ドメイン、および MAC アドレスに基づいて DHCP 設定ファイルを管理することができます。これらすべての情報は、IdM のホストエントリーに保存されます。

Identity Management では、各システムは、**ou=ethers** サブツリーのディレクトリーに含まれる適切な **ethers** エントリーで作成されます。

```
cn=server,ou=ethers,dc=example,dc=com
```

このエントリーは、**ethers** サービスの NIS マップを作成するために使用されます。ethers サービスは、IdM の NIS 互換性プラグインで管理できます。

ethers エントリーの NIS マップを設定するには、以下の手順に従います。

1. ホストエントリーに MAC アドレス属性を追加します。例を示します。

```
[jsmith@server ~]$ kinit admin
[jsmith@server ~]$ ipa host-mod --macaddress=12:34:56:78:9A:BC
server.example.com
```

2. **nsswitch.conf** ファイルを開きます。
3. **ethers** サービスの行を追加し、ルックアップに LDAP を使用するよう設定します。

```
ethers: ldap
```

4. **ethers** 情報がクライアントで利用可能かどうかを確認します。

```
[root@server ~]# getent ethers server.example.com
```

[2] キータイプは、アップロードされたキーに含まれていない場合、キー自体から自動的に判断されます。

第13章 ユーザーおよびホストグループの管理

13.1. ユーザーおよびホストグループの **IDM** での機能

13.1.1. ユーザーおよびホストグループとは

ユーザーグループは、一般的な権限、パスワードポリシー、その他の特長を持つユーザーセットです。

ホストグループは、一般的なアクセス制御ルールやその他の特長を持つ IdM ホストです。

たとえば、企業の部門、物理的な場所、アクセス制御要件をもとにグループを定義することができます。

13.1.2. サポートされるグループメンバー

IdM のユーザーグループには以下が含まれます。

- IdM ユーザー
- その他の IdM ユーザーグループ
- IdM 以外に所属する外部ユーザー

IdM のホストグループには、以下が含まれます。

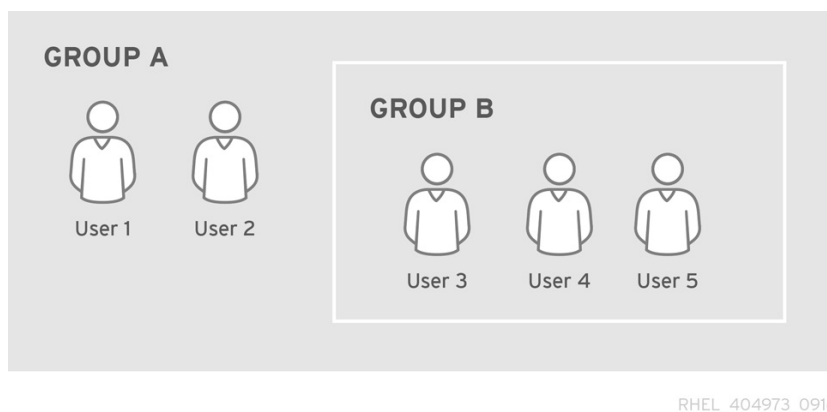
- IdM サーバーおよびクライアント
- その他の IdM ホストグループ

13.1.3. グループの直接および間接メンバー

IdM のユーザーおよびホストグループ属性は、直接および間接メンバー両方に適用されます。たとえば、グループ B がグループ A に所属している場合には、グループ B の全ユーザーは、グループ A のメンバーと見なされます。

たとえば、[図13.1「グループの直接および間接メンバー」](#)では:

- ユーザー 1 およびユーザー 2 は、グループ A の *直接*メンバーです。
- ユーザー 3、ユーザー 4 およびユーザー 5 は、グループ A の *間接*メンバーです。



RHEL_404973_0916

図13.1 グループの直接および間接メンバー

ユーザーグループ A にパスワードポリシーを設定した場合に、このポリシーは、ユーザーグループ B の全ユーザーにも適用されます。

例13.1 グループの直接および間接メンバーの表示

1. **group_A** と **group_B** の 2 つのグループを作成します。「[ユーザーまたはホストグループの追加および削除](#)」を参照してください。
2. 以下を追加します。
 - **group_A** のメンバーとしてユーザーを 1 つ追加します。
 - 別のユーザーを **group_B** のメンバーとして追加します。
 - **group_A** メンバーとして **group_B** を追加します。

「[ユーザーまたはホストグループメンバーの追加および削除](#)」を参照してください。
3. Web UI で、**Identity** → **Groups** を選択します。左側のサイドバーに表示されている各グループタイプから、**User Groups** を選択し、**group_A** の名前をクリックします。**Direct Membership** と **Indirect Membership** を切り替えます。

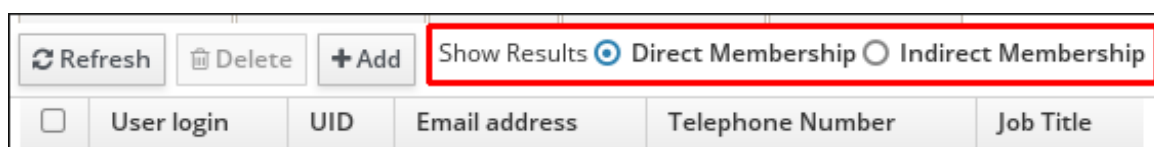


図13.2 グループの直接および間接メンバー

4. コマンドラインを使用する場合は **ipa group-show** コマンドを実行します。

```
$ ipa group-show group_A
...
Member users: user_1
Member groups: group_B
Indirect Member users: user_2
```

13.1.4. IdM のユーザーグループタイプ

POSIX グループ (デフォルト)

POSIX グループは、所属メンバーの POSIX 属性をサポートします。Active Directory と対話するグループは POSIX の属性を使用できない点に注意してください。

POSIX 以外のグループ

このタイプのグループメンバーはすべて、IdM ドメインに所属する必要があります。

外部グループ

外部グループにより、IdM ドメイン以外のアイデンティティストアに存在するグループメンバーを追加できます。外部ストアは、ローカルシステム、Active Directory ドメイン、ディレクトリサービスいずれかを指定できます。

POSIX 以外の外部グループは、POSIX 属性をサポートしません。たとえば、これらのグループには、GID は定義されません。

例13.2 異なるタイプのユーザーグループの検索

1. 全ユーザーグループを表示するには、**ipa group-find** コマンドを実行します。
2. また、全 POSIX グループを表示するには、**ipa group-find --posix** コマンドを実行します。
3. POSIX グループ以外のグループをすべて表示するには **ipa group-find --nonposix** コマンドを実行します。
4. 全外部グループを表示するには **ipa group-find --external** コマンドを実行します。

13.1.5. デフォルトで作成されるユーザーおよびホストグループ

表13.1 デフォルトで作成されるユーザーおよびホストグループ

グループ名	ユーザーまたはホスト	デフォルトのグループメンバー
ipausers	ユーザーグループ	全 IdM ユーザー
admins	ユーザーグループ	管理者権限のあるユーザー。デフォルトでは最初は admin ユーザーです。
editors	ユーザーグループ	管理者ユーザーの権限なしに Web UI で他の IdM ユーザーを編集できるユーザー
trust admins	ユーザーグループ	Active Directory トラストを管理する権限のあるユーザー
ipaservers	ホストグループ	全 IdM サーバーホスト

ユーザーグループにユーザーを追加すると、グループに関連付けられた特権およびポリシーが適用されます。たとえば、**admins** グループにユーザーを追加するには、ユーザーに管理者権限が割り当てられます。



警告

ipaservers ホストグループにホストを追加するときは注意してください。**ipaservers** のホストにはすべて、IdM サーバーにプロモートする機能が割り当てられています。

さらに IdM では、新規ユーザーが IdM に作成されると、デフォルトで ユーザープライベートグループが作成されます。

- ユーザープライベートグループは、作成したユーザーと同じ名前が指定されます。
- このユーザーは、ユーザープライベートグループにのみ所属します。
- プライベートグループの GID は、ユーザーの UID と同じです。

例13.3 ユーザープライベートグループの表示

全ユーザープライベートグループを表示するには、**ipa group-find --private** コマンドを実行します。

```
$ ipa group-find --private
-----
2 groups matched
-----
Group name: user1
Description: User private group for user1
GID: 830400006

Group name: user2
Description: User private group for user2
GID: 830400004
-----
Number of entries returned 2
-----
```

NIS グループや別のシステムグループがユーザーのプライベートグループに割り当てられた GID をすでに使用している場合など、ユーザーのプライベートグループを作成しないほうが良い場合があります。「[ユーザープライベートグループの無効化](#)」を参照してください。

13.2. ユーザーまたはホストグループの追加および削除

グループを追加するには、以下を使用します。

- Web UI (「[Web UI: ユーザーまたはホストグループの追加](#)」を参照してください)

- コマンドライン (「[コマンドライン: ユーザーまたはホストグループの追加](#)」を参照してください)

IdM は、ユーザーグループの作成時にカスタムの GID を指定することができます。これを行う場合には、ID の競合が発生しないように注意してください。「[ID の値が一意であることを確認する](#)」を参照してください。カスタム GID を指定しない場合には、IdM により利用可能な ID 範囲から GID を自動的に割り当てられます。

グループを削除するには以下を使用します。

- Web UI (「[Web UI: ユーザーまたはホストグループの削除](#)」を参照してください)
- コマンドライン (「[コマンドライン: ユーザーまたはホストグループの削除](#)」を参照してください)

グループを削除しても、IdM のグループメンバーは削除されない点に注意してください。

Web UI: ユーザーまたはホストグループの追加

1. **Identity** → **Groups** をクリックして、左のサイドバーで **User Groups** または **Host Groups** を選択してください。
2. **Add** をクリックして、グループの追加を開始します。
3. グループの情報を入力します。

ユーザーグループタイプの詳細は、「[IdM のユーザーグループタイプ](#)」を参照してください。

4. **Add** をクリックして確定します。

コマンドライン: ユーザーまたはホストグループの追加

1. 管理者としてログインします。

```
$ kinit admin
```

2. ユーザーグループを追加するには **ipa group-add** コマンドを実行し、ホストグループを追加するには **ipa hostgroup-add** コマンドを使用します。

```
$ ipa group-add group_name
-----
Added group "group_name"
-----
```

デフォルトでは **ipa group-add** は POSIX ユーザーグループを追加します。異なるグループタイプを指定するには、**ipa group-add** にオプションを追加します。

- POSIX グループ以外を追加するには **--nonposix**
- 外部グループを作成するには **--external**

グループタイプの詳細は、「[IdM のユーザーグループタイプ](#)」を参照してください。

Web UI: ユーザーまたはホストグループの削除

1. **Identity** → **Groups** をクリックして、左のサイドバーで **User Groups** または **Host Groups** を選択してください。

2. 削除するグループを選択して **Delete** をクリックします。

コマンドライン: ユーザーまたはホストグループの削除

1. 管理者としてログインします。

```
$ kinit admin
```

2. ユーザーグループを削除するには **ipa group-del group_name** コマンドを、ホストグループを削除するには **ipa hostgroup-del group_name** コマンドを実行します。

```
$ ipa group-del group_name
-----
Deleted group "group_name"
-----
```

13.3. ユーザーまたはホストグループメンバーの追加および削除

ユーザーグループにメンバーを追加するには、以下を使用します。

- IdM Web UI (「[Web UI: ユーザーまたはホストグループへのメンバーの追加](#)」を参照してください)
- コマンドライン (「[コマンドライン: メンバーのユーザーグループへの追加](#)」を参照してください)



重要

ユーザーグループをメンバーとして追加する際に、再帰グループは作成しないでください。たとえば、グループ A がグループ B のメンバーの場合は、グループ B はグループ A に追加しないでください。再帰グループを作成すると、予期せぬ動作を引き起こす可能性があります。

ユーザーグループからメンバーを削除する場合に、以下を使用することができます。

- IdM Web UI (「[Web UI: ユーザーまたはホストグループからのメンバーの削除](#)」を参照してください)
- コマンドライン (「[コマンドライン: ユーザーグループからのメンバーの削除](#)」を参照してください)

Web UI: ユーザーまたはホストグループへのメンバーの追加

1. **Identity** → **Groups** をクリックして、左のサイドバーで **User Groups** または **Host Groups** を選択してください。
2. グループ名をクリックします。
3. 追加するグループメンバーのタイプを選択します。ユーザーグループの **Users**、**User Groups** または **External** などです。

The screenshot shows a web interface for managing a user group named 'group'. At the top, it says 'User Group: group'. Below this, there's a section titled 'group members:' which contains three tabs: 'Users', 'User Groups', and 'External'. The 'Users' tab is currently selected. To the right of this section, there's another section titled 'group is a member of:' with tabs for 'User Groups', 'Netgroups', and 'Roles'. Below the 'group members:' section, there are three buttons: 'Refresh', 'Delete', and '+ Add'. The '+ Add' button is highlighted with a red box. Below the buttons, there's a table with columns for 'User login', 'UID', and 'Email address'. The table currently shows 'No entries.'

図13.3 ユーザーグループメンバーの追加

4. **Add** をクリックします。
5. 追加するメンバーを選択し、**Add** をクリックして確定します。

コマンドライン: メンバーのユーザーグループへの追加

1. オプション: グループの検索には、**ipa group-find** または **ipa hostgroup-find** コマンドを使用します。
2. ユーザーグループにメンバーを追加するには **ipa group-add-member** コマンドを実行し、ホストグループにメンバーを追加するには **ipa hostgroup-add-member** コマンドを使用します。

ユーザーグループメンバーを追加する場合は、以下のオプションを使用してメンバーを指定します。

- **--users** を指定して、IdM ユーザーを追加します。
- **DOMAIN\user_name** または **user_name@domain** の形式で、**--external** を指定して IdM ドメイン外に存在するユーザーを追加します。
- **--groups** を指定して IdM ユーザーグループを追加します。

ホストグループメンバーを追加する場合は、以下のオプションを使用してメンバーを指定します。

- **--hosts** を指定して IdM ホストを追加します。
- **--groups** を指定して IdM ホストグループを追加します。

たとえば、*group_name* と呼ばれるグループに、*user1*、*user2* および *group1* を追加するには、以下を実行します。

```
$ ipa group-add-member group_name --users=user1 --users=user2 --groups=group1
```

Web UI: ユーザーまたはホストグループからのメンバーの削除

1. **Identity** → **Groups** をクリックして、左のサイドバーで **User Groups** または **Host Groups** を選択してください。
2. グループ名をクリックします。
3. 削除するグループメンバーのタイプを選択します。ユーザーグループの **Users**、**User**

Groups または **External** などです。

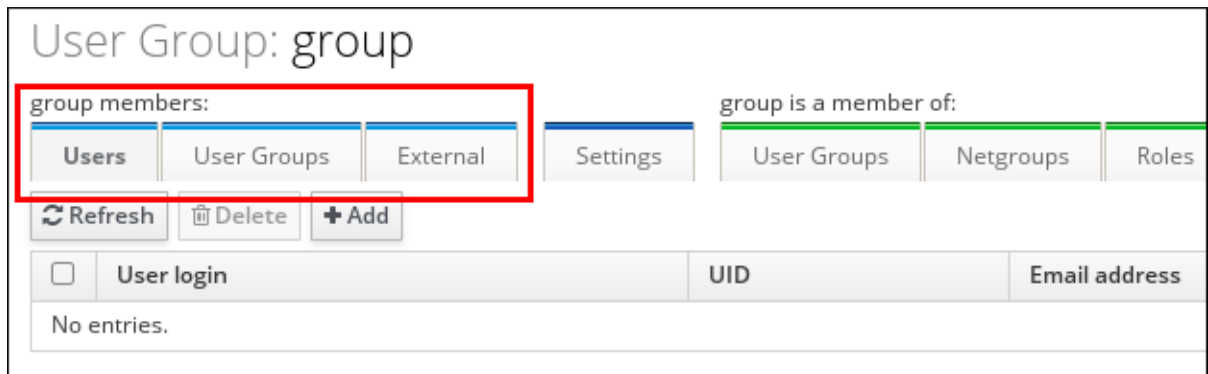


図13.4 ユーザーグループメンバーの削除

4. 所定のメンバーの横にあるチェックボックスを選択します。
5. **Delete** をクリックします。

コマンドライン: ユーザーグループからのメンバーの削除

1. オプション: **ipa group-show** または **ipa hostgroup-show** コマンドを使用して、削除するグループが含まれるグループを確定します。
2. ユーザーグループメンバーを削除するには、**ipa group-remove-member** コマンドをします。ホストグループメンバーを削除するには **ipa hostgroup-remove-member** コマンドを使用します。

ユーザーグループメンバーを削除する場合は、以下のオプションを使用してメンバーを指定します。

- **--users** を指定して IdM ユーザーを削除します。
- **DOMAIN\user_name** または **user_name@domain** の形式で、**--external** を指定して IdM ドメイン外に存在するユーザーを削除します。
- **--groups** を指定して IdM ユーザーグループを削除します。

ホストグループメンバーを削除する場合は、以下のオプションを使用してメンバーを指定します。

- **--hosts** を指定して IdM ホストを削除します。
- **--groups** を指定して IdM ホストグループを削除します。

たとえば、*group_name* と呼ばれるグループから、*user1*、*user2* および *group1* を削除するには、以下を実行します。

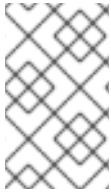
```
$ ipa group-remove-member group_name --users=user1 --users=user2 --groups=group1
```

13.4. ユーザープライベートグループの無効化

IdM が新規ユーザーの作成時にデフォルトのユーザープライベートグループを作成しないようにするには、以下のいずれかを選択します。

- 「ユーザープライベートグループなしでのユーザー作成」
- 「全ユーザーに対してユーザープライベートグループを無効化する方法」

デフォルトのユーザープライベートグループの作成を無効化した後でも、IdM では、新規ユーザーの追加の際には GID が必要です。ユーザーが正常に追加されるようにするには「[ユーザープライベートグループを無効にしたユーザーの追加](#)」を参照してください。



注記

GID が競合してしまうために、デフォルトのユーザープライベートグループの作成を無効にする場合にはデフォルトの UID と GID の割当範囲を変更することを検討してください。[14章 一意の UID および GID 番号の割り当て](#)を参照してください。

13.4.1. ユーザープライベートグループなしでのユーザー作成

--noprivate オプションを **ipa user-add** コマンドに追加します。正常にコマンドが実行されるように、カスタムのプライベートグループを指定する必要があります。「[ユーザープライベートグループを無効にしたユーザーの追加](#)」を参照してください。

13.4.2. 全ユーザーに対してユーザープライベートグループを無効化する方法

1. 管理者としてログインします。

```
$ kinit admin
```

2. IdM は、Directory Server の Managed Entries Plug-in を使用してユーザープライベートグループを管理します。どのようなプラグインがあるかを表示するには以下を実行します。

```
$ ipa-managed-entries --list
```

3. IdM によりユーザープライベートグループが作成されないようにするには、ユーザープライベートグループを管理するプラグインインスタンスを無効にします。

```
$ ipa-managed-entries -e "UPG Definition" disable
Disabling Plugin
```



注記

UPG Definition インスタンスを再度有効にするには、**ipa-managed-entries -e "UPG Definition" enable** コマンドを使用します。

4. Directory Server を再起動して、新しい設定を読み込みます。

```
# systemctl restart dirsrv.target
```

13.4.3. ユーザープライベートグループを無効にしたユーザーの追加

デフォルトのユーザープライベートグループを無効にした状態で新しいユーザーを正常に追加するには、以下のいずれかを選択します。

- 新規ユーザーの追加時にはカスタムの GID を指定します。GID は、既存のユーザーグループと一致させる必要はありません。

たとえば、コマンドラインからユーザーを追加する場合は、**ipa user-add** コマンドに **--gid** オプションを追加します。

- GID がある既存のグループにユーザーを追加するには、**automember** ルールを使用します。「[ユーザーおよびホストの自動グループメンバーシップの定義](#)」を参照してください。

13.5. ユーザーおよびユーザーグループの検索属性の設定

ipa user-find keyword および **ipa group-find keyword** コマンドを使用して特定のキーワードのエントリを検索するには、以下のように IdM は特定の属性のみを検索します。

- ユーザーの検索: 姓、名、ユーザー名 (ログイン ID)、役職、組織単位 (UO)、電話番号、UID、メールアドレス
- グループの検索: グループ名、説明

以下の手順では、IdM が他の属性も検索するように設定する方法を説明します。IdM は常にデフォルトの属性を検索する点に注意してください。たとえば、ユーザー検索属性から役職の属性を削除した場合でも、IdM はユーザーの役職を検索します。

前提条件

新しい属性を追加する前に、この属性に対して適切なインデックスが LDAP ディレクトリーに存在することを確認してください。標準の LDAP 属性の多くには、LDAP にインデックスがあります。カスタムの属性を追加する場合には、手動でインデックスを作成する必要があります。『[Directory Server Administration Guide](#)』の [Creating Standard Indexes](#) を参照してください。

Web UI: 検索属性の設定

1. **IPA Server** → **Configuration** を選択します。
2. **User Options** エリアで、**User search fields** のユーザー検索属性を設定します。
3. **Group Options** エリアで、**Group search fields** のグループ検索属性を選択します。
4. ページ上部にある **Save** をクリックします。

コマンドライン: 検索属性の設定

これらのオプションを指定して、**ipa config-mod** コマンドを実行します。

- **--usersearch** は、ユーザーの検索属性の新規リストを定義します。
- **--groupsearch** は、グループの検索属性の新規リストを定義します。

以下に例を示します。

```
$ ipa config-mod --usersearch={uid,givenname,sn,telephonenumber,ou,title}
$ ipa config-mod --groupsearch={cn,description}
```

13.6. ユーザーおよびホストの自動グループメンバーシップの定義

13.6.1. IdM での自動グループメンバーシップの機能方法

13.6.1.1. 自動グループメンバーシップとは

自動グループメンバーシップを使用して、属性をもとに自動的にユーザーとホストをグループに割り当てることができます。たとえば、以下が可能です。

- 従業員のマネージャー、場所またはその他の属性をもとに従業員のユーザーエントリーを複数のグループに分割することができます。
- クラス、場所、またはその他の属性をもとにホストを分割できます。
- 全ユーザーまたは全ホストを単一のグローバルグループに追加できます。

13.6.1.2. 自動グループメンバーシップの利点

グループメンバーシップの手動管理によるオーバーヘッドの削減

自動グループメンバーシップでは、管理者はユーザーとホストをグループに手動で割り当てる必要がありません。

ユーザーおよびホスト管理での一貫性向上

自動グループメンバーシップでは、厳密に定義され、自動評価された基準をもとに、ユーザーおよびホストが割り当てられます。

グループベースの設定管理の容易化

様々な設定がグループに定義されており、**sudo** ルール、**automount**、またはアクセス制御など、個別のグループメンバーに適用されます。自動グループメンバーを使用する場合には、ユーザーおよびホストは自動的に特定のグループに追加され、グループベースの設定の管理が簡素化されます。

13.6.1.3. Automember ルール

自動グループメンバーシップを設定する際には、管理者は *automember* ルールを定義します。*automember* ルールは、特定ユーザーまたはホストグループに適用されます。このルールには、ユーザーまたはホストが満たす必要のある *条件* が含まれており、それをもとにグループに追加、グループから除外されます。

包含条件

ユーザーまたはホストのエントリーが包含ルールを満たす場合には、グループに含まれます。

除外の条件

ユーザーまたはホストのエントリーが除外の条件を満たす場合にはグループに追加 **されません**。

これらの条件は、Perl-compatible regular expressions (PCRE) 形式の正規表現で指定します。PCRE に関する詳しい情報は、*pcresyntax(3)* の *man* ページを参照してください。

IdM は除外の条件を先に評価してから、包含条件を評価します。競合がある場合には、除外の条件が包含の条件より優先されます。

13.6.2. automember ルールの追加

以下を使用して、*automember* ルールを追加してください。

- IdM web UI を使用する場合は「[Web UI: Automember ルールの追加](#)」を参照してください

い。

- コマンドラインを使用する場合は「[コマンドライン: Automember ルールの追加](#)」を参照してください。

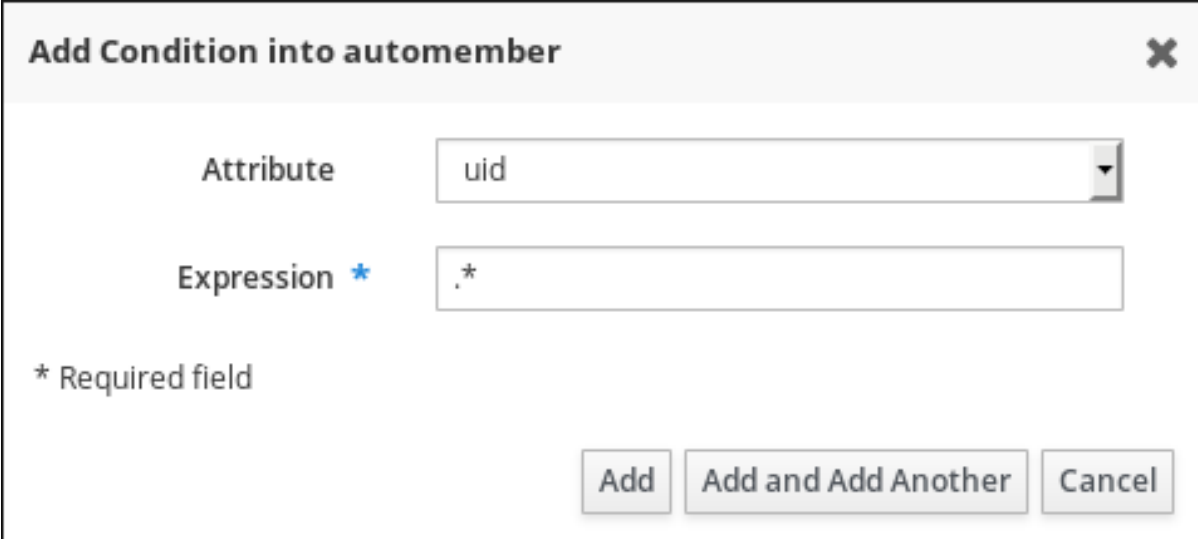
automember ルールを追加した後に、以下を実行します。

- 今後作成されるエントリーはすべて、指定したグループに所属します。エントリーが複数の automember ルールで指定した条件を満たす場合には、該当するすべてのグループに追加されます。
- 既存のエントリーは、指定のグループのメンバーには追加されません。詳しい情報は、「[automember ルールの既存のユーザーおよびホストへの適用](#)」を参照してください。

Web UI: Automember ルールの追加

1. **Identity → Automember → User group rules** または **Host group rules** を選択します。
2. **Add** をクリックします。
3. **Automember rule** フィールドで、ルールを適用するグループを選択します。**Add and Edit** をクリックします。
4. 包含および除外条件を 1 つ以上定義します。詳細は「[Automember ルール](#)」を参照してください。
 - a. **Inclusive** または **Exclusive** セクションで **Add** をクリックします。
 - b. **Attribute** フィールドで、必要な属性を選択します。
 - c. **Expression** フィールドで、正規表現を定義します。
 - d. **Add** をクリックします。

たとえば、以下の条件は、ユーザーログインの属性 (**uid**) のすべての値 (**.***) を対象としています。



Add Condition into automember ✕

Attribute

Expression *

* Required field

Add Add and Add Another Cancel

図13.5 Automember ルール条件の追加

コマンドライン: Automember ルールの追加

1. **ipa automember-add** コマンドを使用して、automember ルールを追加します。プロンプトが表示されたら、以下を指定します。

- 。対象のグループ名と一致する **Automember rule**
- 。ルールの対象がユーザーグループか、ホストグループかを指定する **Grouping Type**。ユーザーグループを対象とするには **group**、ホストグループを対象とする場合は **hostgroup** を入力してください。

たとえば **user_group** という名前のユーザーグループの automember ルールを追加するには以下を実行します。

```
$ ipa automember-add
Automember Rule: user_group
Grouping Type: group
-----
Added automember rule "user_group"
-----
Automember Rule: user_group
```

2. 包含および除外条件を 1 つ以上定義します。詳細は「[Automember ルール](#)」を参照してください。

- a. 条件を追加するには **ipa automember-add-condition** コマンドを使用します。プロンプトが表示されたら、以下を指定します。

- 対象のグループ名と一致する **Automember rule**
- フィルターを適用するエントリー属性を指定する **Attribute Key**。たとえば、ユーザーの **manager** などです。
- ルールの対象がユーザーグループか、ホストグループかを指定する **Grouping Type**。ユーザーグループを対象とするには **group**、ホストグループを対象とする場合は **hostgroup** を入力してください。
- 正規表現として 1 つまたは複数の条件を指定する **Inclusive regex** および **Exclusive regex**。条件を 1 つだけ指定する場合は、他の条件を指定するようにプロンプトが表示されたら、**Enter** を押してください。

たとえば、以下の条件は、ユーザーログインの属性 (**uid**) のすべての値 (.*) を対象としています。

```
$ ipa automember-add-condition
Automember Rule: user_group
Attribute Key: uid
Grouping Type: group
[Inclusive Regex]: .*
[Exclusive Regex]:
-----
Added condition(s) to "user_group"
-----
Automember Rule: user_group
Inclusive Regex: uid=.*
-----
Number of conditions added 1
-----
```

- b. 条件を削除するには **ipa automember-remove-condition** コマンドを使用します。

例13.4 コマンドライン: 単一のグループに全エントリーを追加する Automember ルールの作成

cn または **fqdn** など、全ユーザーまたはホストエントリーが含む属性の包含条件を作成すると、今後作成されるユーザーまたはホストのすべてが単一のグループに追加されるようになります。

1. **all_hosts** 当名前のホストグループなど、グループを作成します。「[ユーザーまたはホストグループの追加および削除](#)」を参照してください。
2. 以下のように、新規ホストグループの automember ルールを追加します。

```
$ ipa automember-add
Automember Rule: all_hosts
Grouping Type: hostgroup
-----
Added automember rule "all_hosts"
-----
Automember Rule: all_hosts
```

3. 全ホストを対象とする包含条件を追加します。以下の例では、包含条件は **fqdn** 属性に含まれるホストすべて (.*) を対象としています。

```
$ ipa automember-add-condition
Automember Rule: all_hosts
Attribute Key: fqdn
Grouping Type: hostgroup
[Inclusive Regex]: .*
[Exclusive Regex]:
-----
Added condition(s) to "all_hosts"
-----
Automember Rule: all_hosts
Inclusive Regex: fqdn=.*
-----
Number of conditions added 1
-----
```

今後追加されるホストはすべて自動的に **all_hosts** グループに所属します。

例13.5 コマンドライン: 同期済みの AD ユーザー用の Automember ルールの作成

Active Directory (AD) から同期された Windows ユーザーは **ntUser** オブジェクトクラスを共有します。**objectclass** 属性に **ntUser** が含まれる全ユーザーを対象とする automember の条件を作成すると、今後作成される同期済みの AD ユーザーは AD ユーザーの共通グループに追加されるようになります。

1. **ad_users** などの AD ユーザーのユーザーグループを作成します。「[ユーザーまたはホストグループの追加および削除](#)」を参照してください。
2. 以下のように、新規ユーザーグループの automember ルールを追加します。

```
$ ipa automember-add
Automember Rule: ad_users
```

```
Grouping Type: group
```

```
-----
Added automember rule "ad_users"
-----
```

```
Automember Rule: ad_users
```

3. AD ユーザーを絞り込む包含条件を追加します。以下の例では、**objectclass** 属性に **ntUser** の値が含まれる全ユーザーを対象とします。

```
$ ipa automember-add-condition
Automember Rule: ad_users
Attribute Key: objectclass
Grouping Type: group
[Inclusive Regex]: ntUser
[Exclusive Regex]:
-----
Added condition(s) to "ad_users"
-----
Automember Rule: ad_users
Inclusive Regex: objectclass=ntUser
-----
Number of conditions added 1
-----
```

今後追加される AD ユーザーはすべて自動的に **ad_users** ユーザーグループに所属します。

13.6.3. automember ルールの既存のユーザーおよびホストへの適用

Automember ルールは、ルールの追加後に作成されたユーザーおよびホストエントリーに自動的に適用されます。ルールの適用前に存在するエントリーについては遡って適用されません。

ルールの追加前に存在するエントリーに automember のルールを適用するには、自動メンバーシップを手動で再構築してください。自動メンバーシップを再構築すると、既存の automember ルールがすべて再評価され、全エントリーまたは固有のエントリーにルールが適用されます。

Web UI: 既存のエントリーの自動メンバーシップの再構築

全ユーザーまたはホストの自動メンバーシップを再構築するには以下を実行します。

1. **Identity** → **Users** または **Hosts** を選択します。
2. **Actions** → **Rebuild auto membership** をクリックします。

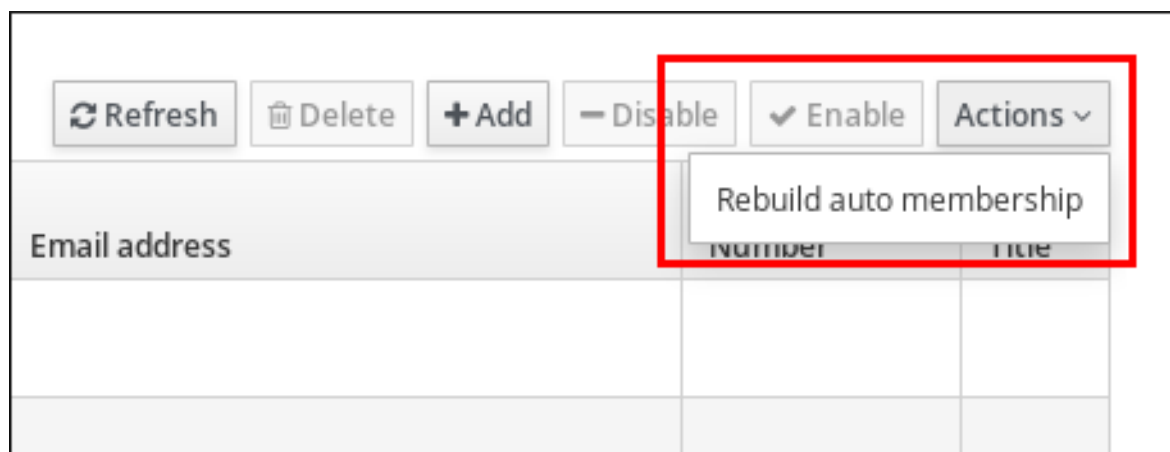


図13.6 全ユーザーまたはホストの自動メンバーシップの再構築

単一ユーザーまたはホストだけに自動メンバーシップを再構築するには、以下を実行します。

1. **Identity** → **Users** または **Hosts** を選択して、必要なユーザーログインまたはホスト名をクリックします。
2. **Actions** → **Rebuild auto membership** をクリックします。

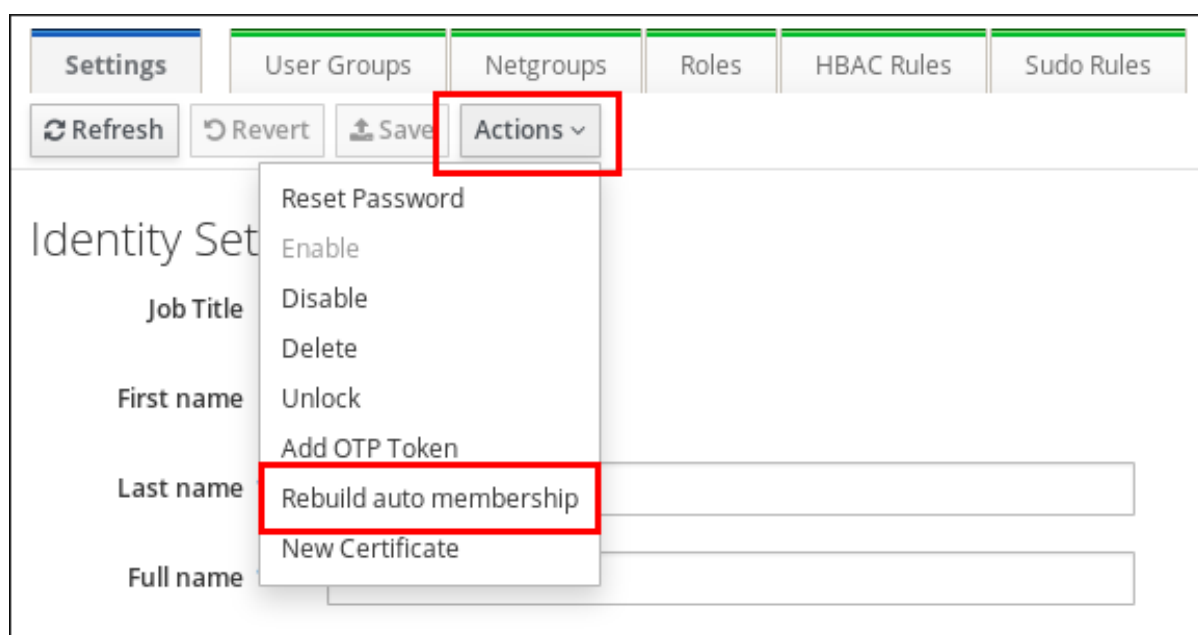


図13.7 単一のユーザーまたはホストの自動メンバーシップの再構築

コマンドライン: 既存のエントリーの自動メンバーシップの再構築

全ユーザーの自動メンバーシップを再構築するには **ipa automember-rebuild --type=group** コマンドを使用します。

```
$ ipa automember-rebuild --type=group
-----
Automember rebuild task finished. Processed (9) entries.
-----
```

全ユーザーの自動メンバーシップを再構築するには、**ipa automember-rebuild --type=hostgroup** コマンドを使用します。

指定のユーザーの自動メンバーシップを再構築するには、**ipa automember-rebuild --users=user** コマンドを使用します。

```
$ ipa automember-rebuild --users=user1 --users=user2
-----
Automember rebuild task finished. Processed (2) entries.
-----
```

指定のホストの自動メンバーシップを再構築するには **ipa automember-rebuild --hosts=example.com** コマンドを使用します。

13.6.4. デフォルトの Automember グループ設定

デフォルトの automember グループを設定すると、automember ルールに一致しないユーザーまたはホストエントリは自動的にデフォルトのグループに追加されます。

1. **ipa automember-default-group-set** コマンドを使用して、デフォルトの automember グループを設定します。プロンプトが表示されたら、以下を指定します。

- 。対象となるグループ名を指定する **Default (fallback) Group**
- 。対象がユーザーグループか、ホストグループかを指定する **Grouping Type**。ユーザーグループを対象とするには **group**、ホストグループを対象とする場合は **hostgroup** を入力してください。

以下に例を示します。

```
$ ipa automember-default-group-set
Default (fallback) Group: default_user_group
Grouping Type: group
-----
Set default (fallback) group for automember "default_user_group"
-----
Default (fallback) Group:
cn=default_user_group,cn=groups,cn=accounts,dc=example,dc=com
```

2. グループが正しく設定されていることを確認するには、**ipa automember-default-group-show** コマンドを使用します。このコマンドでは、現在のデフォルト automember グループが表示されます。以下に例を示します。

```
$ ipa automember-default-group-show
Grouping Type: group
Default (fallback) Group:
cn=default_user_group,cn=groups,cn=accounts,dc=example,dc=com
```

現在のデフォルトの automember グループを削除するには、**ipa automember-default-group-remove** コマンドを使用します。

第14章 一意の UID および GID 番号の割り当て

IdM サーバーはユーザー ID (UID) とグループ ID (GID) の値を生成し、同時にレプリカが同じ ID を生成しないようにします。1 つの組織に異なる複数のドメインがある場合は、一意の UID および GID の必要性は IdM ドメイン全体に適用されます。

14.1. ID の範囲

UID および GID 番号は *ID の範囲* に分けられます。個別のサーバーとレプリカでそれぞれの数的範囲を維持することで、サーバーまたはレプリカがあるエントリーに発行した ID の値が他のサーバーやレプリカが発行したものと重複する可能性が最小限に抑えられます。

Distributed Numeric Assignment (DNA) プラグインはドメイン用バックエンドの 389 Directory Server インスタンスの一部で、これにより範囲がサーバーおよびレプリカ間で更新、共有されます。このプラグインは、全マスターとレプリカの ID 範囲を管理します。各サーバーまたはレプリカには、現行 ID 範囲と新たな **次の** ID 範囲があります。前者の番号を使い果たすと、後者の番号が使用されます。DNA Directory Server プラグインの詳細については、[Red Hat Directory Server Deployment Guide](#) を参照してください。

14.2. インストール中の ID 範囲の割り当て

サーバーのインストール中に、**ipa-server-install** コマンドはデフォルトでランダムな現行 ID 範囲をインストール先に自動的に割り当てます。設定スクリプトがランダムで、合計 1 万 の範囲から 20 万 ID のある範囲を選択します。このようにランダムな範囲を選択することで、もし 2 つの別個の IdM ドメインを将来マージした場合でも、ID が競合する可能性を大幅に抑えられます。

ただし、サーバーインストール中に **ipa-server-install** コマンドで以下の 2 つのオプションを使用すると、手動で現行 ID 範囲を定義することができます。

- **--idstart** は、UID および GID 番号の最初の値を提供します。デフォルトでは、この値はランダムで選択されます。
- **--idmax** は、UID および GID 番号の最大値を提供します。デフォルトでは、この値は **--idstart** の最初の値に 199,999 を加えたものになります。

単一の IdM サーバーがインストールされている場合、新規ユーザーもしくはグループのエントリーには範囲全体からランダムな ID が割り当てられます。新規レプリカをインストールしてこのレプリカが独自の ID 範囲をリクエストすると、サーバーの元の ID 範囲が分割されて、サーバーとレプリカに分配されます。レプリカには、元のマスターで利用可能な残りの ID 範囲の半分が与えられます。その後は、サーバーとレプリカはそれぞれ、新規エントリーに対して各 ID 範囲を使用します。また、レプリカに割り当てられた ID 範囲の残りの ID が 100 を切ると (つまり、割り当て ID 範囲がなくなりそうになると)、レプリカは他の利用可能なサーバーに連絡して新たな ID 範囲をリクエストします。

サーバーが ID 範囲を受け取るのは、DNA プラグインが最初に使われる時です。それまでは、サーバーには定義された ID 範囲がありません。たとえば、マスターサーバーからレプリカを作成する際には、このレプリカは即座に ID 範囲を受け取るわけではありません。レプリカが元のマスターからの ID 範囲をリクエストするのは、最初の ID がレプリカに割り当てられる直前になります。



注記

レプリカが元のマスターから ID 範囲をリクエストする前にこのマスターが機能停止した場合は、レプリカはマスターに連絡できなくなります。このため、新規ユーザーをレプリカに追加使用とすると、これは失敗します。このような場合には、機能しなくなったマスターに割り当てられた ID 範囲を確認し、そこから手動で ID 範囲をレプリカに割り当てることができます。この方法については「[手動での ID 範囲の拡張および新規 ID 範囲の割り当て](#)」で説明しています。

14.3. 現在割り当てられている ID 範囲の表示

サーバーに設定されている ID 範囲を表示するには、以下のコマンドを使用します。

- **ipa-replica-manage dnarange-show** は、全サーバーに設定されている現行 ID 範囲、またはサーバーを指定した場合はそのサーバーの現行 ID 範囲を表示します。

```
# ipa-replica-manage dnarange-show
masterA.example.com: 1001-1500
masterB.example.com: 1501-2000
masterC.example.com: No range set

# ipa-replica-manage dnarange-show masterA.example.com
masterA.example.com: 1001-1500
```

- **ipa-replica-manage dnanextrange-show** は、全サーバーに設定されている次の ID 範囲、またはサーバーを指定した場合はそのサーバーの現行 ID 範囲を表示します。

```
# ipa-replica-manage dnanextrange-show
masterA.example.com: 1001-1500
masterB.example.com: No on-deck range set
masterC.example.com: No on-deck range set

# ipa-replica-manage dnanextrange-show masterA.example.com
masterA.example.com: 1001-1500
```

これらのコマンドについての詳細は、ipa-replica-manage(1) man ページを参照してください。

14.4. レプリカ削除後の ID 範囲の自動拡張

機能していたレプリカを削除する際には、**ipa-replica-manage del** コマンドを使用すると、そのレプリカに割り当てられていた ID 範囲を取得して、他の利用可能な IdM レプリカに次の範囲として追加することができます。こうすることで、他のレプリカに ID 範囲が継続して利用可能になります。

レプリカを削除した後に **ipa-replica-manage dnarange-show** と **ipa-replica-manage dnanextrange-show** コマンドを使用すると、他のサーバーに設定されている ID 範囲を確認することができます。これらのコマンドについては、「[現在割り当てられている ID 範囲の表示](#)」で説明しています。

14.5. 手動での ID 範囲の拡張および新規 ID 範囲の割り当て

場合によっては、手動で ID 範囲を調整する必要があることもあります。

割り当て ID 範囲を使い果たした場合

レプリカが割り当てられた ID 範囲を使い切ってしまうと、他のレプリカの ID 範囲に利用可能な ID がないと、新たな ID のリクエストは失敗します。このような場合には、元のレプリカに割り当てられた ID 範囲を拡張します。これを実行するには、既存の ID 範囲を分割したり、サーバーに設定されていた元の ID 範囲を超える拡張を行ったりします。また、新規の ID 範囲を割り当てることもできます。



注記

新規の ID 範囲を割り当てる場合、サーバーまたはレプリカ上の既存エントリーの UID はそのまま変わりません。現行 ID の範囲を変更しても、IdM は過去に割り当てられた範囲のレコードを維持しているため、これが問題になることはありません。

レプリカが機能停止した場合

レプリカが停止して削除する必要がある場合には、ID 範囲は自動的に取得されません。つまり、そのレプリカに割り当てられていた ID 範囲は使用できなくなります。この ID 範囲を回復させて他のレプリカで使えるようにします。

機能停止してしまったサーバーに属していた ID 範囲を回復させて別のサーバーにこれを割り当てるには、まず **ipa-replica-manage dnarange-show** コマンドを使用して ID 範囲の値を確認します。このコマンドについては、「[現在割り当てられている ID 範囲の表示](#)」で説明しています。次に、その ID 範囲をサーバーに手動で割り当てます。また、UID や GID の重複を避けるために、回復させた範囲からの ID の値がこれまでにユーザーやグループに割り当てられていなかったことを確認します。これは、既存のユーザーおよびグループの UID と GID をチェックすることでわかります。

手動で ID 範囲を定義するには、以下の 2 つのコマンドを使用します。

- **ipa-replica-manage dnarange-set** を使用すると、指定されたサーバーの現行 ID 範囲を定義できます。

```
# ipa-replica-manage dnarange-set masterA.example.com 1250-1499
```

- **ipa-replica-manage dnanextrange-set** を使用すると、指定されたサーバーの次の ID 範囲を定義できます。

```
# ipa-replica-manage dnanextrange-set masterB.example.com 1001-5000
```

これらのコマンドについての詳細は、ipa-replica-manage(1) man ページを参照してください。



重要

ID 範囲を重複しないように注意してください。サーバーまたはレプリカに割り当てられた ID 範囲が重複していると、2 つの別のサーバーが同じ ID の値を別のエントリーに割り当てる結果になります。

1000 およびそれ以下の値の UID を含む ID 範囲は設定しないでください。これらの値はシステム用に予約されています。また、0 を含む ID 範囲は設定しないでください。SSSD サービスは 0 ID の値を処理しません。

手動で ID 範囲を拡張する場合は、新たに拡張された範囲が IdM ID 範囲に含まれていることを確認してください。これは **ipa idrange-find** コマンドを使用することで実行できます。**ipa idrange-find -h** コマンドを実行すると **ipa idrange-find** コマンドのヘルプが表示されます。

14.6. ID の値が一意であることを確認する

UID や GID の競合は避けることが推奨されます。UID および GID は常に一意のものにしてください。2 ユーザー間では同じ UID があるべきではなく、2 グループ間でも同じ GID を持たないようにします。

自動での ID 割り当て

ユーザーやグループが対話形式で作成される、または手動で指定した ID 番号なしで作成される場合は、サーバーが次に利用可能な ID 番号を ID 範囲からユーザーアカウントに割り当てます。こうすることで、UID や GID は常に一意のものになります。

手動での ID 割り当て

ID をユーザーまたはグループエントリーに手動で割り当てる場合には、サーバーは指定された UID または GID が一意のものであることを検証しません。他のエントリーで使用されている値を選択しても、サーバーは競合していることを警告しません。

「[変更された UID および GID 番号の修復](#)」の設定にあるように、SSSD サービスは同一 ID のエントリーを処理しません。2 つのエントリーが同じ ID を共有している場合は、この ID の検索では最初のエントリーのみが返されます。ただし、他の属性を検索するか、**ipa user-find --all** コマンドを実行すると、両方のエントリーが返されます。

UID と GID は両方とも同じ ID 範囲から選択されます。ユーザーとグループが同じ ID を持つことは可能です。UID と GID は **uidNumber** および **gidNumber** という別の属性で設定されるため、この状況では競合は発生しません。



注記

同一 ID をユーザーとグループに設定すると、ユーザーのプライベートグループを設定できるようになります。この方法でユーザーに一意のシステムグループを作成するには、同一 ID をユーザーとグループに設定し、このグループのメンバーをこのユーザーのみにします。

14.7. 変更された UID および GID 番号の修復

ユーザーが IdM システムやサービスにログインすると、そのシステム上の SSSD はそのユーザー名をユーザーの UID および GID とキャッシュします。SSSD はその UID をユーザー特定のキーとして使用します。同一のユーザー名で別の UID を持つユーザーがシステムにログインしようとする、SSSD は 2 つの UID を記録し、競合する名前を持つ別の 2 人のユーザーとみなします。この場合、ユーザーの UID が変更されると問題となる可能性があります。この状況では、SSSD は同一ユーザーが別の UID を持っているとは認識せず、変更された UID のユーザーを間違って新規ユーザーとみなします。既存ユーザーの UID が変更されると、ユーザーは SSSD と関連するサービスおよびドメインにログインできなくなります。これは、識別情報に SSSD を使用するクライアントアプリケーションにも影響を及ぼします。

この問題を回避するには、UID や GID が変更された場合に、SSSD キャッシュをクリアします。これでユーザーが再度ログインできるようになります。たとえば、特定ユーザーの SSSD キャッシュをクリアするには、以下のように **sss_cache** ユーティリティを使用します。

```
[root@server ~]# sss_cache -u user
```


第15章 ユーザーおよびグループスキーマ

ユーザーエントリは作成時に自動的に特定の LDAP オブジェクトクラスが割り当てられ、これにより特定の属性が利用可能になります。LDAP 属性は、情報がディレクトリー内に保存される方法です（これについての詳細は、『Directory Server Deployment Guide』 および 『Directory Server Schema Reference』 で説明されています。）

表15.1 デフォルトの Identity Management ユーザーオブジェクトクラス

オブジェクトクラス	詳細
ipaobject ipasshuser	IdM オブジェクトクラス
person organizationalperson inetorgperson inetuser posixAccount	人物のオブジェクトクラス
krbprincipalaux krbticketpolicyaux	Kerberos のオブジェクトクラス
mepOriginEntry	Managed エントリ (テンプレート) のオブジェクトクラス

ユーザーエントリには多くの利用可能な属性があります。手動で設定されるものや、特定の値が設定されてない場合はデフォルト値を元に設定されるものもあります。その属性に UI やコマンドライン引数がない場合でも、表15.1「デフォルトの Identity Management ユーザーオブジェクトクラス」内のオブジェクトクラスで利用できる属性を追加するオプションもあります。また、デフォルトの属性で生成もしくは使用される値は、「デフォルトのユーザーおよびグループ属性の指定」にあるように設定可能です。

表15.2 デフォルトの Identity Management ユーザー属性

UI フィールド	コマンドラインオプション	必須、オプション、またはデフォルト[a]
User login	username	必須
First name	--first	必須
Last name	--last	必須
Full name	--cn	オプション
Display name	--displayname	オプション

UI フィールド	コマンドラインオプション	必須、オプション、またはデフォルト ^[a]
Initials	--initials	デフォルト
Home directory	--homedir	デフォルト
GECOS field	--gecos	デフォルト
Shell	--shell	デフォルト
Kerberos principal	--principal	デフォルト
Email address	--email	オプション
Password	--password ^[b]	オプション
User ID number	--uid	デフォルト
Group ID number	--gidnumber	デフォルト
Street address	--street	オプション
City	--city	オプション
State/Province	--state	オプション
Zip code	--postalcode	オプション
Telephone number	--phone	オプション
Mobile telephone number	--mobile	オプション
Pager number	--pager	オプション
Fax number	--fax	オプション
Organizational unit	--orgunit	オプション
Job title	--title	オプション
Manager	--manager	オプション
Car license	--carlicense	オプション
	--noprivate	オプション

UI フィールド	コマンドラインオプション	必須、オプション、またはデフォルト[a]
SSH Keys	--sshpubkey	オプション
Additional attributes	--addattr	オプション
Department Number	--departmentnumber	オプション
Employee Number	--employeenumber	オプション
Employee Type	--employeetype	オプション
Preferred Language	--preferredlanguage	オプション
<p>[a] 必須の属性は、すべてのエントリーで設定する必要があります。オプションの属性は設定が可能で、デフォルトの属性は特定の値を提供しない場合は事前設定の値で自動的に追加されます。</p> <p>[b] スクリプトは、引数の値を受け付けずに、新たなパスワードを要求します。</p>		

15.1. デフォルトのユーザーおよびグループスキーマの変更

ユーザーおよびグループエントリーに使用されているオブジェクトクラスおよび属性は、変更することができます (15章ユーザーおよびグループスキーマ)。

IdM 設定は、オブジェクトクラスが変更されると以下の確認を行います。

- すべてのオブジェクトクラスとそれらの指定された属性を LDAP サーバーが認識していること。
- エントリーに設定されたデフォルトの属性はすべて、設定済みのオブジェクトクラスにサポートされていること。

ただし、IdM スキーマの検証には限界があります。最も重要なのは、IdM サーバーは定義済みユーザーもしくはグループオブジェクトクラスに IdM エントリーに必要なオブジェクトクラスすべてが含まれているかどうかを確認しないという点です。たとえば、IdM エントリーはすべて、**ipaobject** オブジェクトクラスが必要です。しかし、ユーザーもしくはグループスキーマが変更されると、このオブジェクトクラスが含まれているかどうかをサーバーは検証しません。このオブジェクトクラスが誤って削除されると、それ以降のエントリー追加操作は失敗することになります。

また、すべてのオブジェクトクラス変更は、漸増的ではなくアトミックです。変更があると毎回、デフォルトのオブジェクトクラス一覧全体を定義する必要があります。たとえば、企業が従業員の誕生日や就業開始日などの情報を保存するためのカスタムのオブジェクトクラスを作成したとします。管理者は単にカスタムのオブジェクトクラスをリストに追加するということはできません。新規オブジェクトクラスに加えて現行のデフォルトのオブジェクトクラス一覧全体を設定する必要があります。設定を更新する際は常に、**既存**のデフォルトのオブジェクトクラスを含める必要があります。これを含めないと現行設定が上書きされ、パフォーマンスに関する重大な問題が発生することになります。

15.2. カスタムのオブジェクトクラスを新規ユーザーエントリーに適用する

ユーザーおよびグループアカウントは、エントリーに適用する定義済みの LDAP オブジェクトクラスとともに作成されます。オブジェクトクラスに属する属性は、ユーザーエントリーに追加することができます。

標準および IdM 固有の LDAP オブジェクトクラスはほとんどの場合のデプロイメントのシナリオをカバーしていますが、管理者はカスタマイズされた属性を持つカスタムのオブジェクトクラスを作成することもできます。管理者がデフォルトのオブジェクトクラス一覧を修正すると、新規エントリーにはカスタムオブジェクトクラスが含まれますが、それ以前のエントリーは自動的に修正されないことに注意してください。

15.2.1. Web UI での操作

1. Identity Management が使用する 389 Directory Server インスタンスにカスタムスキーマ要素すべてを追加します。スキーマ要素の追加については、[the schema chapter of the Directory Server Administrator's Guide](#) で説明しています。
2. **IPA Server** タブを開きます。
3. **Configuration** サブタブを選択します。
4. **User Options** エリアまでスクロールします。



The screenshot shows the 'User Options' section of the Identity Management web interface. It contains three configuration items:

- User search fields**: A text input field containing the string 'uid,givenname,sn,telephonenumber,ou,title'.
- Default e-mail domain**: A text input field containing 'example.com' and an 'Undo' button to the right.
- Default users**: A dropdown menu with 'inausers' selected and a blue downward arrow on the right.

図15.1 サーバー設定でのユーザーオプション

5. ユーザーエリアの下部で **Add** をクリックして別のオブジェクトクラスの新規フィールドを追加します。



重要

設定を更新する際は、常に**既存**のデフォルトオブジェクトクラスを含めてください。これらを含めないと、現行設定は上書きされます。Identity Management で必須のオブジェクトクラスが含まれないと、これ以降にエントリーの追加を試みる際にオブジェクトクラス違反で失敗することになります。

The screenshot shows a web form for configuring user object classes. It has a section titled 'Default user *' and 'objectclasses'. Below this, there are four rows, each with a text input field and a 'Delete' button. The input fields contain 'ipaobject', 'person', 'inetuser', and 'posixaccount' respectively. At the bottom of this section, there is an 'Add' button, which is highlighted with a red rectangular box.

図15.2 デフォルトのユーザーオブジェクトクラスの変更

6. 変更が完了したら、**Configuration** ページ上部の **Save** をクリックします。

15.2.2. コマンドラインでの操作

1. Identity Management が使用する 389 Directory Server インスタンスにカスタムスキーマ要素すべてを追加します。スキーマ要素の追加については、[the schema chapter of the Directory Server Administrator's Guide](#) で説明しています。
2. エントリーに追加するオブジェクトクラス一覧に新規オブジェクトクラスを追加します。ユーザーのオブジェクトクラスのオプションは、**--userobjectclasses** です。



重要

設定を更新する際は、常に**既存**のデフォルトオブジェクトクラスを含めてください。これらを含めないと、現行設定は上書きされます。Identity Management で必須のオブジェクトクラスが含まれないと、これ以降にエントリーの追加を試みる際にオブジェクトクラス違反で失敗することになります。

オブジェクトクラス一覧には、すべてのオブジェクトを含める必要があります。**config-mod** コマンドで渡す情報は、それ以前の値を上書きします。これを行うには、**--userobjectclasses** 引数で各オブジェクトクラスを指定するか、**{attr1,attr2,attr3}** のように中括弧内にすべてのオブジェクトクラスをコンマ区切りの一覧で記載します (ただし、スペースはなし)。リストが長くなる場合は特に、複数のオプションよりも中括弧を使用の方が容易です。例を示します。

```
[bjensen@server ~]$ ipa config-mod --
userobjectclasses={top,person,organizationalperson,inetorgperson,inetuser,posixaccount,krbprincipalaux,krbticketpolicyaux,ipaobject,ipashuser,employeeinfo}
```



注記

中括弧オプションを使用するには、**brace expansion** 機能を有効にする必要があります。これには、以下のように **set** コマンドを使用します。

```
# set -o braceexpand
```

15.3. カスタムのオブジェクトクラスを新規グループエントリーに適用する

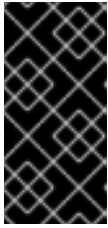
ユーザーエントリーの場合と同様に、管理者はカスタマイズされた属性を持つカスタムのオブジェクトクラスを作成することもできます。オブジェクトクラスを IdM サーバー設定に追加すると、これらは自動的に追加されます。管理者がデフォルトのオブジェクトクラス一覧を修正すると、新規エントリーにはカスタムオブジェクトクラスが含まれますが、それ以前のエントリーは自動的に修正されないことに注意してください。

15.3.1. Web UI での操作

1. Identity Management が使用する 389 Directory Server インスタンスにカスタムスキーマ要素すべてを追加します。スキーマ要素の追加については、[the schema chapter of the Directory Server Administrator's Guide](#) で説明しています。
2. **IPA Server** タブを開きます。
3. **Configuration** サブタブを選択します。
4. **Group Options** エリアまでスクロールします。

図15.3 サーバー設定でのグループオプション

5. **Add** をクリックして別のオブジェクトクラスの新規フィールドを追加します。



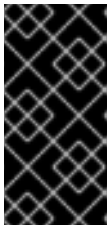
重要

設定を更新する際は、常に**既存**のデフォルトオブジェクトクラスを含めてください。これらを含めないと、現行設定は上書きされます。Identity Management で必須のオブジェクトクラスが含まれないと、これ以降にエントリーの追加を試みる際にオブジェクトクラス違反で失敗することになります。

6. 変更が完了したら、**Configuration** ページ上部の **Save** をクリックします。

15.3.2. コマンドラインでの操作

1. Identity Management が使用する 389 Directory Server インスタンスにカスタムスキーマ要素すべてを追加します。スキーマ要素の追加については、[the schema chapter of the Directory Server Administrator's Guide](#) で説明しています。
2. エントリーに追加するオブジェクトクラス一覧に新規オブジェクトクラスを追加します。グループのオブジェクトクラスのオプションは、**--groupobjectclasses** です。



重要

設定を更新する際は、常に**既存**のデフォルトオブジェクトクラスを含めてください。これらを含めないと、現行設定は上書きされます。Identity Management で必須のオブジェクトクラスが含まれないと、これ以降にエントリーの追加を試みる際にオブジェクトクラス違反で失敗することになります。

オブジェクトクラス一覧には、すべてのオブジェクトを含める必要があります。**config-mod** コマンドで渡す情報は、それ以前の値を上書きします。これを行うには、**--groupobjectclasses** 引数で各オブジェクトクラスを指定するか、**{attr1,attr2,attr3}** のように中括弧内にすべてのオブジェクトクラスをコンマ区切りの一覧で記載します (ただし、スペースはなし)。リストが長くなる場合は特に、複数のオプションよりも中括弧を使用する方が容易です。例を示します。

```
[bjensen@server ~]$ ipa config-mod --
groupobjectclasses={top,groupofnames,nestedgroup,ipausergroup,ipaobject,ipasshuser,employeegroup}
```

15.4. デフォルトのユーザーおよびグループ属性の指定

Identity Management は新規エントリー作成の際にテンプレートを使用します。

ユーザーにとっては、テンプレートは非常に特定のものです。Identity Management は、IdM ユーザーアカウントの中核となる属性のいくつかにはデフォルト値を使用します。これらのデフォルト値はユーザーアカウント属性 (ホームディレクトリーの場所など) の実際の値を定義するか、ユーザー名の長さなどの属性値の形式を定義します。これらの設定は、ユーザーに割り当てられるオブジェクトクラスも定義します。

グループの場合、テンプレートが定義するのは割り当てられたオブジェクトクラスのみです。

これらのデフォルト定義はすべて、IdM サーバーの単一の設定エントリーである **cn=ipaconfig,cn=etc,dc=example,dc=com** に含まれています。

この設定は、**ipa config-mod** コマンドで変更することが可能です。

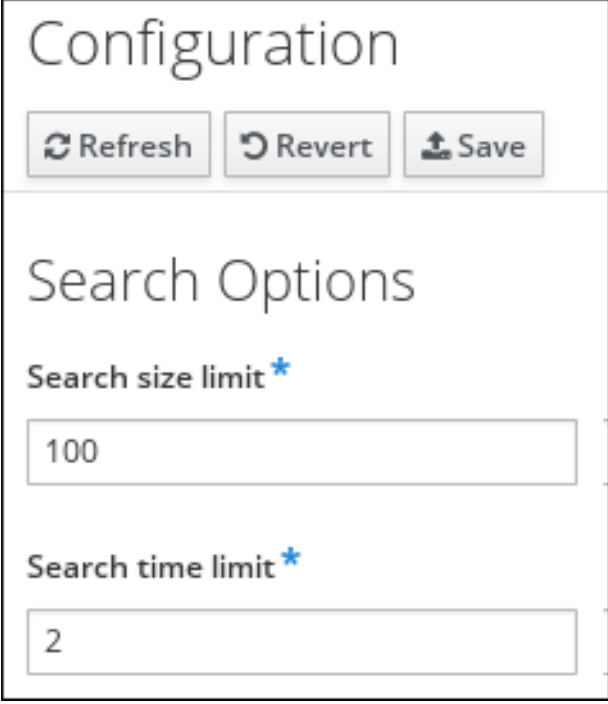
表15.3 デフォルトのユーザーパラメーター

フィールド	コマンドラインオプション	説明
Maximum user name length	--maxusername	ユーザー名の最大文字数を設定します。デフォルト値は 32 です。
Root for home directories	--homedirectory	ユーザーのホームディレクトリーに使用するデフォルトのディレクトリーを設定します。デフォルト値は、 /home です。
Default shell	--defaultshell	ユーザーに使用するデフォルトのシェルを設定します。デフォルト値は、 /bin/sh です。
Default user group	--defaultgroup	新規作成のアカウントを追加するデフォルトグループを設定します。デフォルト値は ipausers で、これは IdM サーバーのインストールプロセスで自動的に作成されます。
Default e-mail domain	--emaildomain	新規アカウントに基づいて電子メールアドレスを作成するために使用する電子メールドメインを設定します。デフォルトは IdM サーバードメインです。
Search time limit	--searchtimelimit	サーバー検索結果を返すまでに費やす最長時間を秒単位で設定します。
Search size limit	--searchrecordslimit	返される検索結果の最大数を設定します。
User search fields	--usersearch	検索文字列として使用可能なユーザーエントリー内のフィールドを設定します。記載される属性にはインデックスがその属性のために維持されるので、多く設定しすぎるとサーバーのパフォーマンスに影響が出る場合があります。
Group search fields	--groupsearch	検索文字列として使用可能なグループエントリー内のフィールドを設定します。

フィールド	コマンドラインオプション	説明
Certificate subject base		クライアント証明書用にサブジェクト DN を作成する際に使用するベース DN を設定します。これはサーバーのセットアップ時に設定されます。
Default user object classes	--userobjectclasses	IdM ユーザーアカウント作成に使用されるオブジェクトクラスを定義します。これは、複数回使用することができます。このリストはコマンド実行時に上書きされるので、オブジェクトクラスの完全一覧を提供する必要があります。
Default group object classes	--groupobjectclasses	IdM グループアカウント作成に使用されるオブジェクトクラスを定義します。これは、複数回使用することができます。このリストはコマンド実行時に上書きされるので、オブジェクトクラスの完全一覧を提供する必要があります。
Password expiration notification	--pwdexpnotify	パスワードの有効期限が切れる何日前にサーバーが通知を送信するかを設定します
Password plug-in features		ユーザーが使用可能なパスワードの形式を設定します。

15.4.1. Web UI で属性を表示する

1. **IPA Server** タブを開きます。
2. **Configuration** サブタブを選択します。
3. 設定エントリーは、全検索の制限、ユーザーテンプレート、およびグループテンプレートの 3 つのセクションで表示されます。



Configuration

Refresh Revert Save

Search Options


Search size limit *

100

Search time limit *

2

図15.4 検索での制限設定

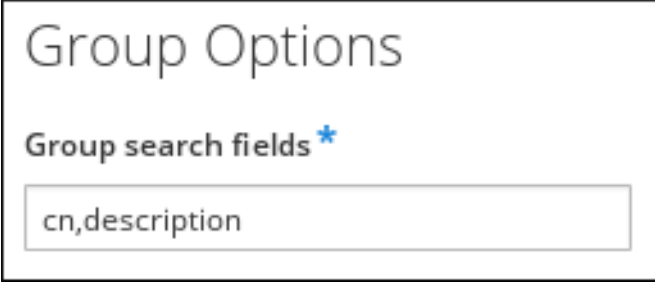


User Options

User search fields *

uid,givenname,sn,telephonenumber,ou,1

図15.5 ユーザー属性



Group Options

Group search fields *

cn,description

図15.6 グループ属性

15.4.2. コマンドラインで属性を表示する

config-show コマンドを使うと、すべての新規ユーザーアカウントに適用される現行設定が表示されます。デフォルトでは、最も一般的な属性のみが表示されます。**--all** オプションを使うと、完全な設定が表示されます。

```
[bjensen@server ~]$ kinit admin
[bjensen@server ~]$ ipa config-show --all
dn: cn=ipaConfig,cn=etc,dc=example,dc=com
Maximum username length: 32
Home directory base: /home
```



```
Default shell: /bin/sh
Default users group: ipausers
Default e-mail domain: example.com
Search time limit: 2
Search size limit: 100
User search fields: uid,givenname,sn,telephonenumber,ou,title
Group search fields: cn,description
Enable migration mode: FALSE
Certificate Subject base: O=EXAMPLE.COM
Default group objectclasses: top, groupofnames, nestedgroup, ipausergroup,
ipaobject
Default user objectclasses: top, person, organizationalperson,
inetorgperson, inetuser, posixaccount, krbprincipalaux,
krbticketpolicyaux, ipaobject, ipasshuser
Password Expiration Notification (days): 4
Password plugin features: AllowNThash
SELinux user map order: guest_u:s0$xguest_u:s0$user_u:s0$staff_u:s0-
s0:c0.c1023$unconfined_u:s0-s0:c0.c1023
Default SELinux user: unconfined_u:s0-s0:c0.c1023
Default PAC types: MS-PAC, nfs:NONE
cn: ipaConfig
objectclass: nsContainer, top, ipaGuiConfig, ipaConfigObject
```

第16章 サービスの管理

ホスト上で実行されるサービスには、IdM ドメインに属するものもあります。Kerberos プリンシパルまたは SSL 証明書のいずれか (またはこれら両方) を保存することができるサービスは、IdM サービスとして設定することができます。IdM ドメインにサービスを追加すると、そのサービスはドメインから SSL 証明書や keytab を要求することができます。(証明書の公開鍵のみがサービスレコードに保存されます。秘密鍵はサービスのローカルになります。)

An IdM ドメインは、共通の ID 情報、ポリシー、共有 サービスで、マシン間に共通性を確立します。あるドメインに所属しているマシンは、そのドメインのクライアントとして機能するので、そのドメインが提供するサービスを使用できます。(1章 *Red Hat Identity Management について* の説明にあるように) An IdM ドメインは特に、以下の 3 つの主要サービスをマシンに提供します。

- DNS
- Kerberos
- 証明書管理

16.1. サービスエントリおよび **KEYTAB** の追加と編集

ホストエントリの場合と同様に、ホストのサービスエントリ (およびドメインに属するホスト上のサービス) は手動で IdM ドメインに追加する必要があります。これは 2 段階のプロセスで、最初にサービスエントリを作成し、次にそのサービスがドメインへのアクセスに使用する keytab を作成します。

デフォルトでは、Identity Management は HTTP keytab を **/etc/httpd/conf/ipa.keytab** に保存します。



注記

この keytab は Web UI に使用します。キーが **ipa.keytab** に保存されその keytab ファイルが削除された場合、元のキーも削除されてしまうので、IdM Web UI は機能しなくなります。

Kerberos 対応とする必要のある各サービスで、同様の場所を指定することができます。特定の場所を使用する必要はありませんが、**ipa-getkeytab** を使用する場合は、**/etc/krb5.keytab** を避けてください。このファイルにはサービス固有の keytab を含めるべきではありません。各サービスは keytab を特定の場所に保存し、そのサービスのみが keytab にアクセスできるようにアクセス権限 (および場合によっては SELinux ルール) を設定します。

16.1.1. Web UI でのサービスと **Keytab** の追加

1. **Identity** タブを開き、**Services** サブタブを選択します。
2. サービス一覧の上部にある **Add** ボタンをクリックします。
3. ドロップダウンメニューからサービスタイプを選択し、名前を付けます。
4. サービスが実行される IdM ホストのホスト名を選択します。ホスト名を使用して、完全なサービスプリンシパル名を構成します。
5. **Add** ボタンをクリックして、新しいサービスプリンシパルを保存します。

6. **ipa-getkeytab** コマンドを使用して、サービスプリンシパルの新規 keytab を生成、割り当てます。

```
[root@ipaserver ~]# # ipa-getkeytab -s ipaserver.example.com -p
HTTP/server.example.com -k /etc/httpd/conf/krb5.keytab -e aes256-cts
```

- レルム名はオプションです。IdM サーバーは、設定される Kerberos レルムを自動的に追加します。別のレルムを指定することはできません。
- Kerberos と連携させるには、DNS A レコードに対してホスト名を解決する必要があります。必要な場合は、**--force** フラグを使用して強制的にプリンシパルを作成することができます。
- **-e** 引数では、keytab に含める暗号タイプの一覧を提供します。これはデフォルトの暗号化タイプに優先します。エントリーのリストは、同一コマンドでオプションを複数回使用するか、**--option={val1,val2,val3}** のように中括弧内にオプションをコンマ区切りの一覧で記載します。



警告

新たなキーを作成すると、指定されたプリンシパルの秘密がリセットされます。つまり、そのプリンシパルの他の keytab すべてが無効になります。

16.1.2. コマンドラインでのサービスと **Keytab** の追加

1. サービスプリンシパルを作成します。サービスは、**service/FQDN** といった名前で認識されます。

```
# ipa service-add serviceName/hostname
```

例を示します。

```
$ ipa service-add HTTP/server.example.com
-----
Added service "HTTP/server.example.com@EXAMPLE.COM"
-----
Principal: HTTP/server.example.com@EXAMPLE.COM
Managed by: ipaserver.example.com
```

2. **ipa-getkeytab** コマンドでサービス keytab ファイルを作成します。このコマンドは、IdM ドメイン内のクライアント上で実行します。(実際には、IdM サーバーまたはクライアント上でコマンドを実行して、キーを適切なマシンにコピーすることが可能です。ただし、サービスが作成されるマシン上でこのコマンドを実行することが最もシンプルな方法です。)

このコマンドでは、Kerberos サービスプリンシパル (**-p**)、IdM サーバー名 (**-s**)、書き込みファイル (**-k**)、および暗号化方法 (**-e**) が必要になります。keytab をサービスの適切なディレクトリーにコピーしてください。

例を示します。

```
# ipa-getkeytab -s server.example.com -p HTTP/server.example.com -k
/etc/httpd/conf/krb5.keytab -e aes256-cts
```

- レルム名はオプションです。IdM サーバーは、設定される Kerberos レルムを自動的に追加します。別のレルムを指定することはできません。
- Kerberos と連携させるには、DNS A レコードに対してホスト名を解決する必要があります。必要な場合は、**--force** フラグを使用して強制的にプリンシパルを作成することができます。
- **-e** 引数では、keytab に含める暗号タイプをコンマ区切りのリストで提供します。これはデフォルトの暗号化タイプに優先します。エントリーのリストは、同一コマンドでオプションを複数回使用するか、**--option={val1,val2,val3}** のように中括弧内にオプションをコンマ区切りの一覧で記載します。



警告

ipa-getkeytab コマンドは、指定されたプリンシパルの秘密をリセットします。つまり、そのプリンシパルの他の keytab すべてが無効になります。

16.2. クラスタサービスの設定

IdM サーバーは、クラスタに対応していません。ただし、Kerberos キーを参加サービスすべてにわたって同期させ、ホスト上で実行中のサービスをクライアントが使用する名前に対応するように設定すると、クラスタサービスを IdM の一部として設定することができます。

1. クラスタ内の全ホストを IdM ドメインに登録します。
2. サービスプリンシパルを作成し、必要な keytab を生成します。
3. **/etc/krb5.keytab** にあるホスト keytab を含む、ホスト上のサービスに設定されたすべての keytab を収集します。
4. **ktutil** コマンドを使って、全 keytab ファイルのコンテンツを含む単一の keytab ファイルを作成します。
 1. 各ファイルで **rkt** コマンドを使ってそのファイルからキーを読み取ります。
 2. 新規 keytab ファイルに読み込まれたキーすべてを書き込むには、**wkt** コマンドを使用します。
5. 各ホスト上の keytab ファイルを新たに作成した結合 keytab ファイルで置き換えます。
6. この時点で、このクラスター内の各ホストは他のホストに偽装することができます。
7. 失敗したサービスを引き継ぐ際にホスト名をリセットしないクラスターメンバーに対応するために、追加の設定が必要になるサービスが複数あります。
 - **sshd** では、**/etc/ssh/sshd_config** 内で **GSSAPIStrictAcceptorCheck no** と設定します。

- `mod_auth_kerb` では、`/etc/httpd/conf.d/auth_kerb.conf` 内で `KrbServiceName Any` と設定します。



注記

SSL サーバーの場合、サーバー証明書のサブジェクト名もしくはサブジェクト代替名は、クライアントがクラスター化したホストに接続する際に、正しく表示される必要があります。可能であれば、全ホスト間で秘密キーを共有してください。

各クラスターメンバーに、他のクラスターメンバーすべての名前を含んでいるサブジェクト代替名が含まれている場合、それでクライアントの接続要件が満たされます。

16.3. 複数サービスでの同一サービスプリンシパルの使用

クラスター内では、異なるマシンに分散している複数サービスに同一のサービスプリンシパルを使用することができます。

1. `ipa-getkeytab` コマンドでサービスプリンシパルを取得します。

```
# ipa-getkeytab -s kdc.example.com -p HTTP/server.example.com -k
/etc/httpd/conf/krb5.keytab -e aes256-cts
```

2. 複数サーバーまたはサービスに同一ファイルを使用するよう指示するか、必要に応じてそのファイルを個別サーバーにコピーします。

16.4. 複数のサーバー向けの既存の **KEYTAB** 取得

クラスターの環境などのシナリオでは、共通のホスト名を使用するサービスに対して、異なるマシンで同じ keytab ファイルが必要な場合があります。各ホストで同じ keytab を取得するには、IdM コマンドを使用することができます。

共通のホスト名とサービスプリンシパルを準備するには、IdM サーバーで以下のコマンドを実行します。

1. **admin** ユーザーとして認証を行います。

```
[root@ipaserver ~]# kinit admin
```

2. このホスト名を共有する全 IP アドレスに対して、共通の正引き DNS レコードを追加します。

```
[root@ipaserver ~]# ipa dnsrecord-add idm.example.com cluster --a-
rec={192.0.2.40,192.0.2.41}
Record name: cluster
A record: 192.0.2.40, 192.0.2.41
```

3. 共通の DNS 名に対して新規ホストエントリーオブジェクトを作成します。

```
[root@ipaserver ~]# ipa host-add cluster.idm.example.com
-----
Added host "cluster.idm.example.com"
-----
Host name: cluster.idm.example.com
Principal name: host/cluster.idm.example.com@IDM.EXAMPLE.COM
```

```

Password: False
Keytab: False
Managed by: cluster.idm.example.com

```

4. ホストのサービスプリンシパルを追加します。

```

[root@ipaserver ~]# ipa service-add HTTP/cluster.idm.example.com
-----
Added service "HTTP/cluster.idm.example.com@IDM.EXAMPLE.COM"
-----
Principal: HTTP/cluster.idm.example.com@IDM.EXAMPLE.COM
Managed by: cluster.idm.example.com

```

5. IdM から keytab を取得できるようにサービスにホストを追加します。

```

[root@ipaserver ~]# ipa service-allow-retrieve-keytab
HTTP/cluster.idm.example.com --hosts=
{node01.idm.example.com,node02.idm.example.com}
Principal: HTTP/cluster.idm.example.com@IDM.EXAMPLE.COM
Managed by: cluster.idm.example.com
Hosts allowed to retrieve keytab: node01.idm.example.com,
node02.idm.example.com
-----
Number of members added 2
-----

```

6. 新規 keytab 作成パーミッションをホスト 1 台に割り当てます。

```

[root@ipaserver ~]# ipa service-allow-create-keytab
HTTP/cluster.idm.example.com --hosts=node01.idm.example.com
Principal: HTTP/cluster.idm.example.com@IDM.EXAMPLE.COM
Managed by: cluster.idm.example.com
Hosts allowed to retrieve keytab: node01.idm.example.com,
node02.idm.example.com
Hosts allowed to create keytab: node01.idm.example.com
-----
Number of members added 1
-----

```

クライアントで以下の手順を実行します。

1. ホストの Kerberos keytab で認証を行います。

```
# kinit -kt /etc/krb5.keytab
```

2. 1. 適切なパーミッションを割り当てたクライアントで、新規 keytab を生成して、ファイルに保存します。

```

[root@node01 ~]# ipa-getkeytab -s ipaserver.idm.example.com -p
HTTP/cluster.idm.example.com -k /tmp/client.keytab

```

2. このコマンドに **-r** オプションを追加して、IdM サーバーから既存の keytab を取得します。

```
[root@node02 ~]# ipa-getkeytab -r -s ipaserver.idm.example.com -p  
HTTP/cluster.idm.example.com -k /tmp/client.keytab
```



警告

-r オプションを省略すると、新しい keytab が生成される点に注意してください。これにより、このサービスプリンシパル用に以前に取得した keytab はすべて無効になります。

16.5. サービスエントリーの無効化および再有効化

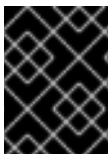
アクティブなサービスは、ドメイン内の他のサービスやホスト、ユーザーからアクセス可能です。アクティビティからホストやサービスを削除する必要がある場合もあります。ただし、サービスやホストを削除するとエントリーやすべての関連する設定も永続的に削除されてしまいます。

16.5.1. サービスエントリーの無効化

サービスを無効にすると、サービスをドメインから永久に削除することなくドメインユーザーがサービスにアクセスすることを防ぎます。これは **service-disable** コマンドを使用することで実行できます。

サービスを無効にするには、サービスのプリンシパルを指定します。

```
[jsmith@ipaserver ~]$ kinit admin  
[jsmith@ipaserver ~]$ ipa service-disable HTTP/server.example.com
```



重要

ホストエントリーを無効にすると、ホストだけでなくそのホスト上で設定されているすべてのサービスが無効になります。

16.5.2. サービスの再有効化

サービスを無効にすると、現行のアクティブな keytab を強制終了することになります。keytab を削除すると、設定エントリーに触れることなく IdM ドメインから該当サービスが削除されます。

サービスを再度有効にするには、**ipa-getkeytab** コマンドを使用するだけです。**-s** オプションはどの IdM サーバーに keytab を要求するかを設定し、**-p** はプリンシパル名を提示し、**-k** では keytab を保存するファイルを提供します。

新規の HTTP keytab を要求する場合は、以下のようになります。

```
[root@ipaserver ~]# ipa-getkeytab -s ipaserver.example.com -p  
HTTP/server.example.com -k /etc/httpd/conf/krb5.keytab -e aes256-cts
```

第17章 ユーザーアクセスのホストおよびサービスへの委任

管理 とは別のホストやサービスの keytab および証明書を取得できることを指します。すべてのホストとサービスには **managedby** エントリーがあり、これにホストやサービスが管理可能なものが記載されています。デフォルトでは、ホストはホスト自体とそのサービスすべてを管理できます。また、適切な委任更新や、適切な **managedby** エントリーの提供により、ホストが他のホストや他のホスト上のサービスを管理可能とすることもできます。

IdM サービスは、そのサービスへのアクセス許可が付与、もしくは**委任** されている IdM ホストであれば、どのホストからでも管理できます。同様に、ホストにはドメイン内の他のホストへの許可を委任することが可能です。

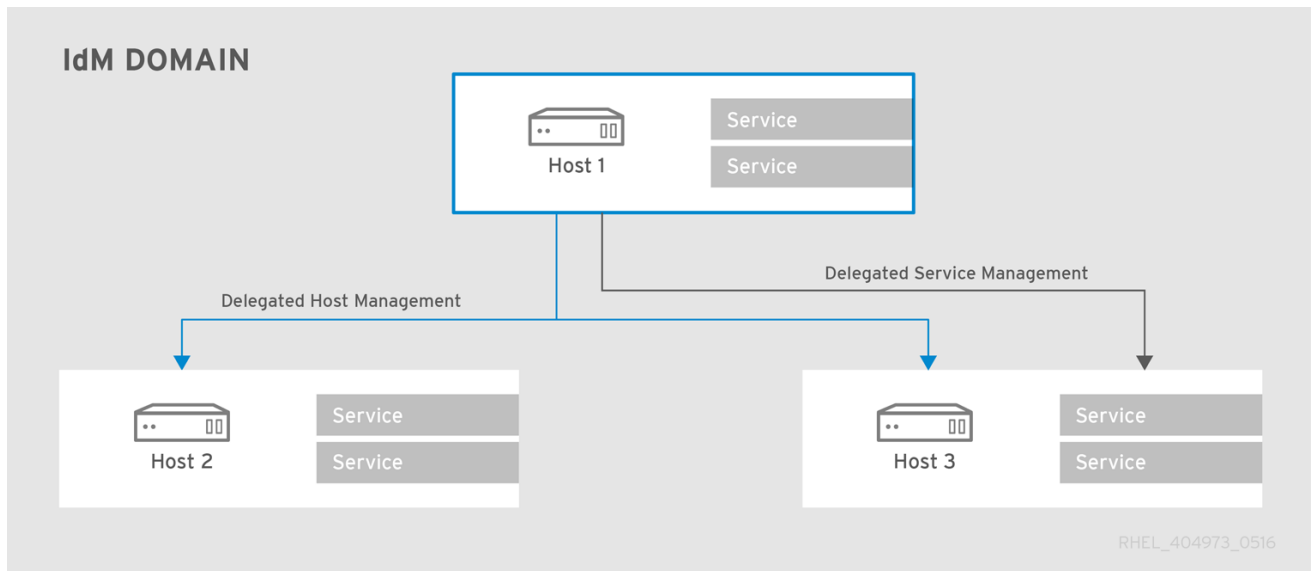


図17.1 ホストおよびサービスの委任



注記

managedBy エントリーで別のホストに権限が委任されている場合に、そのホスト上の全サービスの管理を委任されたわけではありません。委任は個別に行われる必要があります。

17.1. サービス管理の委任

ホストにサービスの制御を委任するには、**service-add-host** コマンドを使用します。サービスの委任は、プリンシパルの特定と制御を行うホストの特定という 2 つの部分で構成されます。

```
# ipa service-add-host principal --hosts=hostnames
```

例を示します。

```
[root@server ~]# ipa service-add-host HTTP/web.example.com --
hosts=client1.example.com
```

ホストに権限が委任されれば、ホストプリンシパルを使ってサービスを管理できます。

```
[root@server ~]# kinit -kt /etc/krb5.keytab host/`hostname`
# ipa-getkeytab -s `hostname` -k /tmp/test.keytab -p HTTP/web.example.com
Keytab successfully retrieved and stored in: /tmp/test.keytab
```


このサービスにチケットを作成するには、委任された権限のあるホストで証明書リクエストを作成し、**cert-request** コマンドを使ってサービスエントリーを作成して証明書情報を読み込みます。

```
[root@server ~]# ipa cert-request --add --principal=HTTP/web.example.com
web.csr
Certificate: MIICETCCAXqgA...[snip]
Subject: CN=web.example.com,O=EXAMPLE.COM
Issuer: CN=EXAMPLE.COM Certificate Authority
Not Before: Tue Feb 08 18:51:51 2011 UTC
Not After: Mon Feb 08 18:51:51 2016 UTC
Serial number: 1005
```

証明書要求の作成や **ipa cert-request** の使用に関する詳しい情報は、「[ユーザー、ホスト、またはサービス向けの新規証明書のリクエスト](#)」を参照してください。

17.2. ホスト管理の委任

ホストに他のホストへの権限を委任するには、**host-add-managedby** コマンドを使用します。これで **managedby** エントリーが作成されます。**managedby** エントリーが作成されると、ホストは権限を委任された別のホストの keytab を取得することができます。

1. 管理者ユーザーとしてログインします。

```
[root@server ~]# kinit admin
```

2. **managedby** エントリーを追加します。たとえば、このコマンドでは、*client2* から *client1* へ権限を委任します。

```
[root@server ~]# ipa host-add-managedby client2.example.com --
hosts=client1.example.com
```

3. ホスト **client1** としてチケットを取得し、**client2** の keytab を取得します。

```
[root@server ~]# kinit -kt /etc/krb5.keytab host/`hostname`
[root@server ~]# ipa-getkeytab -s `hostname` -k /tmp/client2.keytab
-p host/client2.example.com
Keytab successfully retrieved and stored in: /tmp/client2.keytab
```

17.3. WEB UI を使ったホストまたはサービス管理の委任

各ホストおよびサービスエントリーには、どのホストがホストやサービスの管理を委任されているかを示す説明タブがあります。

1. **Identity** タブを開き、**Hosts** または **Services** サブタブを選択します。
2. 委任管理を付与する先となるホストもしくはサービス名をクリックします。
3. ホスト/サービスエントリーの右端にある **Hosts** サブタブをクリックします。このタブでは、選択したホスト/サービスを **管理できる** ホストが表示されます。

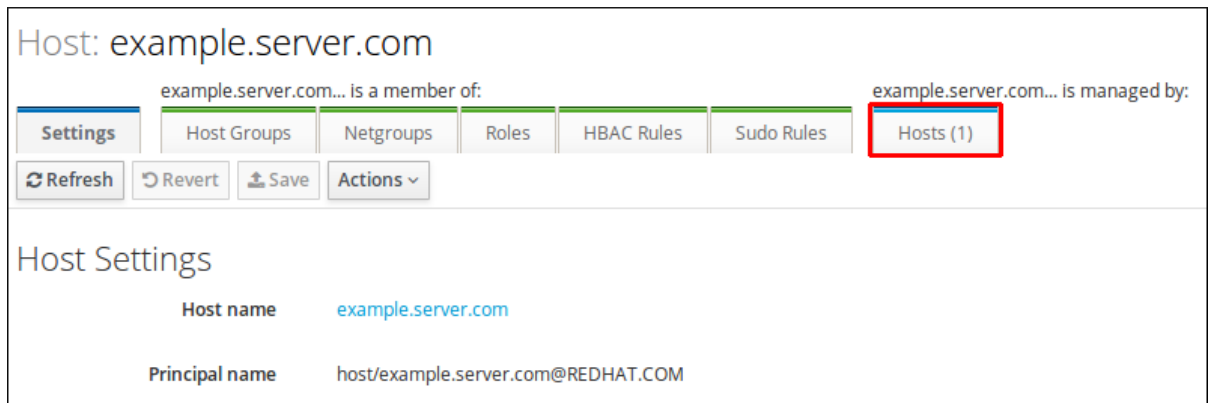


図17.2 ホストのサブタブ

4. 一覧上部にある **Add** をクリックします。
5. ホスト/サービスの管理を委任するホスト名の横にあるチェックボックスを選択し、右矢印 > をクリックして、そのホスト名を選択ボックスに移動します。

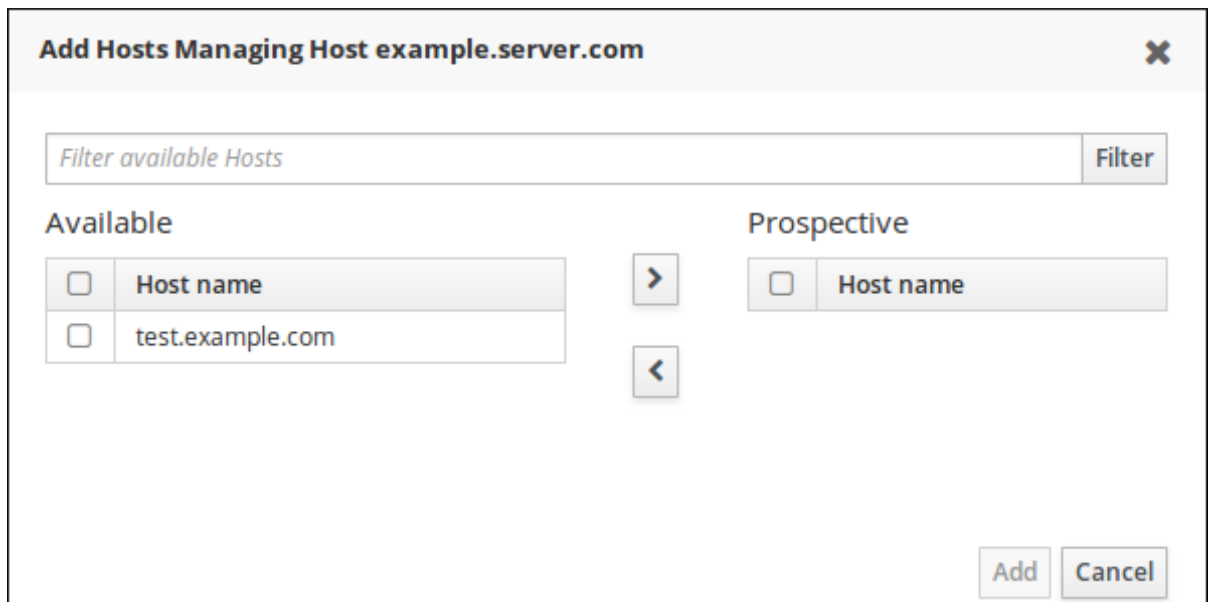


図17.3 ホスト/サービスの委任管理

6. **Add** をクリックして選択ボックスを閉じ、委任設定を保存します。

17.4. 委任サービスへのアクセス

サービスおよびホストの両方において、クライアントに権限が委任されている場合、ローカルマシン上でそのプリンシパルの keytab を取得することができます。サービスの場合は **service/hostname@REALM** という形式になり、ホストの場合は **service** を **host** で置き換えます。

kinit で **-k** オプションを使って keytab を読み込み、**-t** オプションで keytab を指定します。

たとえば、ホストにアクセスするには以下のコマンドを実行します。

```
[root@server ~]# kinit -kt /etc/krb5.keytab
host/ipa.example.com@EXAMPLE.COM
```

サービスにアクセスするには以下のコマンドを実行します。

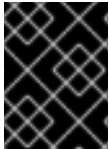
```
[root@server ~]# kinit -kt /etc/httpd/conf/krb5.keytab  
HTTP/ipa.example.com@EXAMPLE.COM
```

第18章 ID ビュー

ID ビューを使用すると、POSIX ユーザーもしくはグループ属性の新規の値を指定でき、どのクライアントホストに新たな値を適用するかを定義することができます。

ID ビューを使用すると、以下のようなことが可能になります。

- 環境ごとに異なる属性値を定義する。詳細は「[ホストごとにユーザーアカウントで異なる属性値を定義する](#)」を参照してください。
- 以前に生成された属性値を異なる値で置き換える。



重要

ID ビューが適用できるのは IdM クライアントのみで、IdM サーバーには適用できません。

SSSD パフォーマンスへのマイナス影響の可能性

ID ビューを適用すると、特定の最適化と ID ビューが同時に実行できなくなるので、SSSD パフォーマンスにマイナス影響が出る可能性があります。たとえば ID ビューは、SSSD によるサーバー上でグループルックアップのプロセス最適化を妨げます。

- ID ビューを使用すると、グループ名が上書きされた場合、SSSD は返されたグループメンバー名リストの各メンバーをチェックする必要があります。
- ID ビューを使用しないと、SSSD はグループオブジェクトのメンバー属性からユーザー名を収集するだけで済みます。

このマイナス影響は、SSSD キャッシュが空であるか、キャッシュをクリアした後 (この場合は全エントリーが無効になります) に顕著になります。

その他のリソース

ID ビューには、Active Directory が関連する環境でのユースケースもあります。詳細は、『Windows 統合ガイド』の [Active Directory 環境での ID ビューの使用](#) を参照してください。

18.1. ID ビューで上書き可能な属性

ID ビューはユーザーおよびグループ ID の上書きで構成されます。上書きは、新規属性値を定義します。

ユーザーおよびグループ ID の上書きで新たな値を定義できるのは、以下の属性です。

ユーザー属性

- ログイン名 (**uid**)
- GECOS エントリー (**gecos**)
- UID 番号 (**uidNumber**)
- GID 番号 (**gidNumber**)
- ログインシェル (**loginShell**)
- ホームディレクトリー (**homeDirectory**)

- SSH 公開キー (**ipaSshPubkey**)
- 証明書 (**userCertificate**)

グループ属性

- グループ名 (**cn**)
- グループ GID 番号 (**gidNumber**)

18.2. ID ビューコマンドのヘルプ

ID ビューおよび上書きに使用する全コマンドを表示するには、以下を実行します。

```
$ ipa help idviews
```

特定コマンドの詳細なヘルプを表示するには、そのコマンドに **--help** オプションを加えて実行します。

```
$ ipa idview-add --help
```

18.3. ホストごとにユーザーアカウントで異なる属性値を定義する

管理者は、ユーザーアカウントで使用される属性値を上書きする複数の ID ビューを作成し、これらの ID ビューを異なるクライアントホストに適用することが可能です。たとえば、サービスアカウントは、異なるホストで認証を行う場合に、異なる SSH 公開キーを使用するように設定できます。

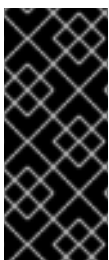
本セクションでは、以下の手順を説明します。

- 「[Web UI: 特定ホスト向けの属性値の上書き](#)」
- 「[コマンドライン: 特定ホスト向けの属性値の上書き](#)」

これらの手順では、**host1.example.com** という名前のクライアントホスト用に ID ビューを作成する方法について説明します。他のホストにある属性値も上書きする場合には、各ホストごとに 1 つずつ、複数の ID ビューを作成する手順を使用してください。

手順では、以下を前提としています。

- **user** は、属性を上書きするユーザーアカウントになります。
- **host1.example.com** は、ID ビューを適用するホストになります。



重要

新規 ID ビューを作成したら、ID ビューを適用する全クライアントで SSSD を再起動します。

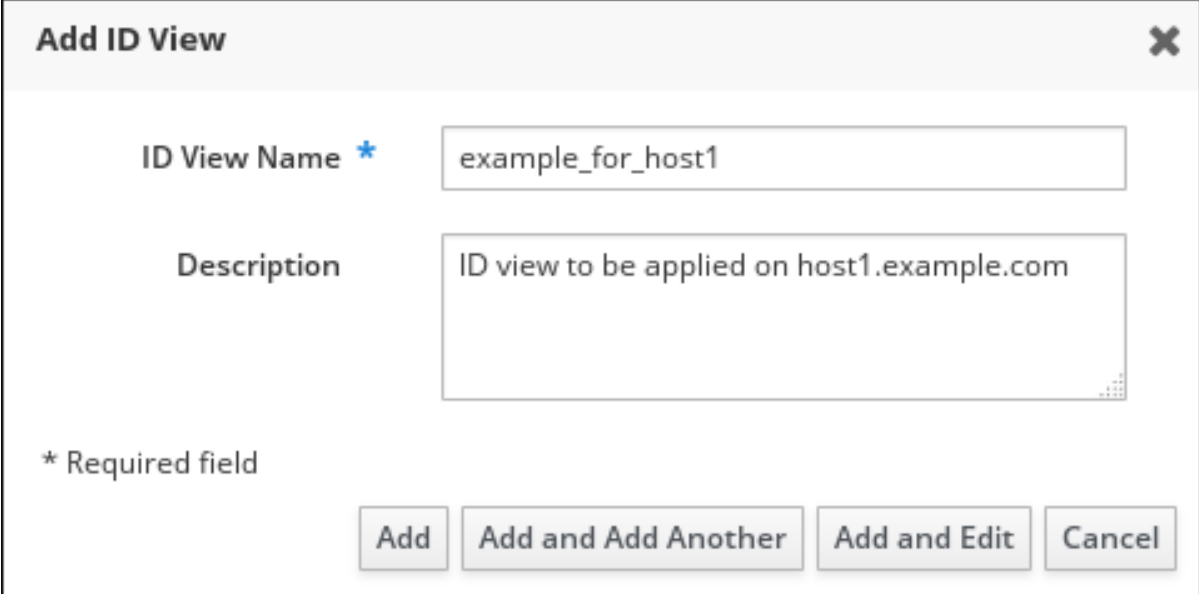
新規 ID ビューが UID または GID を変更した場合は、クライアントで SSSD キャッシュもクリックしてください。

18.3.1. Web UI: 特定ホスト向けの属性値の上書き

ID ビューを管理する前に、**admin** など必要な権限のあるユーザーとして IdM web UI にログインします。

新規 ID ビューの作成

1. **Identity** タブで **ID Views** サブタブを選択します。
2. **Add** をクリックして ID ビューの名前を記入します。



Add ID View [Close]

ID View Name *

Description

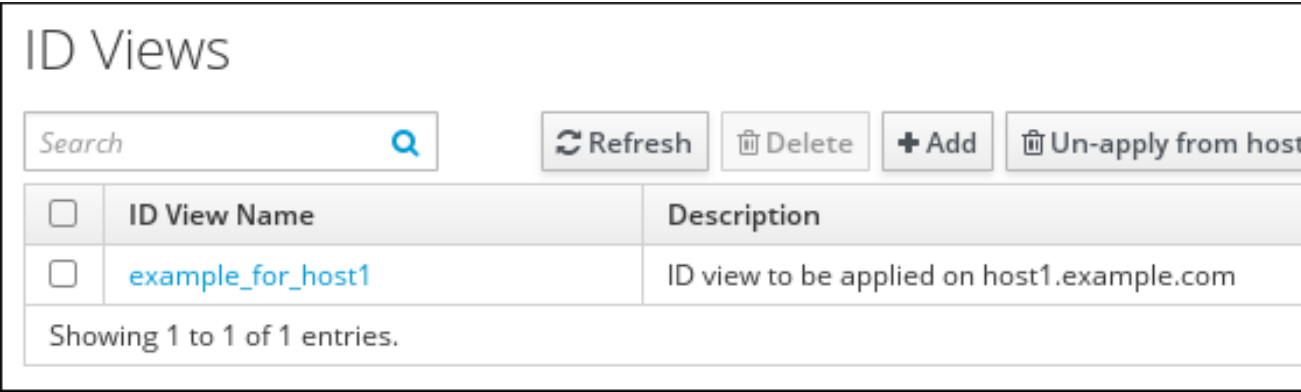
* Required field

[Add] [Add and Add Another] [Add and Edit] [Cancel]

図18.1 ID ビューの追加

3. **Add** をクリックして確定します。

これで新規 ID ビューが ID ビュー一覧に表示されます。



ID Views

Search [Q] [Refresh] [Delete] [Add] [Un-apply from host]

<input type="checkbox"/>	ID View Name	Description
<input type="checkbox"/>	example_for_host1	ID view to be applied on host1.example.com

Showing 1 to 1 of 1 entries.

図18.2 ID ビュー一覧

ユーザー上書きを ID ビューに追加する

1. ID ビュー一覧で、ID ビューの名前をクリックします。

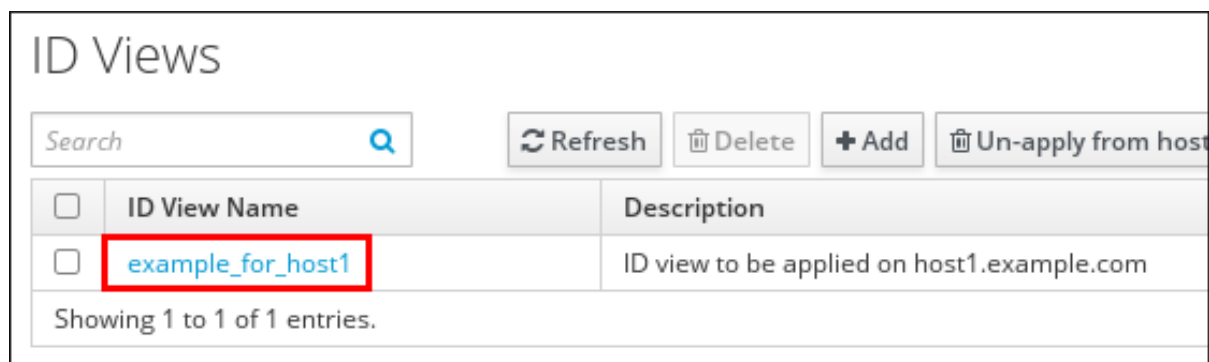


図18.3 ID ビューの編集

2. **Users** タブで **Add** をクリックしてユーザー上書きを追加します。
3. 属性値を上書きするユーザーアカウントを選択して、**Add** をクリックします。

これでユーザー上書きが **example_for_host1** の ID ビューページに表示されます。

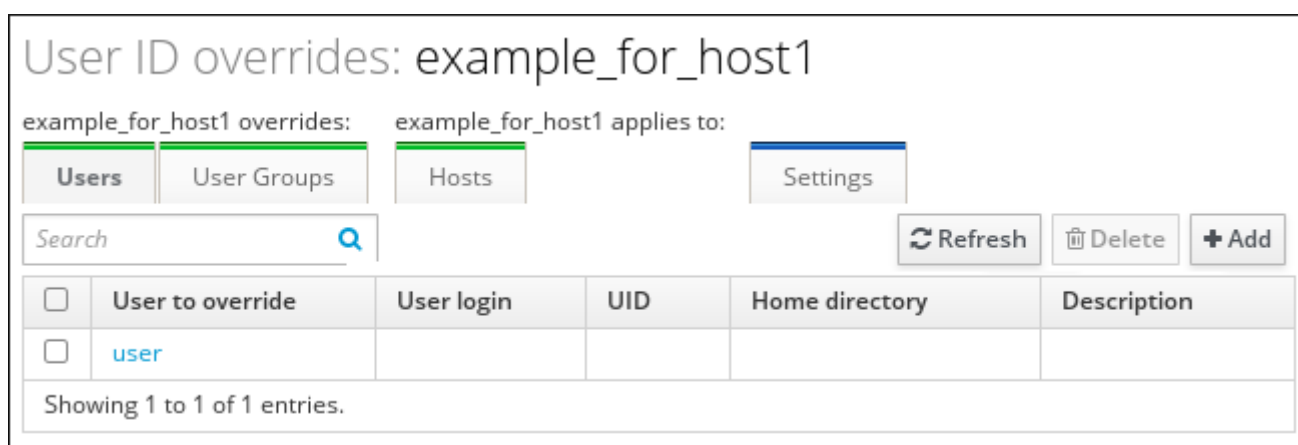


図18.4 上書き一覧

上書きする属性の指定

1. 属性値を変更する上書きをクリックします。

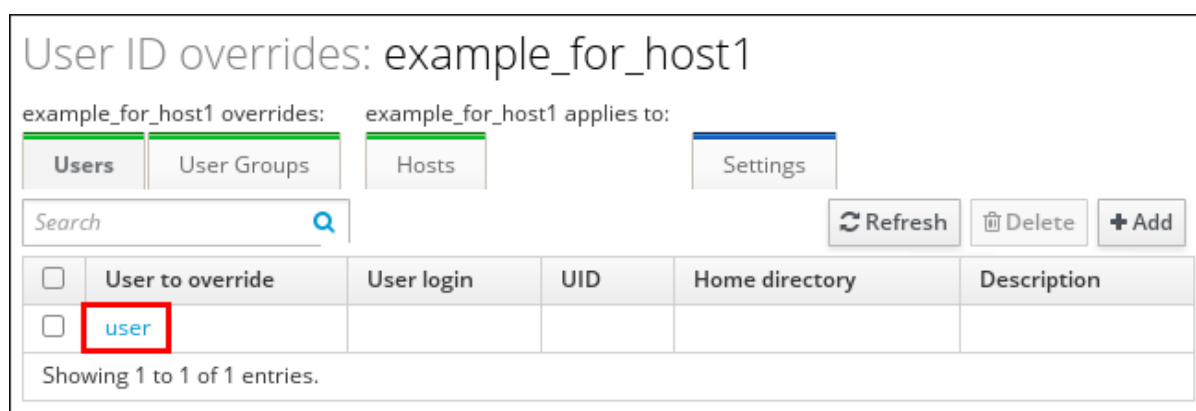
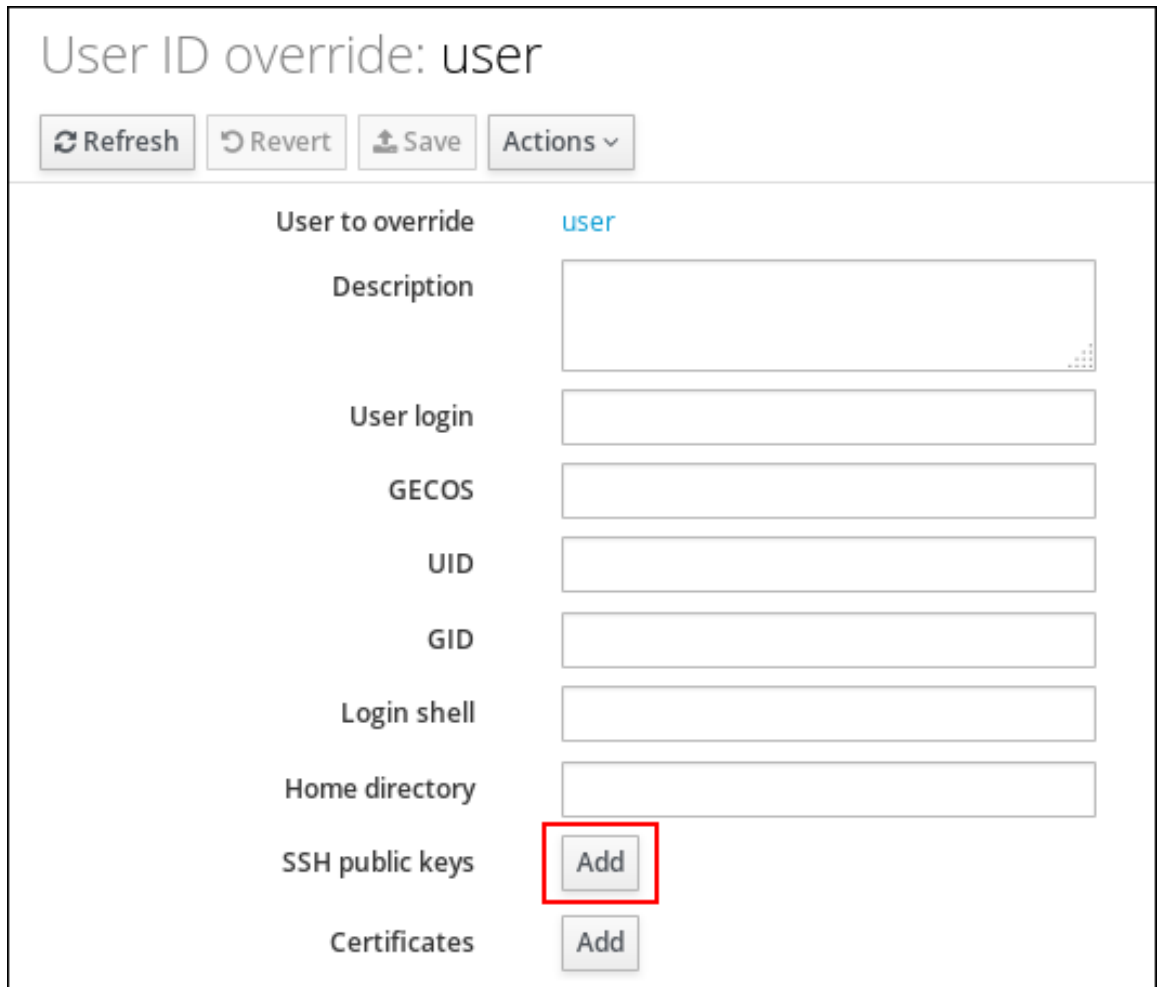


図18.5 上書きの編集

2. 新しい属性値を定義します。

たとえば、ユーザーアカウントで使用する SSH 公開キーを上書きするには、以下を実行します。

- a. **SSH public keys: Add** をクリックします。



User ID override: user

Refresh Revert Save Actions ▾

User to override user

Description

User login

GECOS

UID

GID

Login shell

Home directory

SSH public keys Add

Certificates Add

図18.6 SSH 公開キーの追加

- b. 公開キーに貼り付けます。



注記

IdM へのSSH キーの追加に関する詳細は、[「ユーザーの公開 SSH キーの管理」](#)を参照してください。

3. **Save** をクリックして上書きを更新します。

ID ビューの特定ホストへの適用

1. ID ビュー一覧で、ID ビューの名前をクリックします。

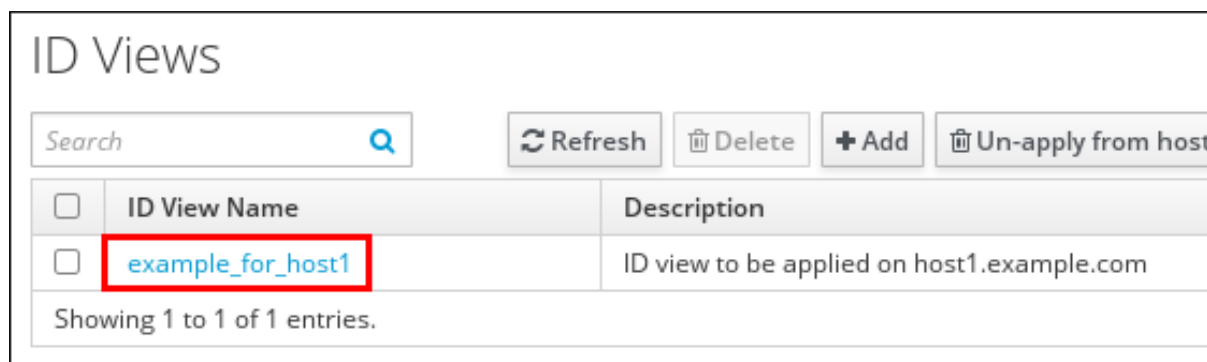


図18.7 ID ビューの編集

2. **Hosts** タブで **Apply to hosts** をクリックします。
3. **host1.example.com** ホストを選択して、**Prospective** コラムに移動します。
4. **Apply** をクリックします。

これでこのホストは ID ビューが適用されるホスト一覧に表示されます。

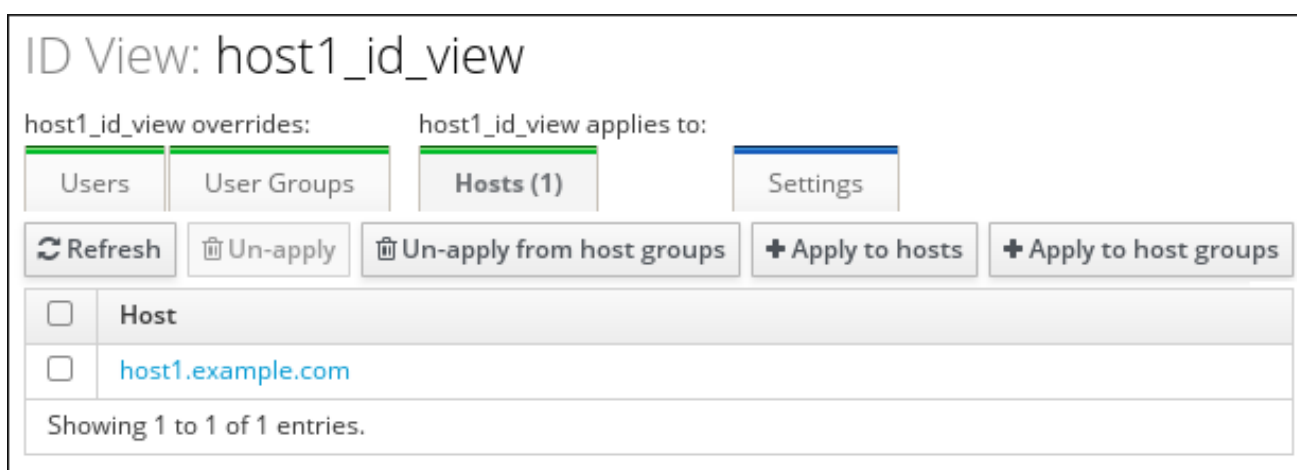


図18.8 ID ビューが適用されるホスト一覧

18.3.2. コマンドライン: 特定ホスト向けの属性値の上書き

ID ビューを管理する前に、以下のように必要な権限のあるユーザー用のチケットをリクエストします。

```
$ kinit admin
```

1. 新規 ID ビューを作成します。たとえば、**example_for_host1** という名前の ID ビューを作成するには、以下を実行します。

```
$ ipa idview-add example_for_host1
-----
Added ID View "example_for_host1"
-----
ID View Name: example_for_host1
```

2. ユーザー上書きを **example_for_host1** ID ビューに追加します。**ipa idoverrideuser-add** コマンドでは、ID ビューの名前と上書きするユーザーが必要になります。

- 新規の属性値を指定するには、対応するコマンドラインオプションも追加します。利用可能なオプション一覧については、**ipa idoverrideuser-add --help** を実行すると確認できます。たとえば、SSH 公開キーの値を上書きするには、**--sshpubkey** オプションを使用します。

```
$ ipa idoverrideuser-add example_for_host1 user --sshpubkey="ssh-
rsa AAAAB3NzaC1yrRqFE...gWRL71/miPIZ user@example.com"
-----
Added User ID override "user"
-----
Anchor to override: user
SSH public key: ssh-rsa
                  AAAAB3NzaC1yrRqFE...gWRL71/miPIZ
                  user@example.com
```



注記

IdM へのSSH キーの追加に関する詳細は、[「ユーザーの公開 SSH キーの管理」](#) を参照してください。

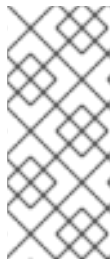
- ipa idoverrideuser-add --certificate** コマンドは、指定した ID ビュー内のアカウントの既存証明書すべてを置き換えます。新たな証明書を追加するには、代わりに **ipa idoverrideuser-add-cert** コマンドを使用します。

```
$ ipa idoverrideuser-add-cert example_for_host1 user --
certificate="MIIEATCC..."
```

- ipa idoverrideuser-mod** コマンドを使用すると、既存のユーザー上書きに新たな属性値を指定することもできます。

3. **example_for_host1** を **host1.example.com** ホストに適用します。

```
$ ipa idview-apply example_for_host1 --hosts=host1.example.com
-----
Applied ID View "example_for_host1"
-----
hosts: host1.example.com
-----
Number of hosts the ID View was applied to: 1
-----
```



注記

ipa idview-apply コマンドは、**--hostgroups** オプションも受け取ります。このオプションは、指定されたホストグループに属するホストに ID ビューを適用しますが、ID ビューをホストグループ自体に関連付けることはしません。代わりに **--hostgroups** オプションは指定されたホストグループのメンバーを拡張し、**--hosts** オプションを個別にメンバーに対して適用します。

第19章 IDM ユーザーのアクセス制御の定義

アクセス制御は、誰がマシンやサービスまたはエントリーなどの特定のリソースにアクセスできるかや、どのような種類の操作の実行を許可されるかななどを定義する、セキュリティー機能セットです。Identity Management はいくつかのアクセス制御エリアを提供し、どのような種類のアクセスが許可されているか、誰に許可されているか、を明確にします。この一部として Identity Management は、ドメイン内のリソースに対するアクセス制御と、IdM 設定自体へのアクセス制御を区別します。

本章では、IdM サーバーおよび他の IdM ユーザーに対する IdM 内のユーザーに利用可能な異なる内部アクセス制御メカニズムに関する詳細は、[10章IdM ユーザーのアクセス制御の定義](#)を参照してください。

第20章 KERBEROS フラグとプリンシパルエイリアスの管理

20.1. サービスおよびホスト向けの KERBEROS フラグ

Kerberos チケット動作の特定の側面を定義するには、各種の Kerberos フラグを使うことができます。これらのフラグは、サービスとホスト Kerberos プリンシパルに追加することができます。

IdM のプリンシパルは、以下の 2 つの Kerberos フラグを受け付けます。

OK_AS_DELEGATE

委任用に信頼する Kerberos チケットを指定するには、このフラグを使用します。

AD クライアントは、Kerberos チケット上の **OK_AS_DELEGATE** フラグをチェックして、ユーザー認証情報を特別のサーバーに転送または委任できるかどうかを判断します。AD は **OK_AS_DELEGATE** セットのあるサービスまたはホストにのみ、TGT を転送します。このフラグがあると、SSSD は AD ユーザー TGT を IdM クライアントマシン上のデフォルトの Kerberos 認証情報キャッシュに追加することができます。

REQUIRES_PRE_AUTH

事前に認証されたチケットのみがプリンシパルに認証可能と指定するには、このフラグを使用します。

REQUIRES_PRE_AUTH フラグが設定されている場合は、KDC は新たな認証を必要します。TGT が事前に認証されている場合にのみ、KDC は **REQUIRES_PRE_AUTH** のあるプリンシパル用の TGT を発行します。

REQUIRES_PRE_AUTH を使うと、選択したサービスまたはホストの事前認証を無効にすることができます。こうすると KDC の負荷は下がりますが、長期のキーへのブルートフォース攻撃が成功する確率が少し高くなります。

20.1.1. Web UI で Kerberos フラグを設定する

IdM web UI からは現在、プリンシパルに **OK_AS_DELEGATE** フラグのみを追加できます。

1. **Identity** メインタブから **Services** サブタブを選択します。

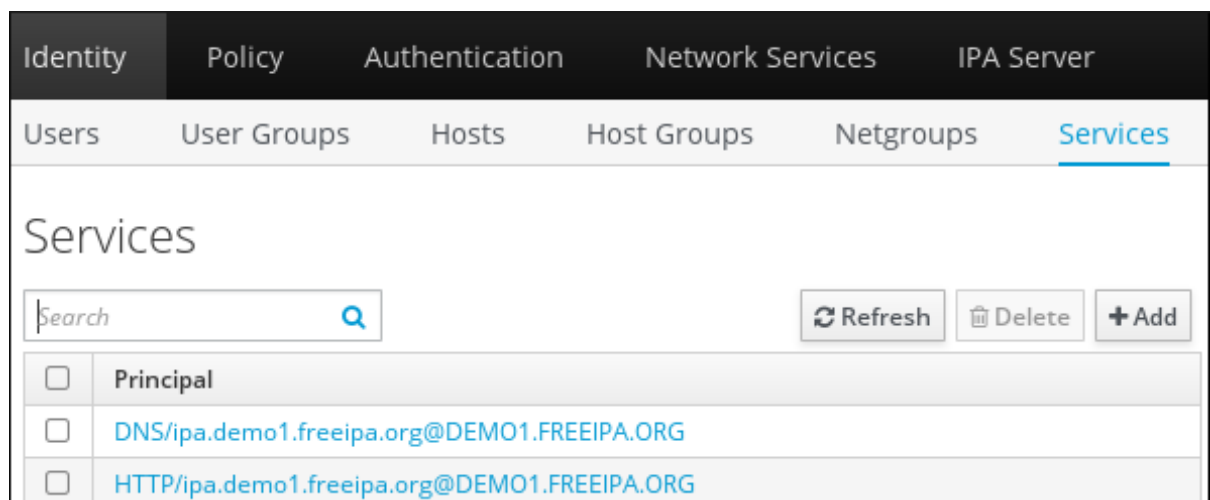


図20.1 サービス一覧

2. フラグを追加するサービスをクリックします。
3. **Trusted for delegation** オプションにチェックを入れます。

Service: DNS/ipa.demo1.freeipa.org@DEMO1.FREEIPA.ORG

DNS/ipa.demo1.fre... is a member of: DNS/ipa.demo1.fre... is managed by:

Settings Roles Hosts (1)

Refresh Reset Update Actions

Service Settings

Principal: DNS/ipa.demo1.freeipa.org@DEMO1.FREEIPA.ORG

Service: DNS

Host Name: ipa.demo1.freeipa.org

PAC type:

- ☒ Inherited from server configuration
- ☐ Override inherited settings
 - ☐ MS-PAC
 - ☐ PAD

Trusted for delegation ☒ Undo

図20.2 OK_AS_DELEGATE フラグの追加

20.1.2. コマンドライン で **Kerberos** フラグを設定する

コマンドラインからプリンシパルにグラフを追加または削除するには、**ipa service-mod** コマンドに以下のいずれかのオプションを追加します。

- **OK_AS_DELEGATE** には **--ok-as-delegate** を追加
- **REQUIRES_PRE_AUTH** には **--requires-pre-auth** を追加

フラグを追加するには、対応するオプションを **1** に設定します。たとえば、**OK_AS_DELEGATE** フラグを **test/ipa.example.com@EXAMPLE.COM** プリンシパルに追加するには、以下を実行します。

```
$ ipa service-mod test/ipa.example.com@EXAMPLE.COM --ok-as-delegate=1
```

フラグを削除または無効にするには、対応するオプションを **0** に設定します。たとえば、**REQUIRES_PRE_AUTH** フラグを **test/ipa.example.com@EXAMPLE.COM** プリンシパルで無効にするには、以下を実行します。

```
$ ipa service-mod test/ipa.example.com@EXAMPLE.COM --requires-pre-auth=0
```

プリンシパルに **OK_AS_DELEGATE** が設定されているかどうかを確認するには、**kvno** ユーティリティを実行してから、**klist -f** コマンドを実行します。**OK_AS_DELEGATE** は、**klist -f** の出力の **0** の文字で表されます。

```
$ kvno test/ipa.example.com@EXAMPLE.COM
$ klist -f
Ticket cache: KEYRING:persistent:0:0
Default principal: admin@EXAMPLE.COM

Valid starting Expires Service principal
02/19/2014 09:59:02 02/20/2014 08:21:33 test/ipa/example.com@EXAMPLE.COM
Flags: FATO
```

プリンシパルにどのフラグが設定されているかを調べるには、**kadmin.local** ユーティリティーを使用します。現行フラグは、以下のように **kadmin.local** の出力の **Attributes** の行に表示されます。

```
# kadmin.local
kadmin.local: getprinc test/ipa.example.com
Principal: test/ipa.example.com@EXAMPLE.COM
Expiration date: [never]
Last password change: Mon Sep 16 15:44:21 EDT 2013
Password expiration date: [none]
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 7 days 00:00:00
Last modified: Mon Oct 14 23:42:53 EDT 2013 (admin/admin@EXAMPLE.COM)
Last successful authentication: Wed Mar 11 08:01:03 EDT 2015
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 6
Key: vno 1, aes256-cts-hmac-sha1-96, no salt
Key: vno 1, aes128-cts-hmac-sha1-96, no salt
Key: vno 1, des3-cbc-sha1, no salt
Key: vno 1, arcfour-hmac, no salt
Key: vno 1, camellia128-cts-cmac, no salt
Key: vno 1, camellia256-cts-cmac, no salt
MKey: vno 1
Attributes: REQUIRES_PRE_AUTH OK_AS_DELEGATE OK_TO_AUTH_AS_DELEGATE
Policy: [none]
```

20.2. ユーザー、ホスト、およびサービス向け **KERBEROS** プリンシパルエイリアスの管理

新規のユーザー、ホスト、またはサービスを作成すると、以下のフォーマットで Kerberos プリンシパルが自動的に追加されます。

- `user_name@REALM`
- `host/host_name@REALM`
- `service_name/host_name@REALM`

場合によっては、ユーザー、ホスト、またはサービスがエイリアスを使って Kerberos アプリケーションに認証できるようにすると管理者に有益なこともあります。たとえば、以下のような場合です。

- ユーザー名を変更したものの、ユーザーが以前のユーザー名と新たなユーザー名の両方を使ってログインする場合。
- IdM Kerberos レalmが E メールドメインと異なる場合でも、ユーザーが E メールアドレスを使用してログインする必要がある場合。

ユーザー名を変更した場合は、オブジェクトがエイリアスと以前の正規のプリンシパル名を保持することに注意してください。

20.2.1. Kerberos プリンシパルのエイリアス

Kerberos プリンシパルエイリアスの追加

エイリアス名 **useralias** をアカウント **user** に追加するには、以下を入力します。

```
[root@ipaserver ~]# ipa user-add-principal user useralias
-----
Added new aliases to user "user"
-----
      User login: user
      Principal alias: user@IDM.EXAMPLE.COM, useralias@IDM.EXAMPLE.COM
```

ホストまたはサービスにエイリアスを追加するには、代わりにそれぞれ **ipa host-add-principal** または **ipa service-add-principal** コマンドを使用します。

認証にエイリアス名を使用する場合は、**-C** オプションを **kinit** コマンドに渡します。

```
[root@ipaserver ~]# kinit -C useralias
Password for user@IDM.EXAMPLE.COM:
```

Kerberos プリンシパルエイリアスの削除

エイリアス **useralias** をアカウント **user** から削除するには、以下を入力します。

```
[root@ipaserver ~]# ipa user-remove-principal user useralias
-----
Removed aliases from user "user"
-----
      User login: user
      Principal alias: user@IDM.EXAMPLE.COM
```

ホストまたはサービスからエイリアスを削除するには、代わりにそれぞれ **ipa host-remove-principal** または **ipa service-remove-principal** コマンドを使用します。

正規のプリンシパル名は削除できないことに注意してください。

```
[root@ipaserver ~]# ipa user-show user
      User login: user
      ...
      Principal name: user@IDM.EXAMPLE.COM
      ...

[root@ipaserver ~]# ipa user-remove-principal user user
ipa: ERROR: invalid 'krbprincipalname': at least one value equal to the
canonical principal name must be present
```

20.2.2. Kerberos エンタープライズプリンシパルのエイリアス

エンタープライズプリンシパルエイリアスには、ユーザープリンシパル名 (UPN) サフィックス、NetBIOS 名、または信頼される Active Directory フォレストドメインのドメイン名を除いて、いかなるドメインサフィックスを使用することができます。



注記

エンタープライズプリンシパルエイリアスを追加または削除する際には、2つのバックスラッシュ (\\) を使って @ 記号をエスケープします。これを使用しないとシェルは @ を Kerberos レalm名の一部として解釈し、以下のエラーが表示されます。

```
ipa: ERROR: The realm for the principal does not match the realm
for this IPA server
```

Kerberos エンタープライズプリンシパルのエイリアスの追加

エンタープライズプリンシパルエイリアス **user@example.com** を **user** アカウントに追加するには、以下を実行します。

```
[root@ipaserver ~]# ipa user-add-principal user user\\@example.com
-----
Added new aliases to user "user"
-----
      User login: user
      Principal alias: user@IDM.EXAMPLE.COM,
user\\example.com@IDM.EXAMPLE.COM
```

ホストまたはサービスにエンタープライズエイリアスを追加するには、代わりにそれぞれ **ipa host-add-principal** または **ipa service-add-principal** コマンドを使用します。

認証にエンタープライズプリンシパル名を使用する場合は、**-E** オプションを **kinit** コマンドに渡します。

```
[root@ipaserver ~]# kinit -E user@example.com
Password for user\\example.com@IDM.EXAMPLE.COM:
```

Kerberos エンタープライズプリンシパルエイリアスの削除

エンタープライズプリンシパルエイリアス **user@example.com** をアカウント **user** から削除するには、以下を入力します。

```
[root@ipaserver ~]# ipa user-remove-principal user user\\@example.com
-----
Removed aliases from user "user"
-----
      User login: user
      Principal alias: user@IDM.EXAMPLE.COM
```

ホストまたはサービスからエイリアスを削除するには、代わりにそれぞれ **ipa host-remove-principal** または **ipa service-remove-principal** コマンドを使用します。

第21章 NIS ドメインおよびネットグループとの統合

21.1. NIS と IDENTITY MANAGEMENT

UNIX 環境では、ネットワーク情報サービス (NIS) が ID と認証を集中管理する一般的な方法です。NIS は **Yellow Pages** (YP) と呼ばれていたこともあり、以下のような認証と ID の情報を集中的に管理します。

- ユーザーとパスワード
- ホスト名および IP アドレス
- POSIX グループ

現代のネットワークでは、NIS はホスト認証を提供せず、ネットワーク上でデータを暗号化せずに送信するため、安全とはみなされていません。この問題を回避するために、NIS は他のプロトコルと統合してセキュリティを強化することがよくあります。

Identity Management (IdM) を利用すると、NIS サーバープラグインを使って、IdM に完全に移行できないクライアントに接続することができます。IdM は、netgroup と他の NIS データを IdM ドメインに統合します。また、ユーザーおよびホストの ID を NIS ドメインから IdM に移行することも容易になります。

Identity Management での NIS

NIS オブジェクトは、[RFC 2307](#) に従って、Directory Server のバックエンドに統合、保存されます。IdM は NIS オブジェクトを LDAP ディレクトリーに作成し、クライアントは暗号化された LDAP 接続を使用して、これらをたとえば System Security Services Daemon (SSSD) や `nss_ldap` から取得します。

IdM は、netgroup、アカウント、グループ、ホスト、および他のデータを管理します。IdM は NIS リスナーを使ってパスワード、グループ、および netgroup を IdM エントリーにマッピングします。

Identity Management での NIS プラグイン

NIS のサポートに IdM は以下のプラグインを使用します。これらは、`slapi-nis` パッケージで提供されます。

NIS サーバープラグイン

NIS サーバープラグインを使用すると、IdM 統合の LDAP サーバーはクライアント用の NIS サーバーとして機能することができます。この場合、設定に従って Directory Server が動的に NIS マップを生成し、更新します。このプラグインを使用することで、IdM は、NIS プロトコルを NIS サーバーとして使用するクライアントの要求に応えます。

詳細情報は、「[Identity Management で NIS を有効にする](#)」を参照してください。

スキーマ互換性プラグイン

このプラグインを使用すると、Directory Server バックエンドがディレクトリー情報ツリー (DIT) の一部の保存されているエントリーの別のビューを提供できるようになります。これには、属性値の追加、削除、名前変更や、ツリーの複数のエントリーから属性の値を取得することなどが含まれます。

詳細については、`/usr/share/doc/slapi-nis-version/sch-getting-started.txt` ファイルを参照してください。

21.1.1. Identity Management での NIS Netgroup

NIS エンティティーは netgroup に保存出来ます。UNIX グループと比べると、netgroup は以下をサポートします。

- 入れ子グループ (他のグループのメンバーとしてのグループ)
- ホストのグループ化

netgroup では、ホスト、ユーザー、およびドメインが定義されます。これら情報のセットは **triple** と呼ばれ、これらのフィールドには以下が含まれます。

- 値
- ダッシュ (-)。これは、有効な値がないことを意味します。
- 値なし。フィールドが空の場合は、ワイルドカードを意味します。

```
(host.example.com,,nisdomain.example.com)
(-,user,nisdomain.example.com)
```

クライアントが NIS netgroup をリクエストすると、IdM は LDAP エントリーを以下に変換します。

- 従来の NIS マップ。NIS プラグインを使用して NIS プロトコルによりクライアントにこれを送信します。
- LDAP 形式。これは [RFC 2307](#) または RFC 2307bis 準拠になります。

21.1.1.1. NIS Netgroup エントリーの表示

IdM はユーザーとグループを **memberUser** 属性に、ホストとホストグループを **memberHost** に保存します。以下の例では、IdM の Directory Server コンポーネントにある netgroup エントリーを表示しています。

例21.1 Directory Server 内の NIS エントリー

```
dn: ipaUniqueID=d4453480-cc53-11dd-ad8b-0800200c9a66,cn=ng,cn=alt,...
...
cn: netgroup1
memberHost: fqdn=host1.example.com,cn=computers,cn=accounts,...
memberHost: cn=VirtGuests,cn=hostgroups,cn=accounts,...
memberUser: cn=demo,cn=users,cn=accounts,...
memberUser: cn=Engineering,cn=groups,cn=accounts,...
nisDomainName: nisdomain.example.com
```

IdM では、**ipa netgroup-*** コマンドを使って netgroup エントリーを管理します。たとえば、netgroup エントリーを表示するには、以下を実行します。

例21.2 Netgroup エントリーの表示

```
[root@server ~]# ipa netgroup-show netgroup1
Netgroup name: netgroup1
Description: my netgroup
NIS domain name: nisdomain.example.com
```

```
Member Host: VirtGuests
Member Host: host1.example.com
Member User: demo
Member User: Engineering
```

21.2. IDENTITY MANAGEMENT で NIS を有効にする

Identity Management で NIS を有効にするには、以下の手順に従います。

1. NIS リスナーと互換性プラグインを有効にします。

```
[root@ipaserver ~]# ipa-nis-manage enable
[root@ipaserver ~]# ipa-compat-manage enable
```

2. オプションで、NIS リモートプロシージャークール (RPC) 用に固定ポートを設定します。

NIS の使用時には、クライアントは IdM サーバー上のどのポートを使って接続を確立するかを知る必要があります。IdM はデフォルト設定を使用して、サーバーの起動時に未使用のランダムポートにバインドします。このポートは、クライアントがポート番号のリクエストに使用するポートマッパーサービスに送信されます。

ファイアウォール設定を厳密にする場合には、固定ポートを設定出来ます。たとえば、ポートを **514** に設定するには、以下を実行します。

```
[root@ipaserver ~]# ldapmodify -x -D 'cn=directory manager' -W
dn: cn=NIS Server,cn=plugins,cn=config
changetype: modify
add: nsslapd-pluginarg0
nsslapd-pluginarg0: 514
```



注記

この設定には、1024 より下のポート番号で未使用のものを使用できます。

3. ポートマッパーサービスを有効にして、起動します。

```
[root@ipaserver ~]# systemctl enable rpcbind.service
[root@ipaserver ~]# systemctl start rpcbind.service
```

4. Directory Server を再起動します。

```
[root@ipaserver ~]# systemctl restart dirsrv.target
```

21.3. NETGROUPS の作成

21.3.1. Netgroup の追加

Netgroup の追加には、以下のいずれかを使用します。

- IdM Web UI (「[Web UI: Netgroup の追加](#)」を参照)

- コマンドライン (「[コマンドライン: Netgroup の追加](#)」を参照)

Web UI: Netgroup の追加

1. **Identity** → **Groups** → **Netgroups** を選択します。
2. **Add** をクリックします。
3. 一意の名前と、オプションで説明を入力します。グループ名は、IdM ドメインの netgroup で使用される識別子です。これは後で変更できません。
4. **Add and Edit** をクリックして変更を保存し、エントリーの編集を開始します。
5. デフォルトの NIS ドメインが IdM ドメイン名に設定されます。オプションで、別の NIS ドメイン名を NIS ドメイン名フィールドに入力することもできます。

The screenshot shows the 'Netgroup: server.example.com' configuration page. At the top, it says 'server.example.com members: server.example.com is a member of:'. Below this are three tabs: 'Settings', 'Netgroups', and 'Netgroups' (the last one is highlighted with a green border). Under the tabs are three buttons: 'Refresh', 'Revert', and 'Save'. The main section is titled 'General' and contains three fields: 'Netgroup name' with the value 'server.example.com', 'Description' with the value 'An example' (in a text area), and 'NIS domain name' with the value 'example.com'. Each field has an 'Undo' button next to it.

図21.1 Netgroup タブ

NIS domain name フィールドでは、netgroup の triple に表示されるドメインを設定します。これは、Identity Management リスナーが応答する NIS ドメインに影響するものではありません。

6. 「[Web UI: Netgroup にメンバーを追加する](#)」にあるように、メンバーを追加します。
7. **保存** をクリックします。

コマンドライン: Netgroup の追加

新規 netgroup は、**ipa netgroup-add** コマンドを使って追加します。以下を指定します。

- グループ名
- オプションで説明
- NIS ドメイン名が IdM ドメイン名とは別の場合、これをオプションで追加



注記

--nisdomain オプションでは、netgroup の triple に表示されるドメインを設定します。これは、Identity Management リスナーが応答する NIS ドメインに影響するものではありません。

例を示します。

```
[root@server ~]# ipa netgroup-add --desc="Netgroup description" --nisdomain="example.com" example-netgroup
```

netgroup にメンバーを追加する方法については、[「コマンドライン: Netgroup にメンバーを追加する」](#)を参照してください。

21.3.2. Netgroup にメンバーを追加する

ユーザーとホスト以外に、netgroup にはユーザーグループ、ホストグループ、および他の netgroup (入れ子グループ) をメンバーとして含めることができます。グループのサイズによっては、子グループのメンバー用に入れ子グループを作成してからそれが親グループのメンバーとして表示されるまで数分かかる場合があります。

Netgroup にメンバーを追加するには、以下のいずれかを使用します。

- IdM web UI ([「Web UI: Netgroup にメンバーを追加する」](#)を参照)
- コマンドライン ([「コマンドライン: Netgroup にメンバーを追加する」](#)を参照)



警告

入れ子グループを作成する際は、再帰グループを作成しないでください。たとえば、*GroupA* が *GroupB* のメンバーの場合、*GroupB* を *GroupA* のメンバーとして追加しないでください。再帰グループはサポートされておらず、予測不可能な動作を引き起こす可能性があります。

Web UI: Netgroup にメンバーを追加する

Web UI で netgroup にメンバーを追加するには、以下の手順に従います。

1. **Identity** → **Groups** → **Netgroups** を選択します。
2. メンバーを追加する netgroup 名をクリックします。

3. メンバータイプ横にある **Add** をクリックします。

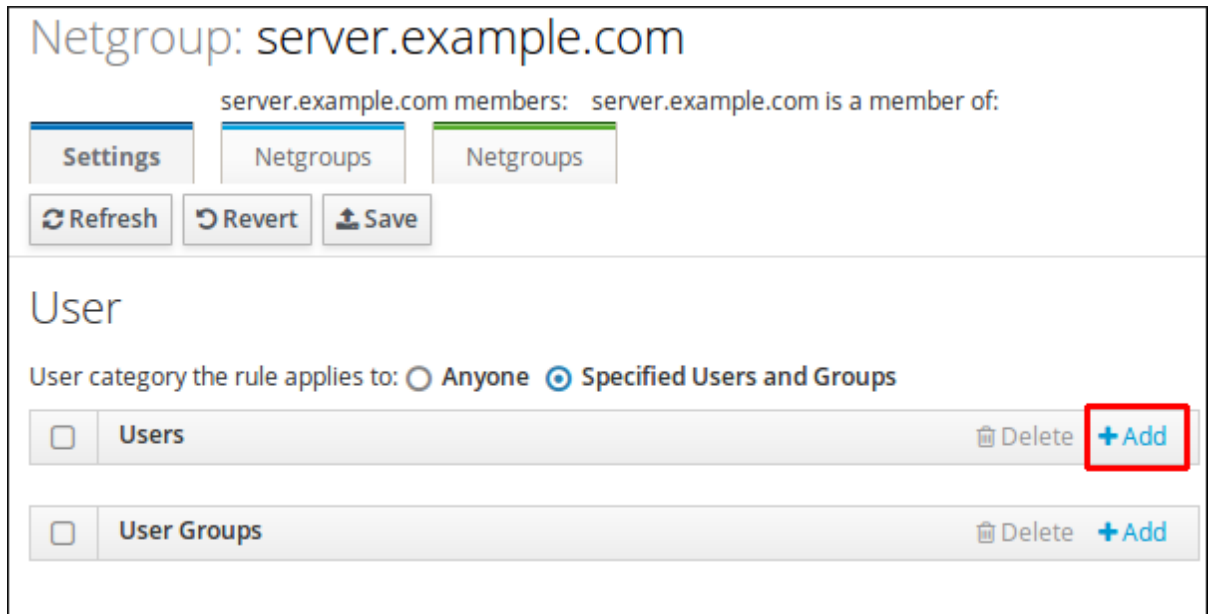


図21.2 Netgroup タブのユーザーメニュー

4. 追加するメンバーを選択し、> をクリックして確定します。

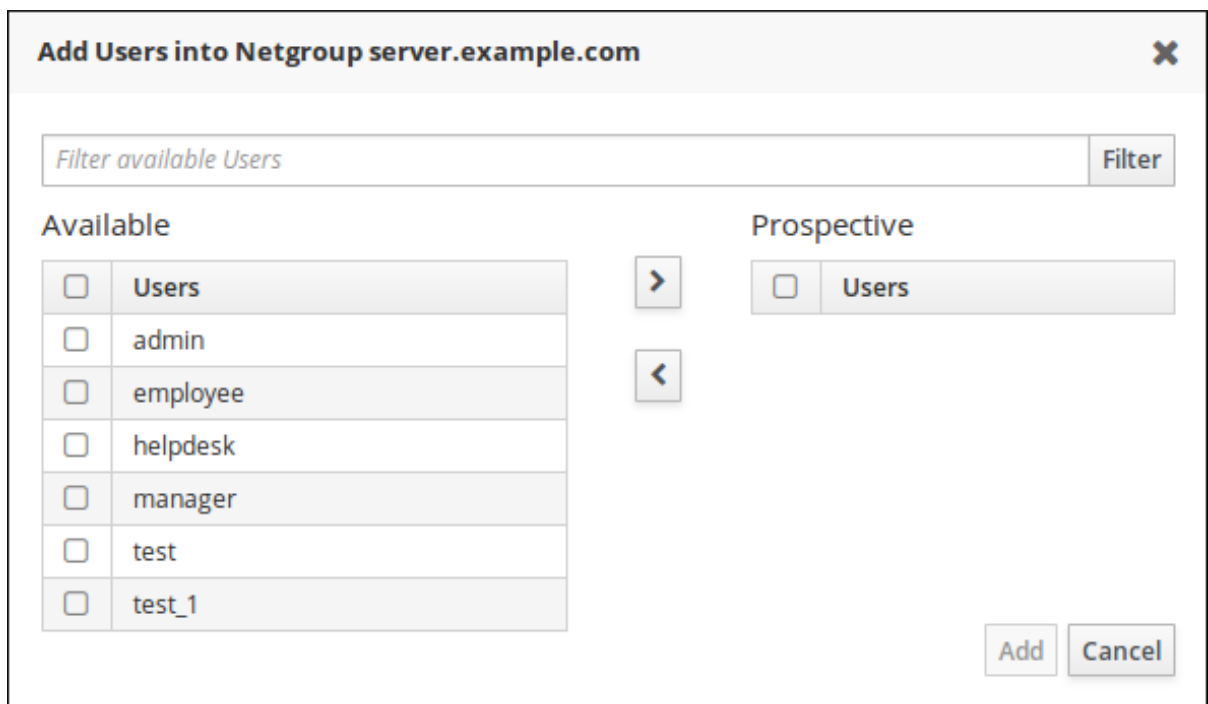


図21.3 Netgroup タブでのユーザー追加メニュー

5. **Add** をクリックします。

コマンドライン: Netgroup にメンバーを追加する

netgroup を作成したら、**ipa netgroup-add-member** コマンドでメンバーを追加します。

```
# ipa netgroup-add-member --users=user_name --groups=group_name --
hosts=host_name \
    --hostgroups=host_group_name --netgroups=netgroup_name group_nameame
```

複数のメンバーを追加するには、以下のように中括弧内にコンマ区切りリストで記載します。

```
[root@server ~]# ipa netgroup-add-member --users={user1;user2,user3} \
--groups={group1,group2} example-group
```

21.4. 自動マウントマップの NIS クライアントへの公開

自動マウントマップが既に定義されている場合、マップを IdM の NIS 設定に手動で追加する必要があります。こうすることで、マップが NIS クライアントに確実に公開されます。

NIS サーバーは、IdM LDAP ディレクトリー内の特別なプラグインエントリーで管理されます。NIS サーバーが使用する各 NIS ドメインおよびマップは、このコンテナでサブエントリーとして追加されます。NIS ドメインエントリーには、以下のものが含まれます。

- NIS ドメイン名
- NIS マップ名
- NIS マップのコンテンツとして使用するためのディレクトリーエントリーの発見方法
- NIS マップのキーおよび値としてどの属性を使用するかについての情報

これら設定のほとんどは、各マップで同じものになります。

21.4.1. 自動マウントマップの追加

IdM は、自動マウントの場所ごとにグループ化された自動マウントマップを IdM ディレクトリーツリーの **cn=automount** ブランチに保存します。NIS ドメインとマップは LDAP プロトコルを使って追加できます。

たとえば、**example.com** ドメイン内の **default** の場所にある **auto.example** という自動マウントマップを追加するには、以下を実行します。

```
[root@server ~]# ldapadd -h server.example.com -x -D "cn=Directory
Manager" -W

dn: nis-domain=example.com+nis-map=auto.example,cn=NIS
Server,cn=plugins,cn=config
objectClass: extensibleObject
nis-domain: example.com
nis-map: auto.example
nis-filter: (objectclass=automount)
nis-key-format: %{automountKey}
nis-value-format: %{automountInformation}
nis-base:
automountmapname=auto.example,cn=default,cn=automount,dc=example,dc=com
```



注記

nis-domain 属性は、自分の NIS ドメイン名に設定します。

nis-base 属性で設定する値は、以下のものに対応する必要があります。

- ***ipa automountmap-**** コマンドを使用して設定した既存の自動マウントマップ
- ***ipa automountlocation-**** コマンドを使用して設定した既存の自動マウントの場所

エントリーを設定したら、以下を実行して自動マウントマップを確認します。

```
[root@server ~]# ypcat -k -d example.com -h server.example.com  
auto.example
```

21.5. NIS から IDM への移行

既存の NIS サーバーから Identity Management (IdM) に移行するには、以下のステップを実行します。

1. [Identity Management で NIS リスナーを有効にする](#)
2. [既存のデータを NIS からエクスポート、インポートする](#)

21.5.1. IdM でのネットグループエントリーの準備

移行前に、現行の NIS サーバーで管理されている ID を確認します。

ユーザーエントリー

NIS が提供しているユーザー情報を使用しているアプリケーションを判定します。**sudo** などのユーティリティは NIS netgroup を必要としますが、通常の UNIX グループを使用できるものもあります。

移行は、以下の手順で実行します。

1. 対応するユーザーアカウントを IdM に作成します。[「ユーザーエントリーの移行」](#) を参照してください。
2. さらに netgroup が必要な場合は、以下を実行します。
 - a. netgroup を追加します。[「Netgroup の追加」](#) を参照してください。
 - b. ユーザーを netgroup に追加します。[「Netgroup エントリーの移行」](#) を参照してください。

ホストエントリー

IdM でホストグループを作成すると、対応するシャドウ NIS グループが自動的に作成されます。これらのネットグループは、***ipa netgroup-**** コマンドを使って管理します。

直接変換の場合

ユーザーおよびホストエントリーですべて同じ名前を使用する必要がある場合は、IdM 内の同一名を使用してエントリーを作成します。

1. netgroup で参照されているユーザーすべてについてエントリーを作成します。
2. netgroup で参照されているホストすべてについてエントリーを作成します。
3. 元の netgroup と同じ名前の netgroup を作成します。
4. ユーザーとホストをこの netgroup の直接のメンバーとして追加します。ユーザーおよびホストがグループまたはホストグループのメンバーである場合は、これらのグループを netgroup に追加します。

21.5.2. Identity Management で NIS リスナーを有効にする

「[Identity Management で NIS を有効にする](#)」を参照してください。

21.5.3. 既存 NIS データのインポートおよびエクスポート

NIS サーバーには、ユーザー、グループ、ホスト、netgroup、および自動マウントマップの情報が含まれます。これらのエントリータイプはいずれも IdM に移行することができます。

以下のセクションでは、**ypcat** コマンドを使ってデータをNIS サーバーからエクスポートし、その出力を使用して対応する **ipa *-add** コマンドでエントリーを IdM にインポートします。

21.5.3.1. ユーザーエントリーの移行

NIS **passwd** マップには、UID、プライマリーグループ、GECOS、シェル、およびホームディレクトリーなどのユーザー情報が含まれます。このデータを使用して NIS ユーザーアカウントを IdM に移行します。

1. オプションで、弱いパスワードのサポートが必要な場合は、「[NIS ユーザー認証用に脆弱なパスワードハッシュを有効にする](#)」を参照してください。
2. 以下のコンテンツで **/root/nis-users.sh** スクリプトを作成します。

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -d $1 -h $2 passwd > /dev/shm/nis-map.passwd 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.passwd) ; do
    IFS=' '
    username=$(echo $line | cut -f1 -d:)
    # Not collecting encrypted password because we need cleartext
    password
    # to create kerberos key
    uid=$(echo $line | cut -f3 -d:)
    gid=$(echo $line | cut -f4 -d:)
    gecos=$(echo $line | cut -f5 -d:)
    homedir=$(echo $line | cut -f6 -d:)
    shell=$(echo $line | cut -f7 -d:)

    # Now create this entry
    echo passwd0rd1 | ipa user-add $username --first=NIS --last=USER \
```

```
--password --gidnumber=$gid --uid=$uid --gecos=$gecos --
homedir=$homedir \
--shell=$shell
ipa user-show $username
done
```

3. IdM **admin** ユーザーとして認証します。

```
[root@nis-server ~]# kinit admin
```

4. 以下のようにスクリプトを実行します。

```
[root@nis-server ~]# sh /root/nis-users.sh nisdomain nis-
master.example.com
```



注記

このスクリプトはハードコーディングされた値を姓、名に使用し、パスワードを **passw0rd1** に設定します。ユーザーは次回ログイン時にこのパスワードを変更する必要があります。

21.5.3.2. グループエントリーの移行

NIS **group** マップには、グループ名、GID、グループメンバーなどのグループ情報が含まれます。このデータを使用して NIS グループを IdM に移行します。

1. 以下のコンテンツで **/root/nis-groups.sh** スクリプトを作成します。

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -d $1 -h $2 group > /dev/shm/nis-map.group 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.group); do
    IFS=' '
    groupname=$(echo $line | cut -f1 -d:)
    # Not collecting encrypted password because we need cleartext
    password
    # to create kerberos key
    gid=$(echo $line | cut -f3 -d:)
    members=$(echo $line | cut -f4 -d:)

    # Now create this entry
    ipa group-add $groupname --desc=NIS_GROUP_$groupname --gid=$gid
    if [ -n "$members" ]; then
        ipa group-add-member $groupname --users={$members}
    fi
    ipa group-show $groupname
done
```

2. IdM **admin** ユーザーとして認証します。

```
[root@nis-server ~]# kinit admin
```

3. 以下のようにスクリプトを実行します。

```
[root@nis-server ~]# sh /root/nis-groups.sh nisdomain nis-master.example.com
```

21.5.3.3. ホストエントリーの移行

NIS **hosts** マップには、ホスト名や IP アドレスなどのホスト情報が含まれます。このデータを使用して NIS ホストエントリーを IdM に移行します。

1. 以下のコンテンツで **/root/nis-hosts.sh** スクリプトを作成します。

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -d $1 -h $2 hosts | egrep -v "localhost|127.0.0.1" >
/dev/shm/nis-map.hosts 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.hosts); do
    IFS=' '
    ipaddress=$(echo $line | awk '{print $1}')
    hostname=$(echo $line | awk '{print $2}')
    master=$(ipa env xmlrpc_uri | tr -d '[:space:]' | cut -f3 -d: | cut
-f3 -d/)
    domain=$(ipa env domain | tr -d '[:space:]' | cut -f2 -d:)
    if [ $(echo $hostname | grep "\." | wc -l) -eq 0 ] ; then
        hostname=$(echo $hostname.$domain)
    fi
    zone=$(echo $hostname | cut -f2- -d.)
    if [ $(ipa dnszone-show $zone 2>/dev/null | wc -l) -eq 0 ] ; then
        ipa dnszone-add --name-server=$master --admin-email=root.$master
    fi
    ptrzone=$(echo $ipaddress | awk -F. '{print $3 "." $2 "." $1 ".in-
addr.arpa."}')
    if [ $(ipa dnszone-show $ptrzone 2>/dev/null | wc -l) -eq 0 ] ;
then
        ipa dnszone-add $ptrzone --name-server=$master --admin-
email=root.$master
    fi
    # Now create this entry
    ipa host-add $hostname --ip-address=$ipaddress
    ipa host-show $hostname
done
```

2. IdM **admin** ユーザーとして認証します。

```
[root@nis-server ~]# kinit admin
```

3. 以下のようにスクリプトを実行します。

```
[root@nis-server ~]# sh /root/nis-hosts.sh nisdomain nis-master.example.com
```



注記

このスクリプトでは、エイリアスなどの特定なホスト設定は移行されません。

21.5.3.4. Netgroup エントリーの移行

NIS **netgroup** マップには、netgroup についての情報が含まれます。このデータを使用して NIS netgroup を IdM に移行します。

1. 以下のコンテンツで **/root/nis-netgroups.sh** スクリプトを作成します。

```
#!/bin/sh
# $1 is the NIS domain, $2 is the NIS master server
ypcat -k -d $1 -h $2 netgroup > /dev/shm/nis-map.netgroup 2>&1

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.netgroup); do
    IFS=' '
    netgroupname=$(echo $line | awk '{print $1}')
    triples=$(echo $line | sed "s/^$netgroupname //")
    echo "ipa netgroup-add $netgroupname --desc=NIS_NG_$netgroupname"
    if [ $(echo $line | grep "(," | wc -l) -gt 0 ]; then
        echo "ipa netgroup-mod $netgroupname --hostcat=all"
    fi
    if [ $(echo $line | grep ",," | wc -l) -gt 0 ]; then
        echo "ipa netgroup-mod $netgroupname --usercat=all"
    fi

    for triple in $triples; do
        triple=$(echo $triple | sed -e 's/-//g' -e 's/(//' -e 's/)//')
        if [ $(echo $triple | grep ",.*," | wc -l) -gt 0 ]; then
            hostname=$(echo $triple | cut -f1 -d,)
            username=$(echo $triple | cut -f2 -d,)
            domain=$(echo $triple | cut -f3 -d,)
            hosts=""; users=""; doms="";
            [ -n "$hostname" ] && hosts="--hosts=$hostname"
            [ -n "$username" ] && users="--users=$username"
            [ -n "$domain" ] && doms="--nisdomain=$domain"
            echo "ipa netgroup-add-member $hosts $users $doms"
        else
            netgroup=$triple
            echo "ipa netgroup-add $netgroup --desc=NIS_NG_$netgroup"
        fi
    done
done
```

2. IdM **admin** ユーザーとして認証します。

```
[root@nis-server ~]# kinit admin
```

3. 以下のようにスクリプトを実行します。

```
[root@nis-server ~]# sh /root/nis-netgroups.sh nisdomain nis-
master.example.com
```

21.5.3.5. 自動マウントマップの移行

自動マウントマップは、場所 (親エントリー) と関連のキーおよびマップを定義する入れ子および相互関連のエントリーになります。以下の手順で NIS 自動マウントマップを IdM に移行します。

1. 以下のコンテンツで `/root/nis-automounts.sh` スクリプトを作成します。

```
#!/bin/sh
# $1 is for the automount entry in ipa

ipa automountlocation-add $1

# $2 is the NIS domain, $3 is the NIS master server, $4 is the map
name
ypcat -k -d $2 -h $3 $4 > /dev/shm/nis-map.$4 2>&1

ipa automountmap-add $1 $4

basedn=$(ipa env basedn | tr -d '[:space:]' | cut -f2 -d:)
cat > /tmp/amap.ldif <<EOF
dn: nis-domain=$2+nis-map=$4,cn=NIS Server,cn=plugins,cn=config
objectClass: extensibleObject
nis-domain: $2
nis-map: $4
nis-base: automountmapname=$4,cn=$1,cn=automount,$basedn
nis-filter: (objectclass=*)
nis-key-format: %{automountKey}
nis-value-format: %{automountInformation}
EOF
ldapadd -x -h $3 -D "cn=Directory Manager" -W -f /tmp/amap.ldif

IFS=$'\n'
for line in $(cat /dev/shm/nis-map.$4); do
    IFS=" "
    key=$(echo "$line" | awk '{print $1}')
    info=$(echo "$line" | sed -e "s#^$key[ \t]*##")
    ipa automountkey-add nis $4 --key="$key" --info="$info"
done
```

このスクリプトでは NIS 自動マウント情報がエクスポートされ、LDAP データ変換形式 (LDIF) の自動マウントの場所と関連マップが生成され、LDIF ファイルが IdM Directory Server にインポートされます。詳細については、「[自動マウントマップの NIS クライアントへの公開](#)」を参照してください。

2. IdM **admin** ユーザーとして認証します。

```
[root@nis-server ~]# kinit admin
```

3. 以下のようにスクリプトを実行します。

```
[root@nis-server ~]# sh /root/nis-automounts.sh location nisdomain \
    nis-master.example.com map_name
```

21.5.4. NIS ユーザー認証用に脆弱なパスワードハッシュを有効にする

Directory Server コンポーネントのデフォルト設定を使用すると、**userPassword** 属性に保存されているパスワードはソルト付き SHA を使ってハッシュ化されます。お使いの NIS クライアントでパスワード用に脆弱なハッシュ化アルゴリズムが必要な場合は、パスワード保存スキームの設定を更新します。

脆弱なパスワードハッシュ化スキームの有効化は、**userPassword** 属性に保存されているパスワードにのみ影響があります。Kerberos はこの属性を使用しないので、この設定は Kerberos 暗号化に影響しないことに留意してください。

たとえば、**CRYPT** ハッシュ化パスワードを有効にするには、以下を実行します。

```
[root@server ~]# ldapmodify -D "cn=Directory Manager" -W -p 389 -h
ipaserver.example.com -x
dn: cn=config
changetype: modify
replace: passwordStorageScheme
passwordStorageScheme: crypt
```



注記

パスワードハッシュは暗号化解除ができないため、Directory Server は既存のパスワードハッシュを変換しません。サーバーは、ストレージスキームの変更後に設定されたパスワードにのみ、新規パスワードストレージを適用します。

パート V. 認証メカニズムの管理

第22章 ユーザー認証

本章では、ユーザーパスワード、SSH キー、証明書の管理、ワンタイムパスワード (OTP) およびスマートカード認証の設定方法に関する情報など、ユーザー認証のメカニズム管理について説明します。



注記

Kerberos を使用した Identity Management (IdM) へのログインに関するドキュメントは[5章IdM サーバーおよびサービスの基本的な管理](#)を参照してください。

22.1. ユーザーパスワード

22.1.1. ユーザーパスワードの変更およびリセット

他のユーザーのパスワードを変更するパーミッションのない通常ユーザーは、自身のパスワードしか変更できません。個人のパスワードは以下のように変更します。

- IdM パスワードポリシーに対応する必要があります。パスワードポリシーの設定に関する詳細は[27章パスワードポリシーの定義](#)を参照してください。

パスワード変更の権限がある管理者およびユーザーは、新規ユーザーの初期パスワードの設定、既存ユーザーのパスワードのリセットが可能です。パスワードは以下のように変更します。

- IdM パスワードポリシーに対応する必要はありません。
- 正しく初回ログインができた時点で失効します。パスワードが失効すると、IdM はユーザーに対して失効したパスワードを直ちに変更するように求めます。この動作を変更するには「[次のログイン時にパスワード変更のプロンプトを表示しないでパスワードのリセットを有効化する方法](#)」を参照してください。



注記

LDAP Directory Manager (DM) ユーザーは、LDAP ツールを使用してユーザーパスワードを変更できます。新しいパスワードは、IdM パスワードポリシーよりも優先されます。DM で設定したパスワードは、初回ログインで失効しません。

22.1.1.1. Web UI: 個人のパスワードの変更

1. 右上隅の **User name** → **Change password** をクリックします。

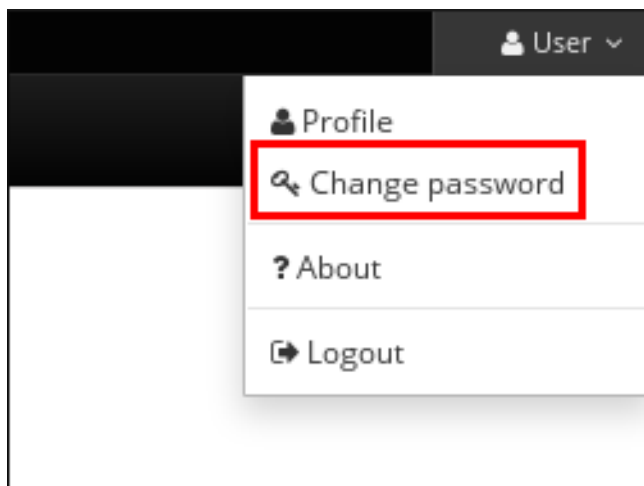


図22.1 パスワードのリセット

2. 新規パスワードを入力します。

22.1.1.2. Web UI: 別ユーザーのパスワードのリセット

1. **Identity** → **Users** を選択します。
2. 編集するユーザー名をクリックします。
3. **Actions** → **Reset password** をクリックします。

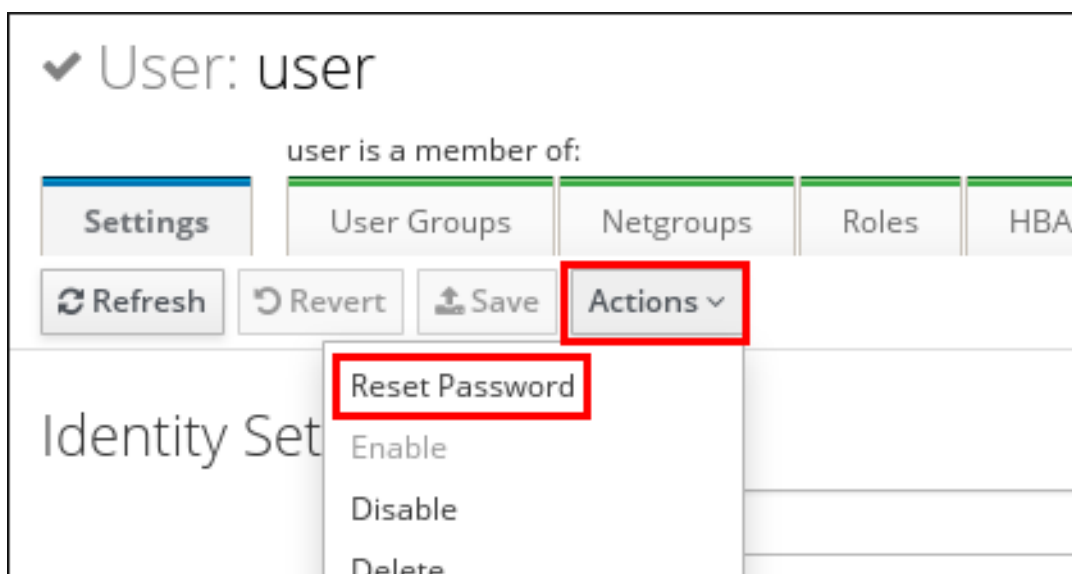


図22.2 パスワードのリセット

4. 新規パスワードを入力して **Reset Password** をクリックします。

The image shows a 'Reset Password' dialog box. It contains two text input fields. The first is labeled 'New Password' with a blue asterisk to its right. The second is labeled 'Verify Password' with a blue asterisk to its right. Both fields contain masked characters (dots). At the bottom right, there are two buttons: 'Reset Password' and 'Cancel'. The 'Reset Password' button is highlighted with a red rectangular border.

図22.3 新規パスワードの確定

22.1.1.3. コマンドライン: 別のユーザーのパスワード変更またはリセット

自身のパスワードの変更または、別のユーザーのパスワード変更またはリセットするには、**ipa user-mod** コマンドに **--password** オプションを追加します。このコマンドにより、新規パスワードを入力するように求められます。

```
$ ipa user-mod user --password
Password:
Enter Password again to verify:
-----
Modified user "user"
-----
...
```

22.1.1.2. 次回のログイン時にパスワード変更のプロンプトを表示しないでパスワードのリセットを有効化する方法

デフォルトでは、管理者が別のユーザーのパスワードをリセットすると、ユーザーが初めて正しくログインできた時点でそのパスワードが失効します。詳細は「[ユーザーパスワードの変更およびリセット](#)」を参照してください。

管理者が設定したパスワードを初めて使用した時点で失効させないようにするには、ドメイン内の全 Identity Management サーバーで以下の変更を加えます。

- パスワードの同期エントリーを編集します (**cn=ipa_pwd_extop,cn=plugins,cn=config**)。
- **passSyncManagersDNs** 属性で管理ユーザーアカウントを指定します。この属性には、複数の値を指定することができます。

たとえば、**ldapmodify**ユーティリティを使用して **admin** ユーザーを指定するには以下を実行します。

```
$ ldapmodify -x -D "cn=Directory Manager" -W -h ldap.example.com -p 389

dn: cn=ipa_pwd_extop,cn=plugins,cn=config
changetype: modify
add: passSyncManagersDNs
passSyncManagersDNs: uid=admin,cn=users,cn=accounts,dc=example,dc=com
```

**警告**

これらの追加の権限が必要なユーザーのみを指定します。 *passSyncManagerDNs* に含まれる全ユーザーは以下が可能です。

- すぐ後にパスワードのリセットせずにパスワードの変更操作を実行する
- パスワード強度や履歴の制限を適用しなくていいようにパスワードポリシーを回避する

22.1.3. ログイン失敗後のユーザーアカウントのロック解除

ユーザーのログイン時にパスワードを特定の回数間違えると、IdM はそのユーザーがログインできないようにアカウントをロックします。IdM からは、ユーザーアカウントがロックされたことを示す警告メッセージは表示されない点に注意してください。

**注記**

ログインに失敗できる回数やロックアウトの期間の設定に関する情報は、[27章 パスワードポリシーの定義](#)を参照してください。

IdM により、指定の期間が経過すると自動的にユーザーアカウントのロックが解除されます。または、管理者が手動でユーザーアカウントのロックを解除してください。

ユーザーアカウントの手動のロック解除

ユーザーアカウントのロックを解除するには **ipa user-unlock** コマンドを使用します。

```
$ ipa user-unlock user
-----
Unlocked account "user"
-----
```

このコマンドを実行すると、ユーザーは再度、ログインできるようになります。

22.1.3.1. ユーザーアカウントのステータス確認

ユーザーのログイン失敗回数を表示するには **ipa user-status** コマンドを使用します。表示の回数がログインを試行できる回数を超えると、ユーザーアカウントはロックされます。

```
$ ipa user-status user
-----
Account disabled: False
-----
Server: example.com
Failed logins: 8
Last successful authentication: 20160229080309Z
Last failed authentication: 20160229080317Z
Time now: 2016-02-29T08:04:46Z
```

 Number of entries returned 1

22.2. ワンタイムパスワード



重要

OTP 認証の IdM ソリューションは、Red Hat Enterprise Linux 7.1 以降を稼働するクライアントのみでサポートされます。

ワンタイムパスワード (OTP) は、1 回の認証セッションのみで有効なパスワードです。これは一度使用すると、無効になります。従来の静的なパスワードとは異なり、認証トークンで生成される OTP は常に変更されます。OTP は、二要素認証の一部として使用されます。

1. 従来のパスワードを使用したユーザー認証
2. ユーザーは、認識済みの OTP トークンで生成された OTP コードを入力します。

二要素認証は、従来のパスワードだけを使用する認証に比べると安全であると考えられています。ログイン中に、侵入者により OTP が傍受された場合でも、傍受された OTP は、一旦認証に成功すると使用できなくなるので、侵入者が使用する頃にすでに無効になっています。



警告

現在、IdM の OTP サポートに関係のあるセキュリティとその他の制限は以下の通りです。

- 最も重要なセキュリティ制限として、システム全体でリプレイアタックの被害を受けやすくなる点です。レプリケーションは非同期で実行されるので、OTP コードはレプリケーション中に再利用でき、ユーザーは同時に 2 台のサーバーにログインできる可能性があります。ただし、包括的な暗号化があると、この脆弱性を悪用するのは困難です。
- OTP 認証をサポートしていないクライアント経由では、ticket-granting ticket (TGT) を取得することができません。これは、`mod_auth_kerb` モジュールまたは Generic Security Services API (GSSAPI) を使用した認証などのユースケースに影響する可能性があります。

22.2.1. IdM での OTP の機能

22.2.1.1. IdM でサポートされる OTP トークン

ソフトウェアおよびハードウェアトークン

IdM は、ソフトウェアおよびハードウェア両方のトークンをサポートします。

ユーザーおよび管理者が管理するトークン

ユーザーは自身のトークンを、管理者はユーザーの代わりにトークンを管理します。

ユーザー管理のトークン

ユーザーは、トークの作成、編集、削除など、Identity Management でユーザーが管理するトークンを完全に制御できます。

管理者が管理するトークン

管理者は、管理者が管理するトークンをユーザーのアカウントに追加し、ユーザーはこれらのトークンへの読み取り専用アクセスが割り当てられます。ユーザーはトークンの管理または変更パーミッションがなく、設定する必要はありません。

アクティブなトークンが 1 つしかない場合には、削除することも、無効にすることもできない点に注意してください。管理者は、自分のアクティブなトークンが 1 つの場合は削除や無効化はできませんが、別のユーザートークンは、アクティブなトークンが 1 つの場合でも削除または無効化することができます。

サポートされる OTP アルゴリズム

Identity Management には、以下にある、標準の OTP メカニズム 2 つをサポートしています。

- HMAC ベースのワンタイムパスワード (HOTP) アルゴリズムは、カウンターに基づいています。HMAC は、Hashed Message Authentication Code (ハッシュメッセージ認証コード) を表しています。
- 時間ベースのワンタイムパスワード (TOTP) アルゴリズムは、時間ベースの移動要素をサポートする HOTP の拡張機能です。

22.2.1.2. 利用可能な OTP 認証方法

OTP 認証を有効化する際に、以下の認証方法から選択できます。

二要素認証 (パスワード + OTP)

この方法では、ユーザーは常に標準のパスワードと OTP コードを入力するように求められます。

パスワード

この方法では、ユーザーは標準のパスワードのみを使用した認証を選択するオプションがあります。

RADIUS プロキシサーバー認証

OTP の検証用に RADIUS サーバーを設定する方法は「[商用 OTP ソリューションからの移行](#)」を参照してください。

グローバルおよびユーザー固有の認証方法

グローバルにも、または個別ユーザーにもこれらの認証方法を設定することができます。

- デフォルトでは、ユーザー固有の認証方法の設定は、グローバルの設定よりも優先されます。認証方法がユーザーに設定されていない場合には、グローバルに定義された方法が適用されます。
- ユーザー毎の認証方法を無効化することができます。これにより、IdM はユーザー毎の設定を無視して常にグローバルの設定を適用することができます。

複数の認証方法の統合

複数の方法を同時に設定すると、どちらか 1 つが成功すれば認証されます。以下に例を示します。

- 二要素認証およびパスワード認証を設定すると、ユーザーはコマンドラインを使用する場合にはパスワード (1 つ目の要素) を指定する必要がありますが、OTP (2 つ目の要素) はオプションです。

```
First Factor:
Second Factor (optional):
```

- Web UI ではユーザーは両要素を指定する必要があります。



注記

OTP などの特定の認証方法を設定する必要がある個別ホストまたはサービスもあります。一段階要素のみを使用してこのようなホストやサービスに対する認証を試行すると、アクセスが拒否されます。[「ユーザーの認証情報をもとにサービスやホストへのアクセス制限」](#)を参照してください。

ただし、RADIUS および他の認証方法が設定されている場合には、例外が少しあります。

- Kerberos は常に RADIUS を使用しますが LDAP は RADIUS を使用しません。LDAP が認識するのは、パスワードと二要素認証のオプションのみです。
- 外部の 2 要素認証プロバイダーを使用する場合は、使用しているアプリケーションから Kerberos を使用します。ユーザーがパスワードのみで認証するようにするには、LDAP を使用します。アプリケーションが Apache モジュールと SSSD を活用するようにすることが推奨されます。そうすることで、Kerberos か LDAP のいずれかを設定することができます。

22.2.1.3. GNOME のキーリングサービスサポート

IdM は、OTP 認証と GNOME キーリングサービスを統合します。GNOME キーリング統合では、一段階要素と二段階要素を別に入力する必要があります。

```
First factor: static_password
Second factor: one-time_password
```

22.2.1.4. OTP でのオフライン認証

IdM では、オフライン OTP 認証がサポートされますが、オフラインでログインできるようにするには、システムがオンラインの時にユーザーは静的なパスワードと OTP を別に入力して最初の認証を行う必要があります。

```
First factor: static_password
Second factor: one-time_password
```

オンラインでのログイン時に両パスワードを個別に入力すると、それ以降のログイン時に中央認証サーバーが利用できない場合でも認証できるようになっています。ユーザーがオフラインで認証を行うと、IdM は第一要素である従来の静的なパスワードだけを求める点に注意してください。

IdM は、**First factor** のプロンプトで静的なパスワードと OTP を 1 つの文字列として入力できるようにサポートします。ただし、オフラインの OTP 認証とは互換性がない点に注意してください。ユーザーが両要素を単一のプロンプトで入力した場合には、IdM は認証時に常に中央認証サーバに問い合わせる必要があるため、認証時にはシステムがオンラインの状態でなければなりません。



重要

ラップトップなどオフラインで操作するデバイスで OTP 認証を使用する場合は、Red Hat は静的なパスワードと OTP を個別に入力して、オフライン認証を利用可能にすることを推奨します。これを行わないと、システムがオフラインになった後にログインすると IdM により拒否されます。

OTP オフライン認証の恩恵を受けるには、静的と OTP パスワードを個別に入力する以外に、以下の条件も満たすようにしてください。

- `/etc/sss/sss.conf` ファイルの `cache_credentials` オプションを **True** に設定すると、第一要素のパスワードのキャッシュが有効になります。
- 第一要素の静的パスワードが、`/etc/sss/sss.conf` の `cache_credentials_minimal_first_factor_length` オプションで定義されているパスワード長の要件を満たしていること。デフォルトでは、少なくとも 8 文字以上指定する必要があります。オプションに関する情報は、`sss.conf(5)` の man ページを参照してください。

`/etc/sss/sss.conf` で `krb5_store_password_if_offline` オプションが **true** に設定されている場合でも、その時点で OTP がすでに無効になっている可能性があるため SSSD は Kerberos ticket-granting ticket (TGT) の更新は試行しません。このような状況で TGT を取得するには、両要素を使用して認証する必要があります。

22.2.2. OTP 認証の有効化

OTP 関連で利用可能な認証方法に関する詳細は「[利用可能な OTP 認証方法](#)」を参照してください。

以下を使用して OTP 認証を有効化します。

- Web UI。「[Web UI: OTP 認証の有効化](#)」を参照してください。
- コマンドライン。「[コマンドライン: OTP 認証の有効化](#)」を参照してください。

Web UI: OTP 認証の有効化

全ユーザーに対してグローバルに認証方法を設定するには、以下を実行します。

1. **IPA Server** → **Configuration** を選択します。
2. **User Options** エリアで、必要な **Default user authentication types** を選択します。

Default user authentication types ⓘ	<input type="checkbox"/> Disable per-user override <input type="checkbox"/> Password <input type="checkbox"/> Radius <input checked="" type="checkbox"/> Two factor authentication (password + OTP)
-------------------------------------	--

図22.4 ユーザー認証方法

グローバル設定がユーザー別の設定で上書きされないようにするには、**Disable per-user override** を選択します。**Disable per-user override** を選択しない場合には、ユーザー別に設定した認証方法がグローバル設定よりも優先されます。

ユーザーベースで個別に認証方法を設定するには、以下を実行します。

1. **Identity** → **Users** を選択して、編集するユーザーの名前をクリックします。
2. **Account Settings** エリアで、必要な **User authentication types** を選択します。

User authentication types ⓘ

☐ Password
☐ Radius
☒ Two factor authentication (password + OTP)

図22.5 ユーザー認証方法

コマンドライン: OTP 認証の有効化

全ユーザーに対してグローバルに認証方法を設定するには、以下を実行します。

1. **ipa config-mod --user-auth-type** コマンドを実行します。たとえば、グローバル認証方法を二要素認証に設定するには以下を実行します。

```
$ ipa config-mod --user-auth-type=otp
```

--user-auth-type に入力できる値の一覧については、**ipa config-mod --help** コマンドを実行します。

2. ユーザー別の上書きを無効にする、つまりユーザーの設定がグローバル設定を上書きしないようにするには、**--user-auth-type=disabled** オプションも追加します。たとえば、二要素認証にグローバル認証方法を設定してユーザーの上書きを無効にするには以下を設定します。

```
$ ipa config-mod --user-auth-type=otp --user-auth-type=disabled
```

--user-auth-type=disabled を設定しない場合には、ユーザー毎に設定した認証方法がグローバル設定よりも優先されます。

指定のユーザーに対して個別に認証方法を設定するには以下を行います。

- **ipa user-mod --user-auth-type** コマンドを実行します。たとえば、**user** が二要素認証を使用するように設定するには以下を実行します。

```
$ ipa user-mod user --user-auth-type=otp
```

複数の認証方法を設定するには、**--user-auth-type** を複数回追加します。たとえば、パスワードと要素認証を全ユーザーにグローバルに設定するには、以下を実行します。

```
$ ipa config-mod --user-auth-type=otp --user-auth-type=password
```

22.2.3. ユーザー管理のソフトウェアトークンの追加

1. 標準のパスワードでログインします。
2. モバイルデバイスに **FreeOTP Authenticator** アプリケーションがインストールされていることを確認します。**FreeOTP Authenticator** のダウンロードについては、[FreeOTP source page](#) を参照してください。
3. IdM Web UI またはコマンドラインでソフトウェアトークンを作成します。

- Web UI でトークンを作成するには、**OTP tokens** タブで **Add** をクリックします。管理者としてログインしている場合は **Authentication** から **OTP Tokens tab** タブにアクセスできます。

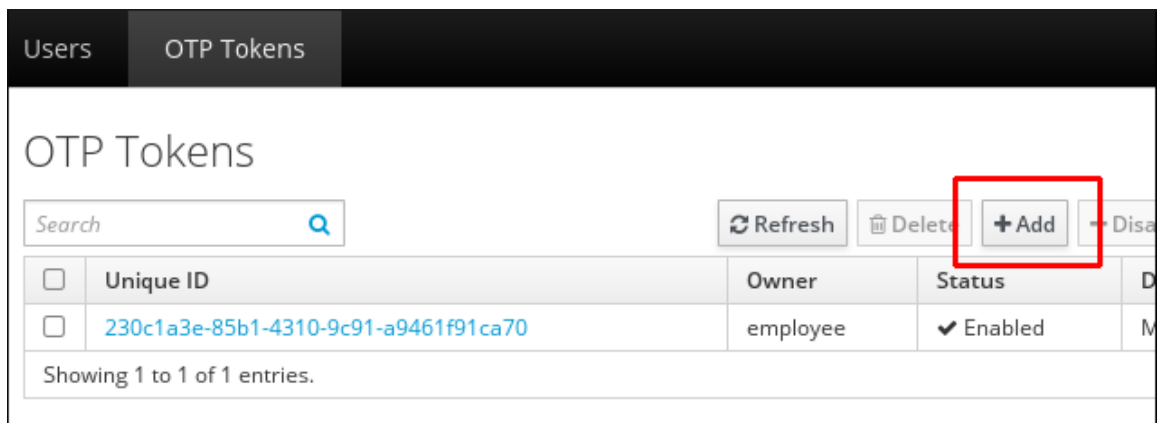


図22.6 ユーザー用の **OTP** トークンの追加

- コマンドラインからトークンを作成するには **ipa otptoken-add** コマンドを実行します。

```
$ ipa otptoken-add
-----
Added OTP token ""
-----
Unique ID: 7060091b-4e40-47fd-8354-cb32fec548a
Type: TOTP
...
```

ipa otptoken-add に関する情報は、**--help** オプションを追加してこのコマンドを実行します。

- Web UI またはコマンドラインに QR コードが表示されます。QR コードを **FreeOTP Authenticator** でスキャンして、モバイルデバイスのトークンを提供します。

22.2.4. ユーザー管理の **YubiKey** ハードウェアトークンの追加

YubiKey トークンなど、プログラム可能なハードウェアトークンは、コマンドラインからのみ追加できます。YubiKey ハードウェアをトークンの所有ユーザーとして追加するには、以下を実行します。

- 標準のパスワードでログインします。
- YubiKey トークンを挿入します。
- ipa otptoken-add-yubikey** コマンドを実行します。
 - YubiKey に空のスロットがある場合には、コマンドは、その空のスロットを自動的に選択します。
 - 空のスロットがない場合には **--slot** オプションを使用して手動でスロットを選択する必要があります。以下に例を示します

```
$ ipa otptoken-add-yubikey --slot=2
```

これは、選択したスロットを上書きする点に注意してください。

22.2.5. 管理者としてユーザー用のトークン追加

管理者としてソフトウェアトークンを追加するには以下を実行します。

1. 管理者としてログインしていることを確認します。
2. モバイルデバイスに **FreeOTP Authenticator** アプリケーションがインストールされていることを確認します。**FreeOTP Authenticator** のダウンロードについては、[FreeOTP source page](#) を参照してください。
3. IdM Web UI またはコマンドラインでソフトウェアトークンを作成します。
 - Web UI でトークンを作成するには、**Authentication** → **OTP Tokens** を選択して、OTP トークンの一覧の上部にある **Add** をクリックします。**Add OTP Token** フォームで、トークンの所有者を選択します。

図22.7 管理者が管理するソフトウェアトークンの追加

- コマンドラインからトークンを作成するには **--owner** オプションを指定して **ipa otptoken-add** コマンドを実行します。以下に例を示します。

```
$ ipa otptoken-add --owner=user
-----
Added OTP token ""
-----
Unique ID: 5303baa8-08f9-464e-a74d-3b38de1c041d
Type: TOTP
...
```

4. Web UI またはコマンドラインに QR コードが表示されます。QR コードを **FreeOTP Authenticator** でスキャンして、モバイルデバイスのトークンを提供します。

管理者として、YubiKey トークンのようなプログラム可能なハードウェアトークンを追加するには、以下の手順に従います。

1. 管理者としてログインしていることを確認します。
2. YubiKey トークンを挿入します。
3. **--owner** オプションを指定して **ipa otptoken-add-yubikey** コマンドを実行します。以下に例を示します。

```
$ ipa otptoken-add-yubikey --owner=user
```

22.2.6. 商用 OTP ソリューションからの移行

商用の OTP ソリューションから IdM をネイティブとする OTP ソリューションに、大規模なデプロイメントを移行できるように、IdM では、OTP 確認をサードパーティーの RADIUS サーバーにオフロードする方法をユーザーのサブセットに提供します。管理者は、複数の個別の RADIUS サーバーを含む RADIUS プロキシのセットを作成します。次にこれらプロキシセットのひとつをユーザーに割り当てます。ユーザーに RADIUS プロキシのセットが割り当てられている限り、IdM は他のすべての認識メカニズムを迂回します。



注記

IdM では、サードパーティーシステムでのトークン管理やトークンの同期をサポートしていません。

OTP 確認用の RADIUS サーバーを設定し、ユーザーをプロキシサーバーに追加するには、以下の手順に従います。

1. **radius** のユーザー認証方法が有効になっていることを確認します。[「OTP 認証の有効化」](#)を参照してください。
2. **ipa radiusproxy-add proxy_name** コマンドを実行して RADIUS プロキシを追加します。このコマンドにより、必要な情報を入力するように求められます。
3. **ipa user-mod radiususer --radius=proxy_name** コマンドを実行して、追加したプロキシにユーザーを割り当てます。
4. 必要に応じて、**ipa user-mod radiususer --radius-username=radius_user** を実行し、RADIUS に送信するユーザー名を設定します。

これで、ユーザー OTP 認証が RADIUS プロキシサーバーで処理されるようになります。

ユーザーが IdM ネイティブの OTP システムに移行する準備ができたなら、そのユーザーへの RADIUS プロキシの割り当てを削除します。

22.2.7. 現在の認証情報の二要素認証へのプロモート

パスワードと二要素認証の両方を設定しているにも関わらず、パスワードでのみ認証を行った場合には、特定のサービスやホストへのアクセスが拒否される可能性があります ([「ユーザーの認証情報をもとにサービスやホストへのアクセス制限」](#)を参照)。このような場合には、もう一度認証を行うことで、一要素から二要素認証にプロモートしてください。

1. 画面をロックします。画面ロックのデフォルトのキーボードショートカットは **Super key+L** です。
2. 画面のロックを解除します。認証情報を求められたら、パスワードと OTP の両方を使用します。

22.2.8. OTP トークンの再同期

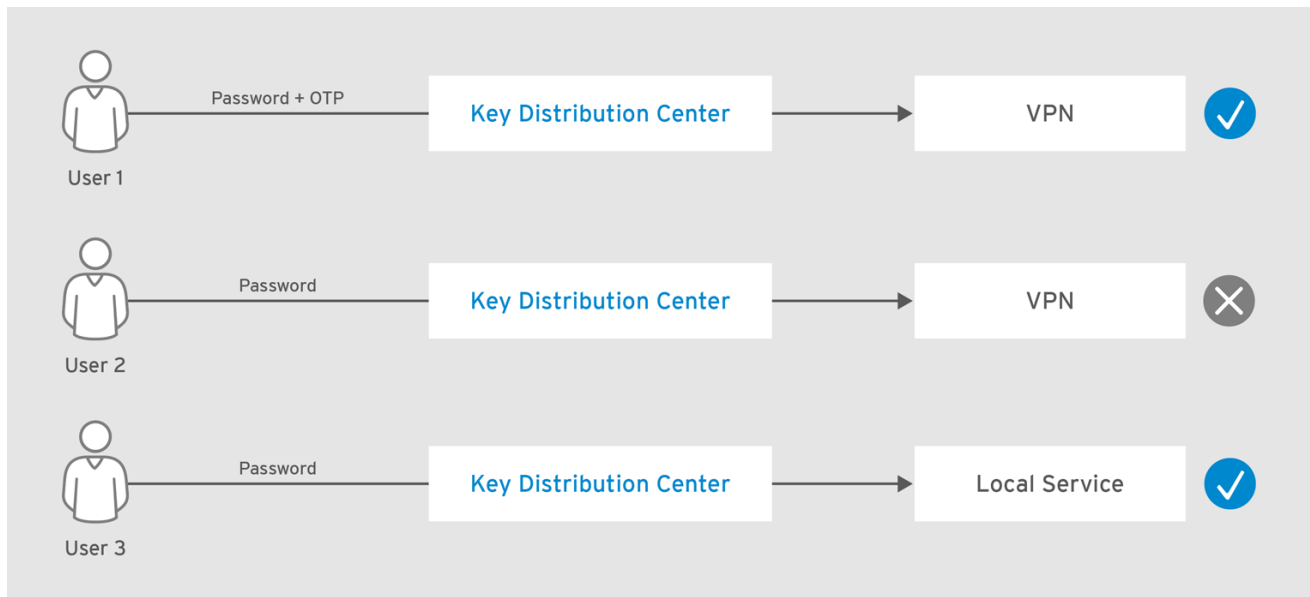
[「OTP トークンが同期されない問題」](#)を参照してください。

22.3. ユーザーの認証情報をもとにサービスやホストへのアクセス制限

IdM でサポートされる認証メカニズムは、認証強度が異なります。たとえば、ワンタイムパスワード (OTP) と標準のパスワードを組み合わせると、標準のパスワードだけを使用するよりは安全であると考えられます。このセクションでは、ユーザーの認証方法をもとに、サービスやホストのアクセスを制限する方法を説明します。

たとえば、以下を設定できます。

- 強力な認証方法を必要とする VPN など極めて重要なセキュリティ関連のサービス
- 強度は低いが便利な認証方法を使用できる、ローカルのログインなど重要でないサービス



RHEL_404973_1016

図22.8 異なる方法を使用した認証例

認証インジケター

サービスやホストへのアクセスは **認証インジケター** で定義されています。

- サービスまたはホストエントリーに含まれるインジケターは、ユーザーがサービスやホストへのアクセスに使用する認証方法を定義します。
- ユーザーの TGT (Ticket Granting Ticket) に含まれるインジケターは、チケットの取得に使用した認証方法を表示します。

プリンシパルのインジケターが TGT のインジケターに一致しない場合は、ユーザーのアクセスは拒否されます。

22.3.1. 特定の認証方法を必要とするホストまたはサービスの設定

以下を使用してホストまたはサービスを設定します。

- Web UI。『[Web UI: 特定の認証方法を必要とするホストまたはサービスの設定](#)』を参照してください。
- コマンドライン。『[コマンドライン: 特定の認証方法を必要とするホストまたはサービスの設定](#)』を参照してください。

Web UI: 特定の認証方法を必要とするホストまたはサービスの設定

1. **Identity** → **Hosts** または **Identity** → **Services** を選択します。
2. 所定のホストまたはサービスの名前をクリックします。
3. **Authentication indicators** で所定の認証方法を選択します。

- 。たとえば **OTP** を選択すると、有効な OTP コードとパスワードを使用したユーザーのみがホストまたはサービスにアクセスができます。
- 。 **OTP** および **RADIUS** 両方を選択すると、OTP または RADIUS のいずれかを提示すれば、アクセスが許可されます。

4. ページ上部にある **Save** をクリックします。

コマンドライン: 特定の認証方法を必要とするホストまたはサービスの設定

1. オプション: **ipa host-find** または **ipa service-find** コマンドを使用して、ホストまたはサービスを特定します。
2. **ipa host-mod** または **ipa service-mod** コマンドに **--auth-ind** オプションを指定して実行し、必要な認証インジケータを追加します。**--auth-ind** で利用できる値の一覧は、**ipa host-mod --help** または **ipa service-mod --help** コマンドの出力を参照してください。

たとえば、**--auth-ind=otp** では、有効な OTP コードとパスワードを使用したユーザーのみがホストまたはサービスへのアクセスが許可されます。

```
$ ipa host-mod server.example.com --auth-ind=otp
```

```
-----  
Modified host "server.example.com"
```

```
-----  
Host name: server.example.com
```

```
...
```

```
Authentication Indicators: otp
```

```
...
```

OTP と RADIUS の両方のインジケータを追加した場合には、OTP または RADIUS のいずれかを提示するだけでアクセスが許可されます。

22.4. ユーザーの公開 SSH キーの管理

Identity Management では、公開 SSH 鍵をユーザーエントリーにアップロードできます。適切な公開 SSH 鍵にアクセスできるユーザーは、**ssh** を使用して Kerberos 認証情報なしに IdM マシンにログインできます。**pam_krb5** が正しく設定されているか、SSSD が IdM サーバーの ID プロバイダーとして使用されている場合には、ログイン後に Kerberos ticket-granting ticket (TGT) を受け取ります。詳細は「[Kerberos チケットの自動取得](#)」を参照してください。

SSH 秘密鍵ファイルを利用できないマシンからログインしている場合には、Kerberos 認証情報を提示した認証も可能である点に注意してください。

SSH 鍵の自動キャッシュおよび取得

IdM サーバーまたはクライアントのインストール中に、SSSD は自動的に ユーザーとホストの SSH 鍵をキャッシュ、取得するよう設定されます。これにより、IdM は SSH 鍵の集約された汎用リポジトリとしての役割を果たすことができます。

サーバーまたはクライアントがインストール時に設定されていない場合には、マシンの SSSD を手動で設定してください。[システムレベルの認証ガイド](#) を参照してください。SSSD での SSH 鍵のキャッシュには、ローカルマシンでの管理者権限が必要だ点に注意してください。

SSH 鍵の形式要件

IdM では、以下の 2 つの SSH 鍵の形式を使用できます。

OpenSSH スタイルの鍵

この形式に関する詳細は [RFC 4716](#) を参照してください。

Raw RFC 4253 スタイルの鍵

この形式に関する詳細は [RFC 4253](#) を参照してください

IdM は自動的に RFC 4253 スタイルの鍵を OpenSSH スタイルの鍵に自動的に変換してから、IdM LDAP サーバーに保存する点に注意してください。

id_rsa.pub などの鍵ファイルは、鍵タイプ、鍵、追加のコメントまたは識別子の 3 つの部分で公正されます。以下の例では、鍵のタイプは RSA で、コメントにより鍵と **client.example.com** ホスト名とを関連付けます。

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDMM4xPu54Kf2dx7C4Ta2F7vnIzuL1i6P21TTKniskjFu
A+r
qW06588e7v14Im4VejwnNk352gp49A62qSV0zp8IKA9xdtyRmHYCTUvmkcyspZvFRI713zfRKQ
VFyJ0qHmW/m
dCmak7QBxYou2ELSPH3pe8MYTQIuIkDSu5Zbsrqedg1VGkSJxf7mDnCSPNWWzAY9AFB9Lmd2m
2xZmNgVAQEQ
nZXNMaIlroLD/51rmMSkJGHGb1068kEq9Z client.example.com
```

鍵を IDM にアップロードする場合には、3 つの部分すべてをアップロードするか、鍵自体のみをアップロードできます。鍵だけをアップロードする場合には、IdM はアップロードされた鍵をもとに、自動的に RSA または DSA など鍵タイプを自動的に識別します。

22.4.1. SSH 鍵の生成

OpenSSH **ssh-keygen** ユーティリティーを使用して SSH 鍵を生成できます。このユーティリティーは、公開鍵の場所に関する情報を表示します。以下に例を示します。

```
$ ssh-keygen -t rsa -C user@example.com
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:GAUIDVVEgly7rs1lTWP6oguHz8BKvyZkpqCqVSsmi7c user@example.com
The key's randomart image is:
+---[ RSA 2048]-----+
|
|      + .
|     + = .
|      = +
|      . E S..
|     . . . .0
|     . . . .00.
|     . 0 . .+.+0
|     0 . 0 . .0+0
+-----+

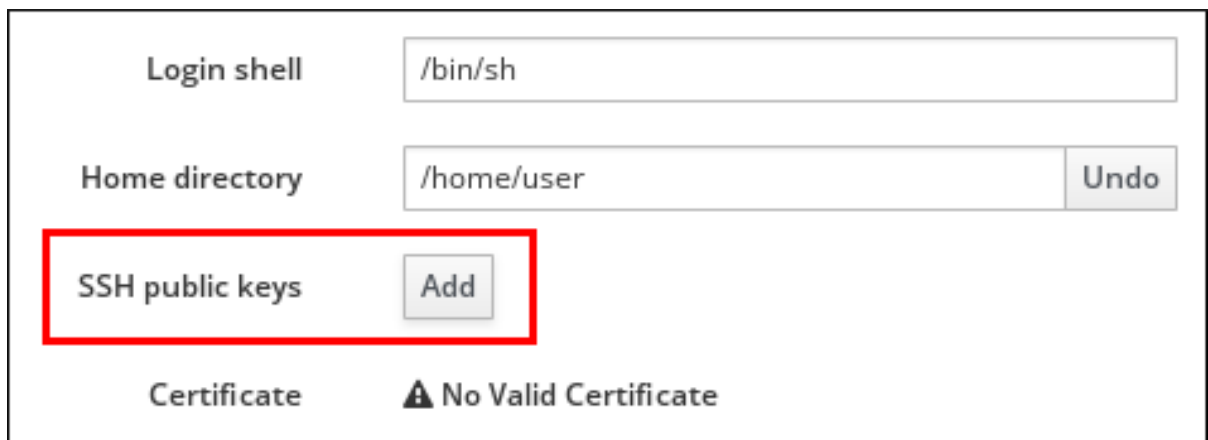
```

ユーザーの SSH 鍵をアップロードするには、表示されたファイルに保存されている公開鍵の文字列を使用します。

22.4.2. ユーザーの SSH 鍵のアップロード

22.4.2.1. Web UI: ユーザーの SSH 鍵のアップロード

1. **Identity** → **Users** を選択します。
2. 編集するユーザー名をクリックします。
3. **Account Settings** エリアの **Settings** タブで、**SSH public keys: Add** をクリックしてください。



The screenshot shows a user settings form. The 'Login shell' field is set to '/bin/sh'. The 'Home directory' field is set to '/home/user' with an 'Undo' button. The 'SSH public keys' section is highlighted with a red rectangle and contains an 'Add' button. Below this, there is a 'Certificate' section showing 'No Valid Certificate' with a warning icon.

図22.9 アカウント設定の SSH 公開鍵

4. Base 64 でエンコードされた公開鍵の文字列を貼り付け、**Set** をクリックします。



The screenshot shows a 'Set SSH key' dialog box. It has a title bar with a close button. The main area is labeled 'SSH public key:' and contains a text area with a Base 64 encoded SSH public key. The key starts with 'ssh-rsa' and ends with 'user@example.com'. At the bottom right, there are 'Set' and 'Cancel' buttons, with the 'Set' button highlighted by a red rectangle.

図22.10 公開鍵の貼り付け

5. ページ上部の **Save** をクリックします。

22.4.2.2. コマンドライン: ユーザーの SSH 鍵のアップロード

ipa user-mod コマンドを使用して、**--sshpubkey** オプションを指定して Base 64 でエンコードされた公開鍵の文字列を渡します。

たとえば、鍵のタイプ、鍵自体、ホスト名の識別子をアップロードするには以下を実行します。

```
$ ipa user-mod user --sshpubkey="ssh-rsa AAAAB3Nza...Snc5dv==
client.example.com"
```

複数の鍵をアップロードするには **--sshpubkey** を複数回実行します。たとえば、SSH 鍵を 2 つアップロードするには以下を実行します。

```
--sshpubkey="AAAAB3Nza...Snc5dv==" --sshpubkey="Rj1zYQo...ZEt0TAo="
```



注記

鍵の文字列をコマンドラインに手動で貼り付ける代わりに、コマンドのリダイレクトを使用して、鍵を含むファイルを参照することができます。以下の例を示します。

```
$ ipa user-mod user --sshpubkey="$(cat ~/.ssh/id_rsa.pub)" --
sshpubkey="$(cat ~/.ssh/id_rsa2.pub)"
```

22.4.3. ユーザーキーの削除

SSH 鍵を削除するには以下を実行します。

- Web UI を使用する場合は「[Web UI: ユーザーの SSH 鍵の削除](#)」を参照してください。
- コマンドラインを使用する場合は「[コマンドライン: ユーザーの SSH 鍵の削除](#)」を参照してください。

22.4.3.1. Web UI: ユーザーの SSH 鍵の削除

1. **Identity → Users** を選択します。
2. 編集するユーザー名をクリックします。
3. **Account Settings** エリアの **Settings** タブで、削除する鍵の横にある **Delete** をクリックします。

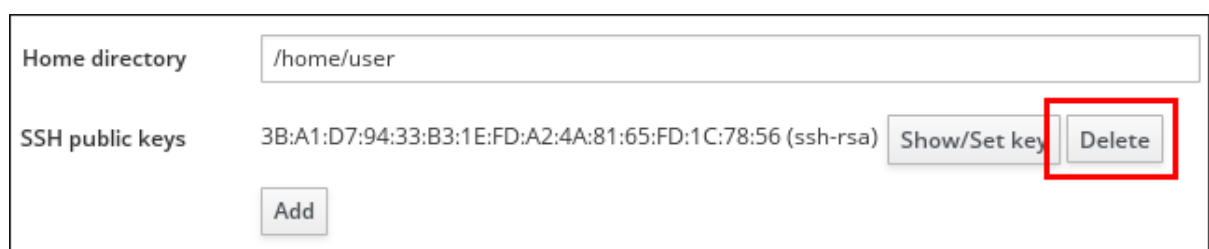


図22.11 ユーザーの SSH 公開鍵の削除

4. ページ上部の **Save** をクリックします。

22.4.3.2. コマンドライン: ユーザーの SSH 鍵の削除

ユーザーアカウントに割り当てられた SSH 鍵を削除するには、鍵を指定せずに、**ipa user-mod** コマンドに **--sshpubkey** オプションを追加します。


```
$ ipa user-mod user --sshpubkey=
```

特定の SSH 鍵を削除する場合には **--sshpubkey** オプションを使用して、保持する鍵を指定します。

22.5. SSSD が OPENSSH サービス用にキャッシュを提供するように設定する方法

System Security Services Daemon (SSSD) は、OpenSSH など複数のシステムサービスへのインターフェースを提供します。詳細は、『「システムレベルの認証ガイド」』の「[SSSD による認証情報の使用とキャッシング](#)」を参照してください。

以下のセクションでは、SSSD がマシンとユーザーの SSH 鍵をキャッシュするように設定する方法を説明します。

22.5.1. SSSD と OpenSSH との連携方法

OpenSSH は SSH プロトコルの実装です。OpenSSH は、認証エンティティーを特定する **公開鍵と秘密鍵**のペアをもとに、2 つのシステム間でセキュアな暗号化接続を構築します。詳細は、『システム管理者のガイド』の「[OpenSSH](#)」を参照してください。

SSSD は、マシンやユーザーの SSH 公開鍵の認証情報キャッシュとして機能します。この設定では、

1. OpenSSH は、SSSD を参照してキャッシュされた鍵があるかを確認するように設定されます。
2. SSSD は Identity Management (IdM) ドメインを使用し、IdM は公開鍵とホストの情報を保存します。



注記

IdM ドメインの Linux マシンのみが OpenSSH の鍵のキャッシュとして SSSD を使用できます。Windows などの他のマシンはできません。

SSSD のホスト鍵の管理方法

SSSD は以下を実行してホスト鍵を管理します。

1. ホストシステムから公開ホスト鍵を取得します。
2. **/var/lib/sss/pubconf/known_hosts** ファイルにホスト鍵を保存します。
3. ホストマシンとの接続を確率します。

必要な設定手順の詳細は、「[OpenSSH がホスト鍵に SSSD を使用するように設定する手順](#)」を参照してください。

SSSD のユーザー鍵の管理方法

SSSD は以下を実行してユーザー鍵を管理します。

1. IdM ドメインのユーザーエントリーからユーザーの公開鍵を取得します。
2. ユーザー鍵を標準の認証鍵形式でカスタムファイル **.ssh/sss_authorized_keys** に保存します。

必要な設定手順の詳細は、「[ユーザー鍵に SSSD を使用するための OpenSSH 設定](#)」を参照してください。

22.5.2. OpenSSH がホスト鍵に SSSD を使用するように設定する手順

システム全体またはユーザー毎の設定を変更することができます。

1. 必要な設定ファイルを開きます。
 - a. ユーザー固有の設定を変更するには、`~/.ssh/config` ファイルを開きます。
 - b. システム全体の設定を変更するには `/etc/ssh/sshd_config` ファイルを開きます。
2. **ProxyCommand** オプションを使用して、どのコマンドを使用して SSH クライアント (`sss_ssh_knownhostsproxy` ユーティリティーで必要な引数とホスト名を指定) に接続するかを指定します。

sss_ssh_knownhostsproxy の詳細は、`sss_ssh_knownhostsproxy(1)` の man ページを参照してください。
3. **GlobalKnownHostsFile** オプションを使用して、SSSD ホストファイル `/var/lib/sss/pubconf/known_hosts` の場所を指定します。このファイルは、デフォルトの OpenSSH `known_hosts` ファイルの代わりに使用します。

以下の例では、SSH が SSSD ドメインで公開鍵を検索して、指定のポートとホストに接続するように設定します。

```
ProxyCommand /usr/bin/sss_ssh_knownhostsproxy -p %p %h
GlobalKnownHostsFile /var/lib/sss/pubconf/known_hosts
```

SSH の設定や設定ファイルの詳細は、`ssh_config(5)` man ページを参照してください。

22.5.3. ユーザー鍵に SSSD を使用するための OpenSSH 設定

システム全体またはユーザー毎の設定を変更することができます。

1. 必要な設定ファイルを開きます。
 - a. ユーザー固有の設定を変更するには、`~/.ssh/config` ファイルを開きます。
 - b. システム全体の設定を変更するには `/etc/ssh/sshd_config` ファイルを開きます。
2. **AuthorizedKeysCommand** オプションを使用して、ユーザー鍵を取得するために実行するコマンドを指定します。
3. **AuthorizedKeysCommandUser** オプションを使用して、コマンドを実行するユーザーアカウントを指定します。

以下の例では、SSH が `user` アカウントで `sss_ssh_authorizedkeys` ユーティリティー実行するように設定します。

```
AuthorizedKeysCommand /usr/bin/sss_ssh_authorizedkeys
AuthorizedKeysCommandUser user
```

sss_ssh_authorizedkeys の詳細は、`sss_ssh_authorizedkeys(1)` の man ページを参照してください。

SSH の設定や設定ファイルの詳細は、`ssh_config(5)` man ページを参照してください。

22.6. IDENTITY MANAGEMENT でのスマートカード認証

Identity Management のスマートカード認証の情報は、[23章Identity Management でのスマートカード認証](#)を参照してください。

22.7. ユーザー証明書

ユーザー証明書に関する情報は[24章ユーザー、ホスト、およびサービス向け証明書の管理](#)を参照してください。

第23章 IDENTITY MANAGEMENT でのスマートカード認証

パスワードの代わりに、スマートカードベースの認証があります。ユーザーの認証情報がスマートカードに格納され、特別なソフトウェアやハードウェアを使用して、その情報にアクセスします。ユーザーはスマートカードリーダーにスマートカードを挿入してから、そのスマートカードの PIN コードを提示します。

本章では、管理者が Identity Management (IdM) でスマートカードベースの認証を設定する方法、および IdM でユーザーがスマートカードを使用して認証を行う方法について説明します。

23.1. IDENTITY MANAGEMENT サーバーのスマートカードリンクの管理

Identity Management の適切なユーザーアカウントとユーザーのスマートカードからの証明書を管理者がリンクしてからでないと、ユーザーは Identity Management ドメインでの認証にスマートカードを使用できません。スマートカードの証明書をユーザーアカウントにリンクすると、ユーザーは所定のロールでスマートカードを認証できるようになります。本セクションでは、Identity Management サーバーでユーザーのスマートカードと 1 つまたは複数ユーザーアカウントのリンクを管理する方法について説明します。

スマートカード証明書を用意してから、スマートカードとユーザーアカウントをリンクしてください。

- スマートカードから証明書を抽出する必要がある場合は「[スマートカードからの証明書のエクスポート](#)」を参照してください。

証明書とユーザーアカウントの間のリンクを作成する方法は以下を参照してください。

- 「[ユーザーアカウントのスマートカード証明書へのリンク](#)」

特定のスマートカード証明書に対応するユーザーアカウントを検索する必要がある場合は、以下を参照してください。

- 「[指定の証明書と合致するユーザーの検索](#)」

23.1.1. スマートカードからの証明書のエクスポート

証明書をエクスポートする手順:

1. スマートカードをリーダーに挿入します。
2. 以下のコマンドを実行してスマートカードの証明書を表示します。出力で認証に使用する証明書を特定して、そのニックネームをメモします。

```
$ certutil -L -d /etc/pki/nssdb/ -h all
Certificate Nickname           Trust Attributes
                               SSL,S/MIME,JAR/XPI

my_certificate                  CT,C,C
```

3. 証明書のニックネームを使用してファイルに証明書を抽出します。たとえば、**user.crt** という名前のファイルに Base64 形式の証明書を抽出するには以下を実行します。

```
$ certutil -L -d /etc/pki/nssdb/ -n 'my_certificate' -r | base64 -w
0 > user.crt
```

base64 ユーティリティーは **coreutils** パッケージに含まれます。

23.1.2. ユーザーアカウントのスマートカード証明書へのリンク

セキュリティオフィサーは、Identity Management サーバーでユーザーアカウントと 1 つまたは複数のユーザーロールアカウントの間のリンクを管理できます。これにより、ユーザーが所定のロールで認証できるようになります。

以下のいずれかのオプションを使用してリンクを作成します。

- 完全な証明書のプロブを使用する
 - Identity Management ユーザーは、「[証明書とユーザーアカウントの間のリンク作成](#)」を参照してください。また、「[証明書およびユーザーアカウントの間のリンクの削除](#)」を使用してこのリンクを削除してください。
 - Active Directory ユーザーは「[Active Directory のユーザーアカウントとスマートカードのリンク](#)」を参照してください。
- 証明書マッピングを使用する: 「[アイデンティティマッピングの設定](#)」

23.1.2.1. 証明書とユーザーアカウントの間のリンク作成

Identity Management のユーザーアカウントおよび証明書をリンクするにはユーザーアカウントに証明書を保存します。

Identity Management システムで以下の手順を実行します。

コマンドライン: 証明書とユーザーアカウントの間のリンク作成

1. Identity Management の管理者としてログインします。

```
$ kinit admin
```

2. **ipa user-add-cert** コマンドを使用してユーザーアカウントにスマートカード証明書を追加します。以下の例を示します。

```
$ cat cert.pem | tail -n +2 | head -n -1 | tr -d '\r\n' | ipa user-add-cert idm_user
```

Web UI: 証明書とユーザーアカウントの間のリンク作成

1. **Identity** → **Users** を選択し、対象のユーザーアカウントをクリックします。
2. **Certificates** エントリーの横にある **Add** をクリックして証明書を入力します。
3. ユーザーアカウントページの上部にある **Save** をクリックします。

その他のリソース

- 外部の証明局 (CA) が発行する証明書の追加と削除に関する詳細は、「[外部 CA 発行の証明書の管理](#)」を参照します。

23.1.2.2. 証明書およびユーザーアカウントの間のリンクの削除

Identity Management のユーザーアカウントと証明書間のリンクを削除するには、ユーザーアカウントから証明書を削除します。

Identity Management システムで以下の手順を実行します。

コマンドライン: 証明書とユーザーアカウントの間のリンク作成

1. Identity Management の管理者としてログインします。

```
$ kinit admin
```

2. 必要なユーザーアカウントを検索します。

```
$ ipa user-show idm_user
User login: idm_user
First name: first_name
Last name: last_name
...
Certificate: MIIC3...
```

3. アカウントから証明書を削除します。

```
$ ipa user-remove-cert idm_user --certificate MIIC3...
```

Web UI: 証明書とユーザーアカウントの間のリンク削除

1. **Identity** → **Users** を選択し、対象のユーザーアカウントをクリックします。
2. 削除する証明書の横にある **Actions** をクリックして、**Delete** を選択します。

その他のリソース

外部の証明局 (CA) が発行する証明書の追加と削除に関する詳細は、[「外部 CA 発行の証明書の管理」](#)を参照します。

23.1.2.3. Active Directory のユーザーアカウントとスマートカードのリンク

Active Directory でユーザーエントリを変更できる場合には、Active Directory のユーザーエントリにユーザーの証明書を保存します。詳細は、Active Directory のドキュメントを参照してください。これにより、Identity Management が Active Directory のユーザーオブジェクトからスマートカードの証明書を読み込むことができますようになります。

Active Directory でユーザーエントリを変更できない場合には、ID ビューにユーザーの証明書を保存します。Identity Management の Default Trust View を使用して証明書を保存するか、新規 ID ビューを作成することができます。ID ビューの情報は、Active Directory のユーザーオブジェクトの情報を上書きし、Active Directory に登録されているシステムや、Identity Management に登録されているシステムに異なるスマートカードのリンクを設定できるようになります。

Identity Management サーバーで以下の手順を実行します。

コマンドライン: Active Directory のユーザーアカウントとスマートカードのリンク

1. Identity Management の管理者としてログインします。

```
$ kinit admin
```

2. ユーザー証明書の環境変数 (**CERT**) を作成します。

```
$ CERT=`cat cert.pem | tail -n +2 | head -n -1 | tr -d '\r\n'`
```

3. 新規 ID オーバーライドを作成してし、ユーザー証明書を ID ビューに追加します。以下の手順では、Default Trust View を使用しています。

```
$ ipa idoverrideuser-add 'Default Trust View' ad_user@ad.example.com  
--certificate $CERT
```

Web UI: Active Directory のユーザーアカウントとスマートカードのリンク

1. **Identity → ID Views** を選択し、必要な ID ビューをクリックします。
2. 新規 ID オーバーライドを作成して、ユーザー証明書を ID ビューに追加します。**Add** をクリックして、**Add User ID override** フォームで必要な情報を入力します。

その他のリソース

- ID ビューの管理に関する詳細は「[18章/ID ビュー](#)」を参照してください。
- Default Trust View の詳細は「[Active Directory 環境での ID ビューの使用](#)」を参照してください。

23.1.2.4. アイデンティティマッピングの設定

セキュリティオフィサーは、ID 照合およびマッピングルールを使用して、ユーザーアカウントと 1 つまたは複数のユーザーロールアカウントの間のリンクを管理できます。これにより、物理的にスマートカードにアクセスできない場合など、ユーザーエントリーや ID オーバーライドにスマートカードの証明書を保存できない場合でさえも、ユーザーは Identity Management にアクセスできます。

アイデンティティマッピングの基本的な概要については、以下を参照してください。

- 「[Identity Management でのアイデンティティマッピング](#)」

アイデンティティマッピングの設定手順については、以下を参照してください。

- 「[証明書のアイデンティティマッピングルールの作成](#)」
- 「[ユーザーアカウントとスマートカード証明書のリンク](#)」

さまざまな例については以下を参照してください。

- 「[アイデンティティマッピングルールの例](#)」
- 「[証明書の発行者を照合ルールに変換する例](#)」

23.1.2.4.1. Identity Management でのアイデンティティマッピング

アイデンティティマッピングは、[アイデンティティマッピングルール](#) を作成して設定します。Identity Management は、[アイデンティティマッピングルール](#)の以下のコンポーネントをサポートします。すべてのコンポーネントはオプションです。

マッピングルール

マッピングルールは、証明書を 1 つ以上のユーザーアカウントに関連付けます (または マッピングします)。このルールで、対象のユーザーアカウントと証明書を関連付ける LDAP 検索フィルターを定義します。

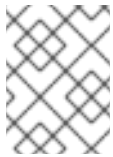
発行した証明局 (CA) が違う証明書には、異なるプロパティが設定されており、異なるドメインで使用される可能性があります。そのため、Identity Management は条件なしでマッピングルールを適用せず、適切な証明書にのみ適用します。照合ルールを使用して、適切な証明書を定義します。

照合ルール

照合ルールは、マッピングルールを適用する CA の証明書を選択します。

ドメイン一覧

ドメイン一覧は、アイデンティティマッピングルールを処理する際に Identity Management がユーザーを検索する DNS ドメイン名を指定します。



注記

ドメインが指定されていない場合は、Identity Management により、クライアントが所属するローカルドメインのユーザーのみが検索されます。

Priority

複数のルールが証明書に適用可能な場合には、最も優先順位の高いルールが先に適用され、他のルールはすべて無視されます。

- 数値が低いほど、アイデンティティマッピングの優先度が高くなります。たとえば、優先度が 1 のルールは、2 のルールより優先順位が高くなります。
- ルールに優先順位の値が定義されていない場合には、優先順位が最も低くなります。

23.1.2.4.2. 証明書のアイデンティティマッピングルールの作成

証明書のアイデンティティマッピングルールを作成すると、Identity Management により、スマートカードの証明書がユーザーアカウントに正しくマッピングされます。

コマンドライン: 証明書のアイデンティティマッピングルールの作成

サーバーで以下の手順を実行します。

1. 管理者としてログインします。

```
$ kinit admin
```

2. **ipa certmaprule-add** コマンドを使用してルールを作成します。アイデンティティマッピングルールのコンポーネントを指定するには、以下のオプションを使用します。

- **--maprule** はマッピングルールを定義します。
- **--matchrule** は照合ルールを定義します。
- **--domain** はユーザーエントリを検索するドメインを定義します。
- **--priority** はアイデンティティマッピングルールの優先順位を定義します。

たとえば、マッピングルールと照合ルールのみで構成される単純なアイデンティティマッピングルールを作成するには以下を実行します。

```
$ ipa certmaprule-add rule_name --matchrule '<ISSUER>CN=Smart Card
CA,0=EXAMPLE.ORG' --maprule '(ipacertmapdata=X509:<I>
{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})'
-----
Added Certificate Identity Mapping Rule "rule_name"
-----
Rule name: rule_name
Mapping rule: (ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})
Matching rule: <ISSUER>CN=Smart Card CA,0=EXAMPLE.ORG
Enabled: TRUE
```

このルールは、スマートカード証明書の件名や発行者を、ユーザーアカウントの **ipacertmapdata** 属性の値とリンクします。

Web UI: 証明書のアイデンティティマッピングルールの作成

1. **Authentication → Certificate Identity Mapping Rules** を選択します。
2. **Add** をクリックします。
3. ルールのコンポーネントの情報を入力して **Add** を追加します。

その他のリソース

- 証明書マッピングおよび照合ルールの構文に関する詳細は `sss-certmap(5) man` ページを参照してください。
- **ipa certmaprule-add** コマンドの使用方法に関する情報は **--help** オプションを指定してコマンドを実行してください。
- アイデンティティマッピングの管理に関する他のコマンドは、**ipa help certmap** コマンドを使用します。

23.1.2.4.3. ユーザーアカウントとスマートカード証明書のリンク

特定のユーザーアカウントとスマートカード証明書をリンクするには、**ipacertmapdata** 属性に証明書の件名と発行者を保存します。

コマンドライン: ユーザーアカウントとスマートカード証明書のリンク

- 証明書にアクセスできる場合には、完全な証明書のプロブを使用します。

```
$ CERT=`cat cert.pem | tail -n +2 | head -n -1 | tr -d '\r\n'`
$ ipa user-add-certmapdata idm_user --certificate $CERT
-----
Added certificate mappings to user "idm_user"
-----
User login: idm_user
Certificate mapping data: X509:<I>0=EXAMPLE.ORG,CN=Smart Card
CA<S>CN=test,0=EXAMPLE.ORG
```

- 証明書にアクセスできるが件名と発行者が分かっている場合には **--subject** および **--issuer** オプションを使用します。

```
$ ipa user-add-certmapdata idm_user --subject
"O=EXAMPLE.ORG,CN=test" --issuer "CN=Smart Card CA,O=EXAMPLE.ORG"
-----
Added certificate mappings to user "idm_user"
-----
User login: idm_user
Certificate mapping data: X509:<I>O=EXAMPLE.ORG,CN=Smart Card
CA<S>CN=test,O=EXAMPLE.ORG
```

- マッピングの形式に精通している場合には、マッピングデータを直接指定してください。

```
$ ipa user-add-certmapdata idm_user 'X509:<I>O=EXAMPLE.ORG,CN=Smart
Card CA<S>CN=test,O=EXAMPLE.ORG'
-----
Added certificate mappings to user "idm_user"
-----
User login: idm_user
Certificate mapping data: X509:<I>O=EXAMPLE.ORG,CN=Smart Card
CA<S>CN=test,O=EXAMPLE.ORG
```

スマートカードを使用して、Identity Management サーバーにログインできるようになりました。

Web UI: ユーザーアカウントとスマートカード証明書のリンク

1. **Identity** → **Users** をクリックして、対象のユーザーログインをクリックします。
2. **Certificate mapping data** エントリーの横にある **Add** をクリックします。

The screenshot shows a web interface for managing certificates. At the top, there is a 'Certificates' section with an 'Add' button. Below it, a 'Certificate mapping data' entry is highlighted with a red rectangular box, and it also has an 'Add' button next to it. Further down, there is a 'User authentication types' section with three options: 'Password', 'RADIUS', and 'Two factor authentication (password + OTP)', each with an unchecked checkbox.

図23.1 証明書マッピングデータの追加

3. **Add Certificate Mapping Data** フォームで、必要な情報を入力します。以下のいずれかを指定します。
 - **Certificate** で完全な証明書プロブを指定します。
 - **Issuer and subject** で件名と発行者を指定します。
 - **Certificate mapping data** でマッピングデータを直接指定します。

その他のリソース

- **ipa user-add-certmapdata** コマンドの詳細は **--help** オプションを指定してコマンドを実行してください。

23.1.2.4.4. アイデンティティーマッピングルールの例

例23.1 Identity Management ユーザーの Active Directory の証明書

```
$ ipa certmaprule-add ad_cert_for_ipa_users \
  --maprule='(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})' \
  --matchrule='<ISSUER>CN=AD-R00T-CA,DC=ad,DC=example,DC=com' \
  --domain=idm.example.com
```

例23.2 Active Directory ユーザーの Active Directory 証明書

```
$ ipa certmaprule-add ad_cert_for_ad_users \
  --maprule='(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>
{subject_dn!ad_x500})' \
  --matchrule='<ISSUER>CN=AD-R00T-CA,DC=ad,DC=example,DC=com' \
  --domain=ad.example.com
```

例23.3 Identity Management と Active Directory ユーザー両方の Active Directory 証明書

```
$ ipa certmaprule-add ad_cert_for_ipa_and_ad_users \
  --maprule='(|(ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>
{subject_dn!nss_x500})(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}
<S>{subject_dn!ad_x500}))' \
  --matchrule='<ISSUER>CN=AD-R00T-CA,DC=ad,DC=example,DC=com' \
  --domain=ad.example.com
```

上記の例では **--maprule** オプションの絞り込み定義には以下の条件が含まれます。

- **ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}** は、スマートカード証明書の件名および発行者と、Identity Management ユーザーアカウントの **ipacertmapdata** 属性値をリンクするフィルターです。
- **altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}** は、スマートカード証明書の件名および発行者と Active Directory ユーザーアカウントの **altSecurityIdentities** 属性値をリンクするフィルターです。

--maprule オプションのフィルター定義では、論理演算子 **|** (or) が使用できるので、複数の条件を指定できます。今回の場合、このルールは 1 つ以上の条件を満たす全ユーザーアカウントをマッピングします。

例23.4 Identity Management および Active Directory ユーザーの Identity Management 証明書

```
$ ipa certmaprule-add ipa_cert_for_ad_users \
  --maprule='(|(userCertificate;binary={cert!bin}))(ipacertmapdata=X509:
<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500})
(altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>
{subject_dn!ad_x500}))' \
  --matchrule='<ISSUER>CN=Certificate Authority,0=REALM.EXAMPLE.COM' \
  --domain=idm.example.com --domain=ad.example.com
```

上記の例では **--maprule** オプションの絞り込み定義には以下の条件が含まれます。

- **userCertificate;binary={cert!bin}** は、Identity Management または Active Directory のユーザーエントリ (全証明書を含む) を返すフィルターです。
- **ipacertmapdata=X509:<I>{issuer_dn!nss_x500}<S>{subject_dn!nss_x500}** は、スマートカード証明書の件名および発行者と、Identity Management ユーザーアカウントの **ipacertmapdata** 属性値をリンクするフィルターです。
- **altSecurityIdentities=X509:<I>{issuer_dn!ad_x500}<S>{subject_dn!ad_x500}** は、スマートカード証明書の件名および発行者と Active Directory ユーザーアカウントの **altSecurityIdentities** 属性値をリンクするフィルターです。

--maprule オプションのフィルター定義では、論理演算子 | (or) が使用できるので、複数の条件を指定できます。今回の場合、このルールは 1 つ以上の条件を満たす全ユーザーアカウントをマッピングします。

23.1.2.4.5. 証明書の発行者を照合ルールに変換する例

照合ルールに必要な発行者の形式を取得するには、パスを逆方向から指定して、区切り文字 (/) をコンマに置き換えます。

例23.5 Identity Management が発行する証明書の発行者の変換

Identity Management の発行する証明書の例

```
# openssl x509 -in user.crt -noout -issuer
issuer= /O=REALM.EXAMPLE.COM/CN=Certificate Authority
```

照合ルールで規定の形式で表現したこの証明書の発行者:

```
'<ISSUER>CN=Certificate Authority,0=REALM.EXAMPLE.COM'
```

例23.6 メールを含む証明書の発行者の変換

メールを含む証明書の例

```
# openssl x509 -in expired_user.pem -noout -issuer
issuer= /C=US/ST=North Carolina/L=Raleigh/O=Red
Hat/OU=QE/CN=ExampleCA/emailAddress=admin@example.com
```

照合ルールで規定の形式で表現したこの証明書の発行者:

```
'<ISSUER>emailAddress=admin@example.com,CN=ExampleCA,OU=QE,O=Red
Hat,L=Raleigh,ST=North Carolina,C=US'
```

23.1.2.5. その他のリソース

- スマートカードの証明書のリンクを検証するには「[指定の証明書と合致するユーザーの検索](#)」を参照してください。
- 証明書のアイデンティティマッピングに関する詳しい情報は、アップストリームの SSSD ドキュメントの「[Matching and Mapping Certificates](#)」を参照してください。

23.1.3. 指定の証明書と合致するユーザーの検索

ロールアカウントと証明書が合致する全従業員を表示するには、Identity Management サーバーと従業員のスマートカード証明書を提示します。

Identity Management システムで以下の手順を実行します。

コマンドライン: 指定の証明書と合致するユーザーの検索

1. 管理者としてログインします。

```
$ kinit admin
```

2. ユーザーを検索するには、以下のいずれかを指定します。

- 証明書ファイルの名前:

```
$ ipa certmap-match cert.pem
-----
1 user matched
-----
Domain: IDM.EXAMPLE.COM
User logins: idm_user
-----
Number of entries returned 1
-----
```

- 証明書の内容:

```
$ ipa certmap-match --certificate="MII...."
-----
1 user matched
-----
Domain: IDM.EXAMPLE.COM
User logins: idm_user
-----
Number of entries returned 1
-----
```

ユーザーエントリーに完全な証明書のブロブが含まれる場合には、このコマンドは、信頼される Active Directory ドメインでユーザーも返します。

```
$ ipa certmap-match --certificate="MII...."
-----
2 users matched
-----
Domain: ad.domain.com
User logins: ad_user

Domain: IDM.EXAMPLE.COM
User logins: idm_user
-----
Number of entries returned 2
-----
```

Web UI: 指定の証明書と合致するユーザーの検索

1. **Authentication** → **Certificate Identity Mapping Rules** → **Certificate Mapping Match** をクリックします。
2. **Certificate** フィールドの証明書の内容を入力して **Match** をクリックします。
Identity Management は、**Matched Users** に証明書と合致するユーザーを表示します。

Matched Users	
User Login	Domain
user	EXAMPLE.COM
1 user matched	

図23.2 証明書と合致するユーザーの表示

その他のリソース

- 証明書のアイデンティティマッピングのコマンドに関する詳細は、**ipa help certmap** コマンドを使用します。
- **ipa certmap-match** コマンドの詳細は **--help** オプションを指定してコマンドを実行してください。

23.1.4. その他のリソース

- Red Hat Certificate System のアプリケーション、Enterprise Security Client を使用して個人の証明書やキーを管理する情報は、Certificate System ドキュメントの「[Managing Smart Cards with the Enterprise Security Client](#)」を参照してください。

23.2. スマートカードで IDENTITY MANAGEMENT クライアントへ認証する方法

Identity Management サーバーで複数のロールアカウントが指定された Identity Management ユーザーとして、スマートカードを使用して、Identity Management ドメインに参加するデスクトップクライアントシステムに対して認証を行うことができます。

サポートされているオプションの基本的な概要については、以下を参照してください。

- [「Identity Management クライアントでサポートされるスマートカード認証オプション」](#)

認証が有効な環境を設定するための情報は、以下を参照してください。

- [「スマートカード認証に向けた Identity Management の準備」](#)

認証方法に関する情報は、以下を参照してください。

- [「コンソールログインを使用してスマートカードで Identity Management クライアントへ認証する方法」](#)
- [「SSH を使用してスマートカードで Identity Management クライアントへ認証する方法」](#)

23.2.1. Identity Management クライアントでサポートされるスマートカード認証オプション

Identity Management のユーザーは、Identity Management クライアントでスマートカードを使用して認証する際に、以下のオプションを使用できます。

ローカル認証

ローカル認証には、以下を使用した認証が含まれます。

- テキストコンソール
- Gnome Display Manager (GDM) などのグラフィカルコンソール
- **su** または **sudo** などのローカル認証サービス

ssh でのリモート認証

スマートカードの証明書は、PIN で保護される SSH の秘密鍵と合わせて保存されます。

FTP などの他のサービスを使用するスマートカードベースの認証はサポートされていません。

23.2.2. スマートカード認証に向けた Identity Management の準備

Identity Management の管理者として、以下の手順を実行します。

1. サーバーで、shell スクリプトを作成してクライアントを設定します。
 - a. **ipa-adviser config-client-for-smart-card-auth** コマンドを使用して出力をファイルに保存します。

```
# ipa-adviser config-client-for-smart-card-auth >  
client_smart_card_script.sh
```

- b. スクリプトファイルを開き、内容を確認します。
- c. **chmod** ユーティリティーを使用して実行パーミッションをファイルに追加します。

```
# chmod +x client_smart_card_script.sh
```

2. スクリプトをクライアントにコピーして実行します。スマートカード証明書を署名した認証局 (CA) を含む PEM ファイルへのパスを追加します。

```
# ./client_smart_card_script.sh CA_cert.pem
```

さらに、外部の証明局 (CA) がスマートカードの証明書を署名した場合に、スマートカード CA を信頼された CA として追加します。

1. Identity Management サーバーで CA 証明書をインストールします。

```
# ipa-cacert-manage -n "SmartCard CA" -t CT,C,C install ca.pem
# ipa-certupdate
```

すべてのレプリカおよびクライアントでも **ipa-certupdate** を繰り返します。

2. HTTP サーバーを再起動します。

```
# systemctl restart httpd
```

すべてのレプリカでも **systemctl restart httpd** を実行します。



注記

SSSD では、管理者は **certificate_verification** パラメーターで証明書の検証プロセスを調節できます。たとえば、証明書に定義されている Online Certificate Status Protocol (OCSP) サーバーは、クライアントから到達できません。詳しい情報は `sssd.conf(5)` の man ページを参照してください。

23.2.3. コンソールログインを使用してスマートカードで Identity Management クライアントへ認証する方法

Identity Management ユーザーとして認証するには、ユーザー名と PIN を入力します。

- コマンドラインからログインする場合:

```
client login: idm_user
PIN for PIV Card Holder pin (PIV_II) for user
idm_user@idm.example.com:
```

- Gnome Desktop Manager (GDM) を使用してログインすると、GDM により、所定のユーザー名を選択した後にスマートカードと PIN が求められます。

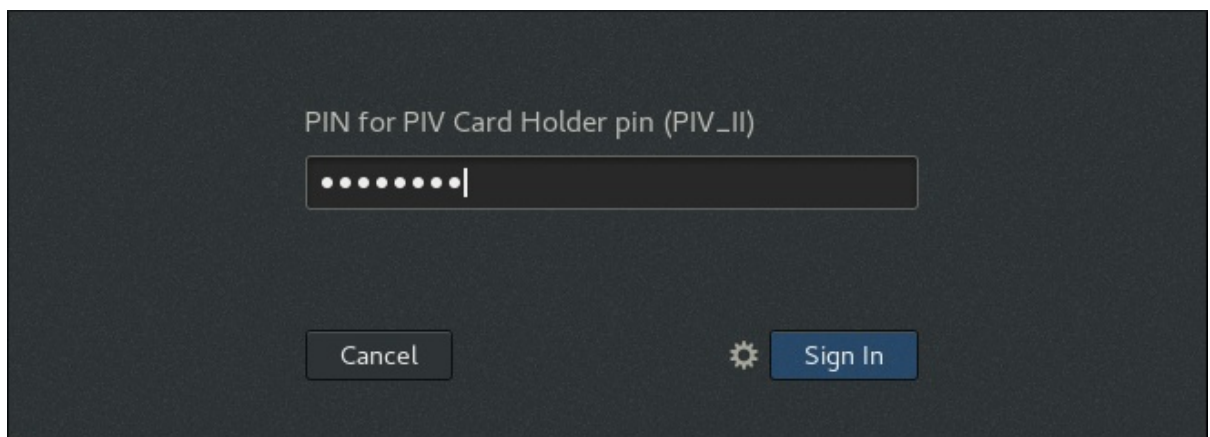


図23.3 Gnome Desktop Manager でのスマートカードの PIN の入力

Active Directory ユーザーとして認証するには、**AD.EXAMPLE.COM\ad_user** または **ad_user@AD.EXAMPLE.COM** など、NetBIOS ドメイン名を使用する形式でユーザー名を入力します。

認証が失敗した場合には、「[スマートカード認証の失敗](#)」を参照してください。

23.2.4. SSH を使用してスマートカードで **Identity Management** クライアントへ認証する方法

ssh ユーティリティーを使用する場合に、スマートカードのレンダリングモジュールへのパスを指定します。以下に例を示します。

```
$ ssh -I /usr/lib/libmypkcs11.so -l user@example.com host.example.com
Enter PIN for 'Smart Card':
```

認証が失敗した場合には、「[スマートカード認証の失敗](#)」を参照してください。

23.2.5. その他のリソース

- OpenSSH でのスマートカード認証の詳細は、『セキュリティガイド』の「[OpenSSH に認証情報を提供するスマートカードの使用](#)」を参照してください。

23.3. スマートカードを使用してリモートで **IDENTITY MANAGEMENT** システムに対する認証を行う方法

Identity Management サーバーに複数のロールアカウントを持つ Identity Management ユーザーとして、**ssh** ユーティリティーを使用して (Identity Management ドメインに登録されていない) ローカルシステムから (Identity Management ドメインに登録されている) リモートシステムにスマートカードで認証を行うことができます。これにより、選択したロールでリモートシステムを使用できるようになります。

認証が有効な環境を設定するための情報は、以下を参照してください。

- 「[スマートカード認証用のローカルシステムの準備](#)」
- 「[スマートカード認証用のリモートの Identity Management システムの準備](#)」
- 「[Active Directory のユーザーエントリーとスマートカード証明書のリンク](#)」

認証方法に関する情報は、以下を参照してください。

- 「[ローカルシステムからリモートシステムへの認証](#)」

23.3.1. スマートカード認証用のローカルシステムの準備

管理者として、ローカルシステムで以下の手順を実行します。

1. opensc パッケージをインストールします。

```
# yum install opensc
```

2. スマートカードデーモンの **pcscd** サービスが起動され、有効であることを確認します。

```
# systemctl start pcsd.socket pcsd.service
# systemctl enable pcsd.socket pcsd.service
```

さらに、外部の証明局 (CA) がスマートカードの証明書を署名した場合に、スマートカード CA を信頼された CA として追加します。

1. Identity Management サーバーで CA 証明書をインストールします。

```
# ipa-cacert-manage -n "SmartCard CA" -t CT,C,C install ca.pem
# ipa-certupdate
```

すべてのレプリカおよびクライアントでも **ipa-certupdate** を繰り返します。

2. Identity Management サーバーの HTTP サーバーを再起動します。

```
# systemctl restart httpd
```

すべてのレプリカでも **systemctl restart httpd** を実行します。

23.3.2. スマートカード認証用のリモートの Identity Management システムの準備

管理者として以下の手順を実行します。

1. リモートシステム上の **/etc/pki/nssdb/** データベースにスマートカード証明局 (CA) の証明書をインストールします。

```
# certutil -A -d /etc/pki/nssdb/ -n "SmartCard CA" -t CT,C,C -i
ca.pem
```

2. sssd-dbus パッケージがインストールされていることを確認します。

23.3.3. Active Directory のユーザーエントリーとスマートカード証明書のリンク

ユーザーエントリーが Active Directory に保存されている場合には、管理者は必ずそのエントリーとスマートカード証明書をリンクする必要があります。[「Active Directory のユーザーアカウントとスマートカードのリンク」](#)を参照してください。

23.3.4. ローカルシステムからリモートシステムへの認証

ローカルシステムで以下の手順を実行します。

1. スマートカードを挿入します。
2. **ssh** を起動して **-I** オプションで PKCS#11 ライブラリーを指定します。
 - Identity Management ユーザーとして以下を実行します。

```
$ ssh -I /usr/lib64/opensc-pkcs11.so -l idm_user
server.idm.example.com
```

```
Enter PIN for 'PIV_II (PIV Card Holder pin)':
Last login: Thu Apr 6 12:49:32 2017 from 10.36.116.42
```

- Active Directory ユーザーとして以下を実行します。

```
$ ssh -I /usr/lib64/opensc-pkcs11.so -l ad_user@ad.example.com
server.idm.example.com

Enter PIN for 'PIV_II (PIV Card Holder pin)':
Last login: Thu Apr 6 12:49:32 2017 from 10.36.116.42
```

3. オプション: **id** ユーティリティーを使用して所定のユーザーとしてログインしていることを確認します。

- Identity Management ユーザーとして以下を実行します。

```
$ id
uid=1928200001(idm_user) gid=1928200001(idm_user)
groups=1928200001(idm_user)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

- Active Directory ユーザーとして以下を実行します。

```
$ id
uid=1171201116(ad_user@ad.example.com)
gid=1171201116(ad_user@ad.example.com)
groups=1171201116(ad_user@ad.example.com),1171200513(domain
users@ad.example.com)
context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

認証が失敗した場合には、「[スマートカード認証の失敗](#)」を参照してください。

23.3.5. その他のリソース

- **ssh** を使用してスマートカードで認証を行うと、リモートシステムの TGT (Ticket Granting Ticket) は取得されません。リモートシステムで TGT を取得するには、管理者はローカルシステムに Kerberos を設定して、Kerberos の委譲を有効化する必要があります。所定の設定例は、「[this Kerberos knowledge base entry](#)」を参照してください。
- OpenSSH でのスマートカード認証の詳細は、『セキュリティガイド』の「[OpenSSH に認証情報を提供するスマートカードの使用](#)」を参照してください。

23.4. スマートカード認証のユーザー名ヒントのポリシー設定

Identity Management の管理者として、複数のアカウントにリンクされているスマートカードに **ユーザー名のヒント** ポリシーを設定することができます。

23.4.1. Identity Management でのユーザー名のヒント

ユーザー名ヒントポリシーでは、スマートカードユーザーにユーザー名を尋ねるように Identity Management を設定します。ユーザーが Identity Management の複数のユーザーアカウントに合致するスマートカード証明書で認証しようとする、以下のいずれかが発生します。

- ユーザー名ヒントポリシーが有効な場合には、ユーザーはユーザー名を求められ、認証に進むことができます。
- ユーザー名品とポリシーが無効な場合には、プロンプトが表示されず、認証に失敗します。

Identity Management は、デフォルトでユーザー名を聞かずにスマートカードの PIN の入力を求めるアプリケーションにユーザー名ヒントを追加します。Red Hat Enterprise Linux では、現在 Gnome Desktop Manager (GDM) ログインのみで対応しています。

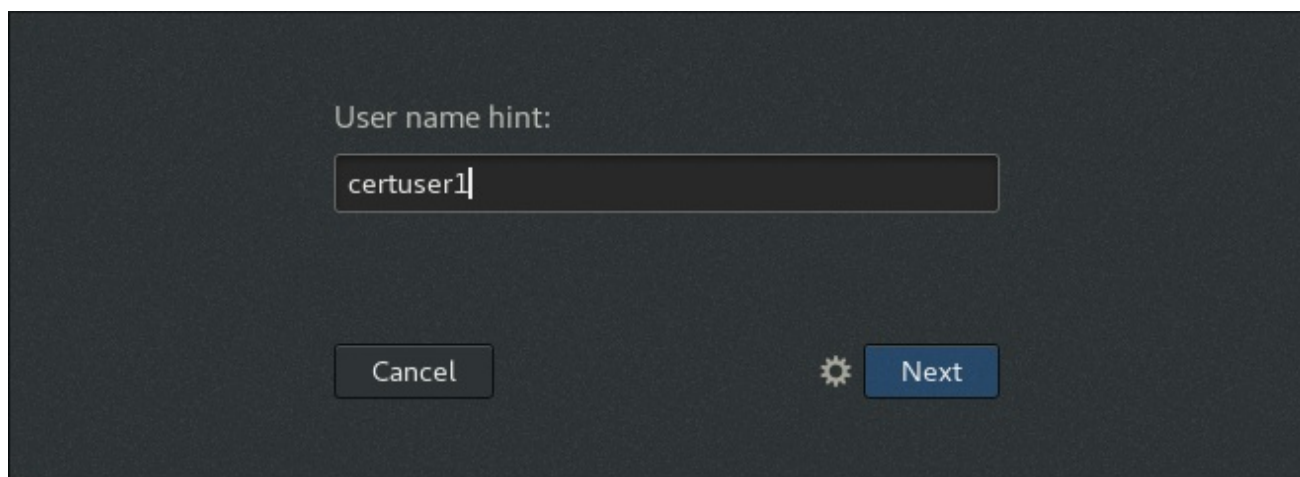


図23.4 Gnome Desktop Manager でのユーザー名のヒント

Identity Management は、以下のようにデフォルトでユーザー名を確認するアプリケーションにはユーザー名のヒントを追加しません。

- Identity Management の Web UI 認証。GUI で **Username** フィールドが常に表示されるため。
- **ssh** 認証。**ssh** は現在のユーザーのログイン名または **-l** オプションや **username@host** 形式で指定される名前を使用するため。
- コンソールの認証。ログイン名を常に指定するため。

このような場合に、複数のユーザーが合致する証明書での認証は常に許可されます。

23.4.2. Identity Management でのユーザー名ヒントの有効化

Identity Management の管理者は、ユーザー名ヒントのポリシーを集約的に設定します。ポリシーは、Identity Management に登録された全ホストに適用されます。

Identity Management システムで以下の手順を実行します。

コマンドライン: Identity Management でのユーザー名ヒントの有効化

1. Identity Management の管理者としてログインします。

```
$ kinit admin
Password for admin@IDM.EXAMPLE.COM:
```

2. **--promptusername=True** オプションを指定して **ipa certmapconfig-mod** コマンドを使用して、ユーザー名ヒントを有効にします。

```
$ ipa certmapconfig-mod --promptusername=TRUE
Prompt for the username: TRUE
```

ユーザー名ヒントを無効にするには **--promptusername=False** オプションを使用します。

Web UI: Identity Management でのユーザー名ヒントの有効化

1. **Authentication → Certificate Identity Mapping Rules → Certificate Identity Mapping Global Configuration** をクリックします。
2. **Prompt for the username** を選択し、**Save** をクリックします。

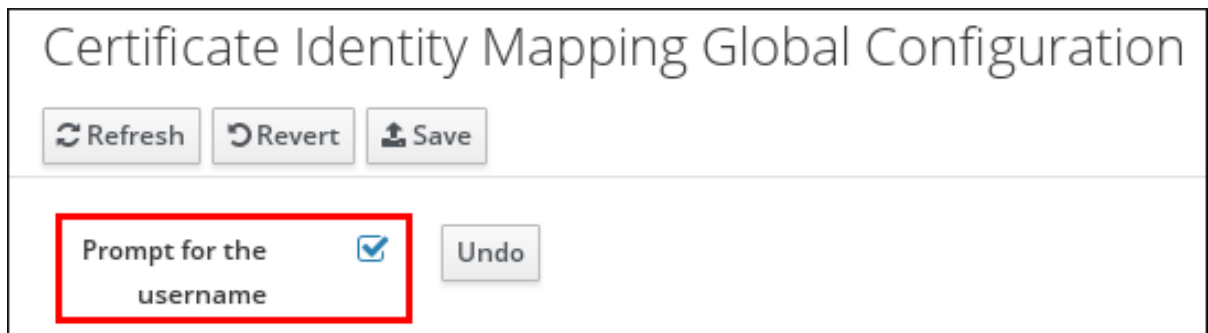


図23.5 Web UI でのユーザー名ヒントの有効化

その他のリソース

- **ipa certmapconfig-mod** コマンドに関する詳細は、**--help** オプションを指定して実行します。

23.5. IDENTITY MANAGEMENT での PKINIT スマートカード認証

Identity Management ユーザーは、スマートカードを使用して、Identity Management に登録されているデスクトップクライアントシステムに対して認証を行い、自動的に TGT (Ticket-Granting Ticket) を取得することができます。ユーザーは、クライアントからチケットを使用して、さらにシングルサインオン (SSO) 認証を行うことができます。

23.5.1. PKINIT 認証に向けた Identity Management クライアントの準備

Identity Management の管理者として、ユーザー認証を行うクライアントマシンで以下の手順を実行します。

1. サーバーで、shell スクリプトを作成してクライアントを設定します。
 - a. **ipa-adviser config-client-for-smart-card-auth** コマンドを使用して出力をファイルに保存します。

```
# ipa-adviser config-client-for-smart-card-auth >
client_smart_card_script.sh
```

- b. スクリプトファイルを開き、内容を確認します。
- c. **chmod** ユーティリティーを使用して実行パーミッションをファイルに追加します。

```
# chmod +x client_smart_card_script.sh
```

2. スクリプトをクライアントにコピーして実行します。スマートカード証明書を署名した認証局 (CA) を含む PEM ファイルへのパスを追加します。

```
# ./client_smart_card_script.sh CA_cert.pem
```

3. krb5-pkinit パッケージがインストールされていることを確認します。

さらに、外部の証明局 (CA) がスマートカードの証明書を署名した場合に、スマートカード CA を信頼された CA として追加します。

1. Identity Management サーバーで CA 証明書をインストールします。

```
# ipa-cacert-manage -n "SmartCard CA" -t CT,C,C install ca.pem
# ipa-certupdate
```

すべてのレプリカおよびクライアントでも **ipa-certupdate** を繰り返します。

2. HTTP サーバーを再起動します。

```
# systemctl restart httpd
```

すべてのレプリカでも **systemctl restart httpd** を実行します。



注記

SSSD では、管理者は **certificate_verification** パラメーターで証明書の検証プロセスを調節できます。たとえば、証明書に定義されている Online Certificate Status Protocol (OCSP) サーバーは、クライアントから到達できません。詳しい情報は `sssd.conf(5)` の man ページを参照してください。

23.5.2. Identity Management ユーザー: Identity Management クライアントでの PKINIT を使用した認証

Identity Management クライアントで **kinit** ユーティリティを使用した認証:

```
$ kinit -X X509_user_identity='PKCS11:opensc-pkcs11.so' idm_user
```

-X オプションは、事前認証の属性として **opensc-pkcs11.so** モジュールを指定します。詳しい情報は、`kinit(1)` の man ページを参照してください。

23.5.3. Active Directory ユーザー: Identity Management クライアントでの PKINIT を使用した認証

前提条件

管理者として、環境が Active Directory ユーザーの PKINIT 認証をサポートするように設定します。

- Active Directory サーバーが、スマートカード証明書を発行する認証局 (CA) を信頼するように設定します。NTAuth ストアに CA をインポートして ([Microsoft サポート](#) を参照)、信頼された CA として CA を追加します。詳細は Active Directory ドキュメントを参照してください。
- スマートカード証明書を発行する CA を信頼するように Kerberos クライアントを設定します。

1. Identity Management クライアントで **/etc/krb5.conf** ファイルを開きます。

2. ファイルに以下の行を追加します。

```
[libdefaults]
```

```
[... file truncated ...]
pkinit_eku_checking = kpServerAuth
pkinit_kdc_hostname = adserver.ad.domain.com
```

- ユーザー証明書に、証明書失効リスト (CRL: Certificate Revocation List) の Distribution Point Extension が含まれていない場合には、Active Directory が失効エラーを無視するように設定します。
1. 以下の REG 形式の内容をプレーンテキストファイルに保存して、ファイルをダブルクリックして Windows レジストリーにインポートします。

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc]
"UseCachedCRLOnlyAndIgnoreRevocationUnknownErrors"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA\Kerberos\Parameters]
"UseCachedCRLOnlyAndIgnoreRevocationUnknownErrors"=dword:00000001
```

または、**regedit.exe** アプリケーションを使用して手動で値を設定します。

2. Windows システムを再起動して変更を適用します。

手順

Identity Management クライアントで **kinit** ユーティリティを使用して認証し、ユーザーとドメイン名で Active Directory ユーザーを指定します。

```
$ kinit -X X509_user_identity='PKCS11:opensc-pkcs11.so'
ad_user@ad.domain.com
```

-X オプションは、事前認証の属性として **opensc-pkcs11.so** モジュールを指定します。詳しい情報は、**kinit(1)** の man ページを参照してください。

23.6. スマートカードを使用して IDENTITY MANAGEMENT WEB UI への認証を行う方法

Identity Management サーバーで複数のロールアカウントが指定された Identity Management ユーザーとして、選択したロールでスマートカードを使用して、Identity Management Web UI に対して認証を行うことができます。このように、選択したロールで Web UI を使用できるようになります。



注記

Identity Management ユーザーのみが、スマートカードで Web UI にログイン できます。詳細は、[「AD ユーザーとしての IdM Web UI への認証」](#)を参照してください。

認証が有効な環境を設定するための情報は、以下を参照してください。

- [「Web UI でのスマートカード認証に向けた Identity Management サーバーの準備」](#)
- [「スマートカード認証用のブラウザーの準備」](#)

認証方法に関する情報は、以下を参照してください。

- 「Identity Management ユーザーとしてスマートカードを使用して Identity Management Web UI への認証を行う方法」

23.6.1. Web UI でのスマートカード認証に向けた Identity Management サーバーの準備

Identity Management の管理者として:

1. Identity Management サーバーで、shell スクリプトを作成してサーバーを設定します。
 - a. **ipa-adviser config-server-for-smart-card-auth** コマンドを使用して出力をファイルに保存します。

```
# ipa-adviser config-server-for-smart-card-auth >  
server_smart_card_script.sh
```

- b. スクリプトファイルを開き、内容を確認します。
- c. **chmod** ユーティリティーを使用して実行パーミッションをファイルに追加します。

```
# chmod +x server_smart_card_script.sh
```

2. Identity Management ドメインの全サーバー上でスクリプトを実行します。
3. sssd-dbus パッケージがインストールされていることを確認します。

さらに、外部の証明局 (CA) がスマートカードの証明書を署名した場合:

1. Identity Management サーバーで、CA 証明書を、HTTP サーバーが使用する NSS データベースに追加します。

```
# ipa-cacert-manage -n "SmartCard CA" -t CT,C,C install ca.pem  
# ipa-certupdate
```

すべてのレプリカおよびクライアントでも **ipa-certupdate** を繰り返します。

2. HTTP サーバーと Kerberos サーバーを再起動します。

```
# systemctl restart httpd  
# systemctl restart krb5kdc
```

すべてのレプリカでこのコマンドを繰り返し実行します。

23.6.2. スマートカード認証用のブラウザーの準備

スマートカード用にブラウザーを設定するには、Web UI にアクセスする Web ブラウザーを起動するクライアントで以下の手順を実行します。ブラウザーが実行されるシステムは、Identity Management ドメインに所属する必要はありません。以下の手順では、Firefox ブラウザーを使用します。

1. Firefox を起動します。
2. スマートカードから証明書が読み取られるように Firefox を設定します。

- a. **Edit** → **Preferences** → **Advanced** → **Certificates** → **Security Devices** を選択します。

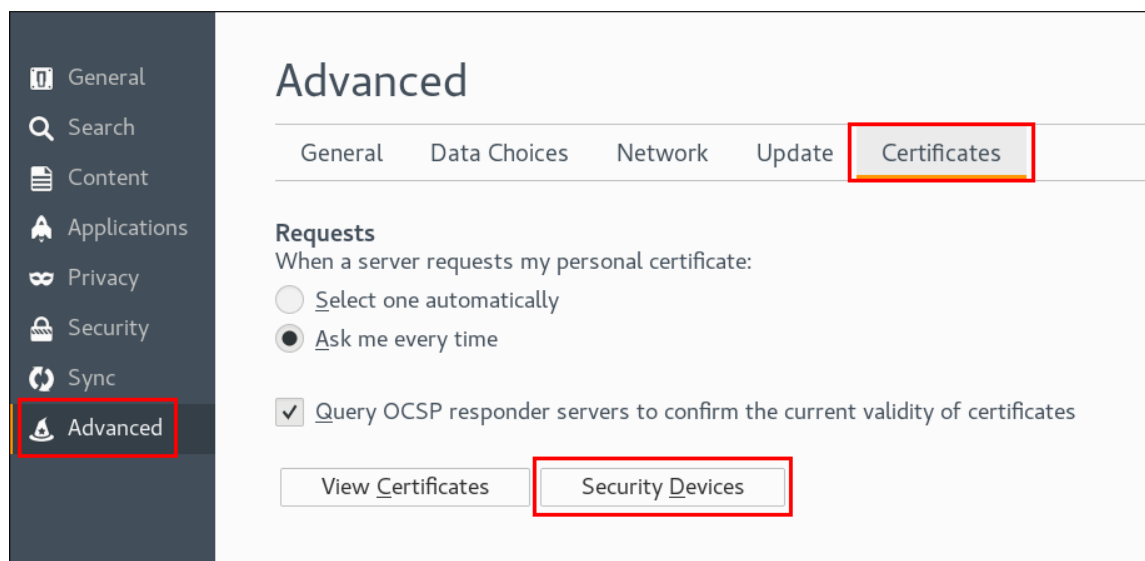


図23.6 Firefox でのセキュリティーデバイスの設定

- b. **Load** をクリックします。Load PKCS#11 Device ウィンドウで以下の情報を入力します。

- **Module Name:** OpenSC
- **Module filename:** /usr/lib64/opensc-pkcs11.so

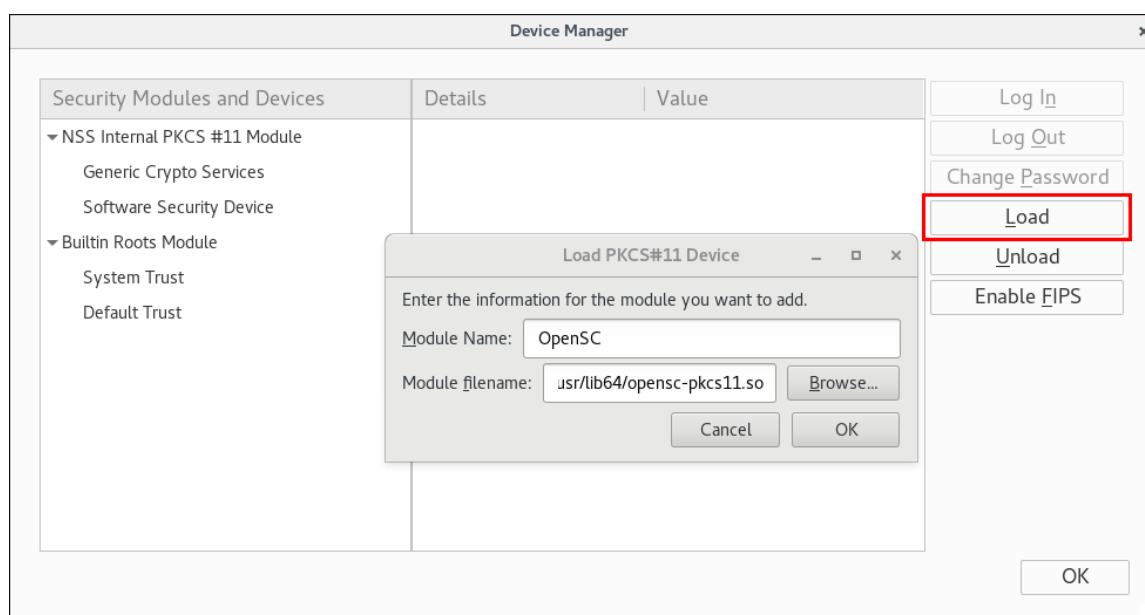


図23.7 Firefox のデバイスマネージャー

- c. **OK** をクリックして確定してから、再度 **OK** をクリックしてデバイスマネージャーを終了します。

Firefox で、認証にスマートカード証明書が使用できるようになりました。

23.6.3. Identity Management ユーザーとしてスマートカードを使用して Identity Management Web UI への認証を行う方法

認証方法:

1. スマートカードをスマートカードリーダーに挿入します。
2. ブラウザーで Identity Management Web UI (<https://ipaserver.example.com/ipa/ui>) に移動します。
3. スマートカード証明書が単一のユーザーアカウントにリンクされている場合には、**Username** フィールドには入力しないでください。

スマートカード証明書が複数のユーザーアカウントにリンクされている場合には **Username** フィールドに入力して所定のアカウントにを指定します。

4. **Login Using Certificate** をクリックします。

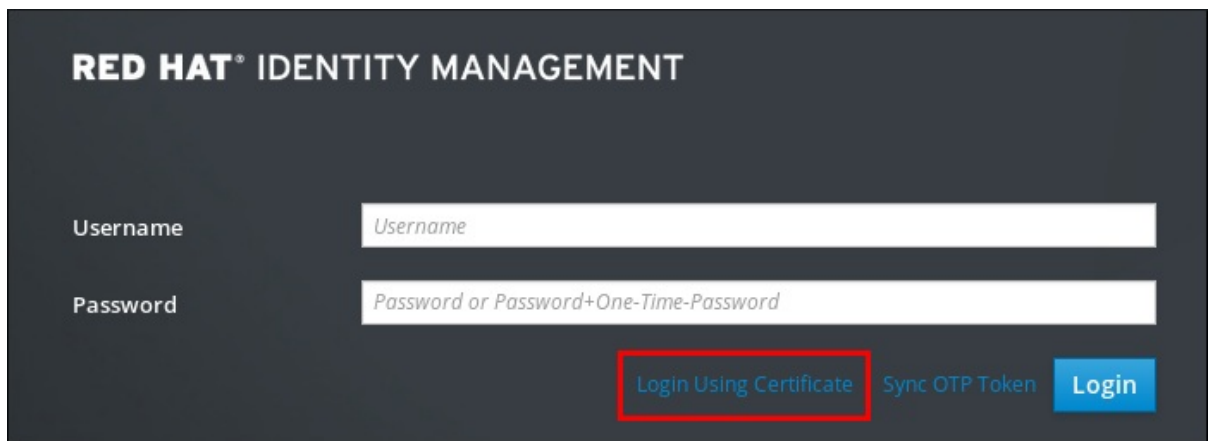


図23.8 Identity Management Web UI の Login Using Certificate

5. プロンプトが表示されたらスマートカードの情報を入力します。

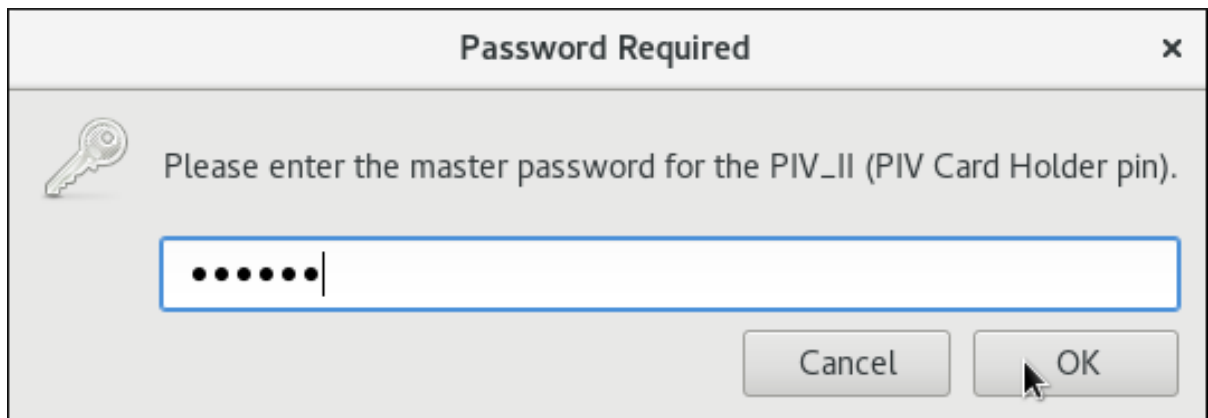


図23.9 スマートカード PIN の入力

6. 新しいウィンドウが開き、使用する証明書を提示するように求められます。スマートカード証明書を選択します。

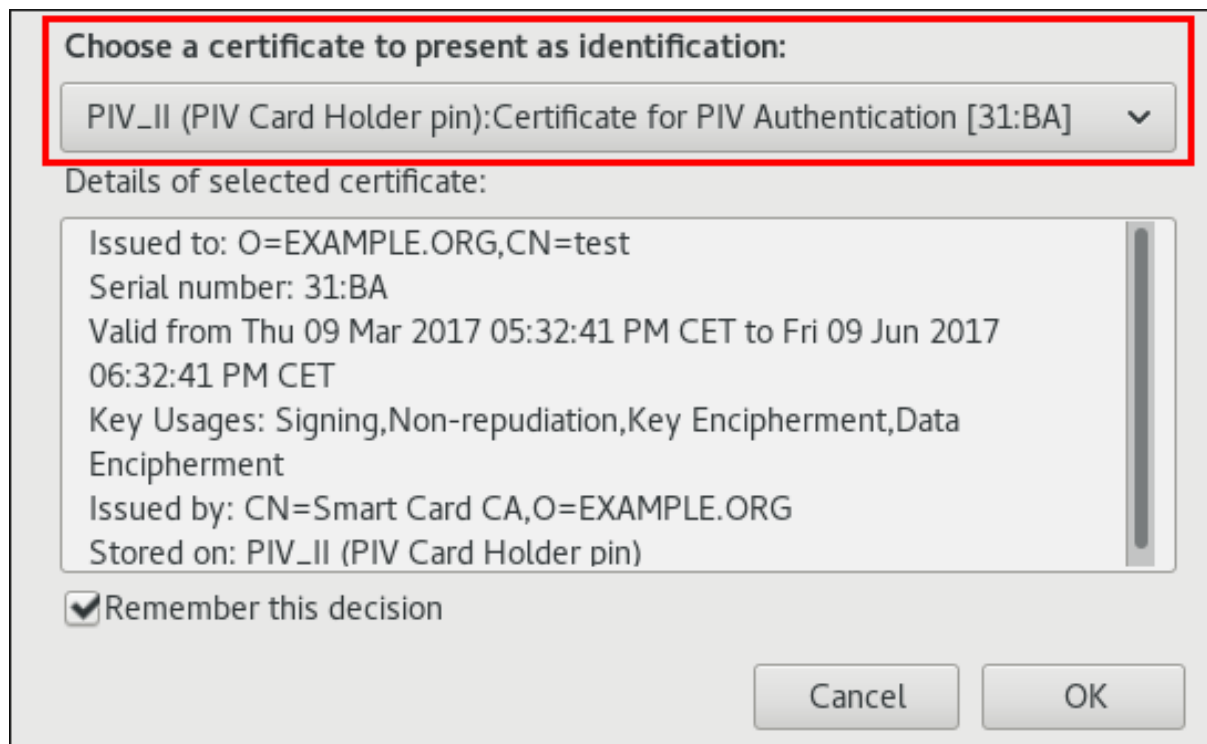


図23.10 スマートカード証明書の選択

スマートカード証明書に対応するユーザーとして認証されました。

その他のリソース

- 認証が失敗した場合には、「[スマートカード認証の失敗](#)」を参照してください。

23.6.4. その他のリソース

- Identity Management の Web UI の詳細は「[IdM Web UI](#)」を参照してください。

23.7. WEB アプリケーションと IDENTITY MANAGEMENT のスマートカード認証の統合

開発者は、Identity Management Web インフラストラクチャーの Apache モジュール経由で、認証バックエンドとして Identity Management サーバーを使用するアプリケーションでは、スマートカードに複数のロールアカウントがリンクされているユーザーを認証できるようにアプリケーションを設定することができます。

23.7.1. スマートカードでの **Web** アプリケーション認証における前提条件

Apache Web アプリケーションが実行中のサーバー:

- Identity Management ドメインのクライアントとしてサーバーを登録します。
- `sssd-dbus` および `mod_lookup_identity` パッケージをインストールします。
- Apache に、`mod_nss` モジュールを使用して、機能する HTTPS 接続が設定されていることを確認します。

23.7.2. Web アプリケーション向けの Identity Management スマートカード認証の設定

1. `/etc/httpd/conf.d/nss.conf` ファイルの `mod_nss` 設定で TLS ネゴシエーションを有効化します。

```
NSSRenegotiation
NSSRequireSafeNegotiation on
```

2. ユーザー証明書を発行する CA が `mod_nss` 証明書データベースのクライアント証明書向けに信頼されていることを確認します。データベースのデフォルトの場所は `/etc/httpd/alias` です。
3. Web アプリケーションを追加します。この手順では、ログインページおよび保護エリアだけが含まれる最小限のアプリケーションを例として使用します。
 - `/login` エンドポイントでは、ユーザーはユーザー名のみが指定でき、アプリケーションの保護エリアに送信されます。
 - `/app` エンドポイントは `REMOTE_USER` 環境変数をチェックします。ログインが成功すると、変数にはログインユーザーの ID が含まれます。ログインに成功しなかった場合には、変数は設定されません。
4. ディレクトリーを作成して、グループを `apache` に、モードを最低でも `750` に設定します。この手順では、`/var/www/app/` という名前のディレクトリーを使用します。
5. ファイルを作成して、グループを `apache` に、モードを最低でも `750` に設定します。この手順では、`/var/www/app/login.py` という名前のファイルを使用します。

以下の内容をファイルに保存します。

```
#!/usr/bin/env python

def application(environ, start_response):
    status = '200 OK'
    response_body = """
<!DOCTYPE html>
<html>
  <head>
    <title>Login</title>
  </head>
  <body>
    <form action='/app' method='get'>
      Username: <input type='text' name='username'>
      <input type='submit' value='Login with certificate'>
    </form>
  </body>
</html>
"""
    response_headers = [
        ('Content-Type', 'text/html'),
        ('Content-Length', str(len(response_body)))
    ]
    start_response(status, response_headers)
    return [response_body]
```

6. ファイルを作成して、グループを `apache` に、モードを最低でも `750` に設定します。この手順では、`/var/www/app/protected.py` という名前のファイルを使用します。

以下の内容をファイルに保存します。

```
#!/usr/bin/env python

def application(environ, start_response):
    try:
        user = environ['REMOTE_USER']
    except KeyError:
        status = '400 Bad Request'
        response_body = 'Login failed.\n'
    else:
        status = '200 OK'
        response_body = 'Login succeeded. Username:
        {}\n'.format(user)

    response_headers = [
        ('Content-Type', 'text/plain'),
        ('Content-Length', str(len(response_body)))
    ]
    start_response(status, response_headers)
    return [response_body]
```

7. アプリケーション用の設定ファイルを作成します。この手順では、**/etc/httpd/conf.d/app.conf** という名前のファイルを使用し、以下のコンテンツを追加します。

```
<IfModule !lookup_identity_module>
    LoadModule lookup_identity_module modules/mod_lookup_identity.so
</IfModule>

WSGIScriptAlias /login /var/www/app/login.py
WSGIScriptAlias /app /var/www/app/protected.py

<Location "/app">
    NSSVerifyClient require
    NSSUserName SSL_CLIENT_CERT
    LookupUserByCertificate On
    LookupUserByCertificateParamName "username"
</Location>
```

このファイルでは、以下が設定されます。

- 最初の部分では、すでに読み込まれていない場合には **mod_lookup_identity** を読み込みます。
- 次の部分では **/login** および **/app** エンドポイントを適切な Web Server Gateway Interface (WSGI) スクリプトにマッピングします。
- 最後の部分では、TLS ハンドシェイク時にクライアントの証明書が必要でそれを使用できるように **/app** エンドポイントの **mod_nss** を設定します。さらに、オプションの要求パラメーター **username** がユーザーの ID を検索するように設定します。

第24章 ユーザー、ホスト、およびサービス向け証明書の管理

Identity Management (IdM) では、以下の 2 つのタイプの証明局 (CA) をサポートしています。

統合 IdM CA

統合 CA は、ユーザー、ホスト、およびサービス向けの証明書を作成、取り消し、発行することができます。詳細は「[統合 IdM CA での証明書の管理](#)」を参照してください。

IdM 軽量サブ CA の作成に対応しています。詳細は「[軽量のサブ証明局 \(CA\)](#)」を参照してください。

外部 CA

外部 CA は、統合 IdM CA 以外の CA です。

IdM ツールを使ってこの CA が発行した証明書をユーザー、サービス、またはホストに追加したり削除したりします。詳細は「[外部 CA 発行の証明書の管理](#)」を参照してください。

ユーザー、ホスト、サービスには複数の証明書を割り当てることができます。



注記

IdM サーバーでサポートされる CA 設定の詳細については、「[CA 設定の決定](#)」を参照してください。

24.1. 統合 IDM CA での証明書の管理

24.1.1. ユーザー、ホスト、またはサービス向けの新規証明書のリクエスト

証明書のリクエスト方法については、以下を参照してください。

- IdM Web UI を使用の場合、「[Web UI: 新規証明書のリクエスト](#)」
- コマンドラインを使用の場合、「[コマンドライン: 新規証明書のリクエスト](#)」

証明書のリクエスト自体はサードパーティーのツールで生成する必要があります。以下の手順では、**certutil** および **openssl** のユーティリティを使用しています。

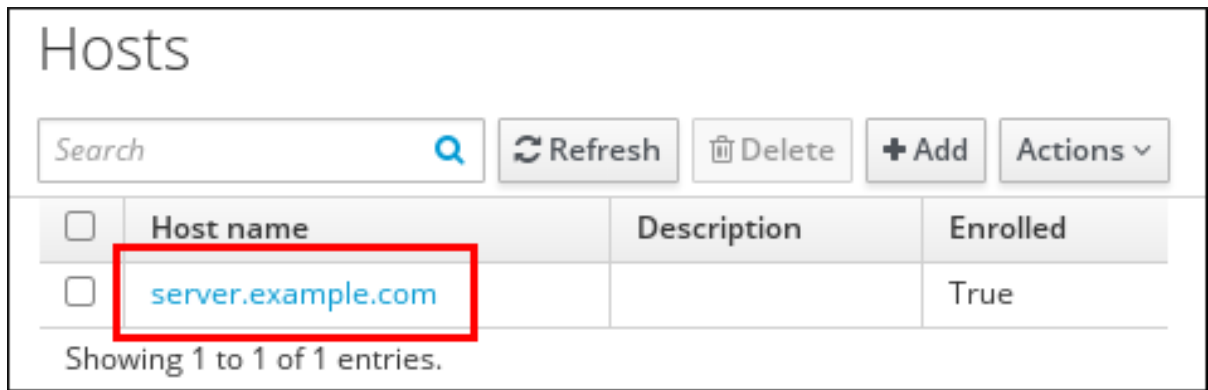


重要

サービスは通常、秘密キーが保存されている専用のサービスノードで稼働しています。このサービスの秘密キーを IdM サーバーにコピーすることは、安全とはみなされません。このため、サービス向けの証明書をリクエストする際には、サービスノード上で CSR を作成してください。

Web UI: 新規証明書のリクエスト

1. **Identity** タブを開き、**Users**、**Hosts**、または **Services** のサブタブを選択します。
2. 設定を開くユーザー、ホスト、またはサービス名をクリックします。



	Host name	Description	Enrolled
<input type="checkbox"/>	server.example.com		True

Showing 1 to 1 of 1 entries.

図24.1 ホストの一覧

3. **Actions** → **New Certificate** をクリックします。
4. オプションで発行元となる CA とプロファイル ID を選択します。
5. 画面の指示に従って **certutil** を使用します。
6. **Issue** をクリックします。

コマンドライン: 新規証明書のリクエスト

通常の状況で **certutil** を使用して新規証明書をリクエストする方法については、[「certutil を使用した新規証明書のリクエスト」](#) を参照してください。**openssl** を使用して新規証明書をリクエストし、Kerberos エイリアスがホストまたはサービス証明書を使用できるようにする方法については、[「openssl を使用した新規証明書のリクエスト」](#) を参照してください。

24.1.1.1. certutil を使用した新規証明書のリクエスト

1. 以下のように、新規の一時証明書データベースを作成します。

```
# certutil -N -d ~/certdb/
```

2. 証明書署名リクエスト (CSR) を作成し、出力をファイルにリダイレクトします。たとえば、4096 ビットの証明書向け CSR を作成して、サブジェクトを `CN=server.example.com,O=EXAMPLE.COM` に設定するには、以下を実行します。

```
# certutil -R -d ~/certdb/ -a -g 4096 -s  
"CN=server.example.com,O=EXAMPLE.COM" -8 server.example.com >  
certificate_request.csr
```

3. 証明書リクエストをサーバーに送信します。新規発行の証明書に関連付けるための Kerberos プリンシパルを必ず指定してください。

```
# ipa cert-request certificate_request.csr --  
principal=host/server.example.com
```

IdM では以下のデフォルト値を使用します。

- 証明書のプロファイル: **caIPAserviceCert**

カスタムのプロファイルを選択する場合は、**--profile-id** オプションを **ipa cert-request** コマンドと使用します。

- 統合 CA: **ipa** (IdM root CA)

サブ CA を選択する場合は、**--ca** オプションを**ipa cert-request** コマンドと使用します。

24.1.1.2. openssl を使用した新規証明書のリクエスト

1. Kerberos プリンシパル *test/server.example.com* 向けに、*test1/server.example.com* や *test2/server.example.com* と言ったエイリアスを作成します。詳細は「[Kerberos プリンシパルのエイリアス](#)」を参照してください。
2. CSR で *dnsName (server.example.com)* および *otherName (test2/server.example.com)* 向けに *subjectAltName* を追加します。これを実行するには、**openssl.conf** ファイルを編集して、UPN *otherName* と *subjectAltName* を指定する以下の行を含めます。

```
otherName=1.3.6.1.4.1.311.20.2.3;UTF8:test2/server.example.com@EXAMPLE.COM
DNS.1 = server.example.com
```

3. **openssl** を使用して証明書リクエストを作成します。

```
openssl req -new -newkey rsa:2048 -keyout test2service.key -sha256 -nodes -out certificate_request.csr -config openssl.conf
```

24.1.2. 統合 IdM CA での証明書の取り消し

証明書の有効期間が過ぎる前にこれを無効にする場合は、取り消すことができます。証明書の取り消しには以下の 2 つの方法があります。

- IdM Web UI を使用の場合、「[Web UI: 証明書の取り消し](#)」
- コマンドラインを使用の場合、「[コマンドライン: 証明書の取り消し](#)」

取り消された証明書は無効で、認証には使用できません。以下の「理由 6: 証明書保留」以外は、取り消しはすべて永続的なものです。

表24.1 取り消しの理由

ID	理由	説明
0	原因不明	
1	キーのセキュリティ侵害	証明書を発行したキーが信頼できません。 考えられる原因: トークンの損失、ファイルへの不適切なアクセス。
2	認証局のセキュリティ侵害	証明書を発行した CA が信頼できません。

ID	理由	説明
3	所属の変更	考えられる原因: <ul style="list-style-type: none"> • ユーザーが会社を離れた、または別の部署に移動した。 • ホストまたはサービスがリタイアしている。
4	交代	現行の証明書が新しい証明書に置き換えられています。
5	運用の停止	ホストまたはサービスが使用停止になっています。
6	証明書の保留	証明書が一時的に取り消されています。証明書は後で復元させることができます。
8	CRL から削除	証明書が、証明書取り消し一覧 (CRL) に含まれていません。
9	特権の撤回	ユーザー、ホスト、またはサービスによる証明書の使用が許可されなくなりました。
10	属性証明局 (AA) のセキュリティ侵害	AA 証明書が信頼できません。

Web UI: 証明書の取り消し

証明書を取り消すには、以下の手順に従います。

1. **Authentication** タブを開き **Certificates** サブタブを選択します。
2. 情報ページを開く証明書のシリアル番号をクリックします。

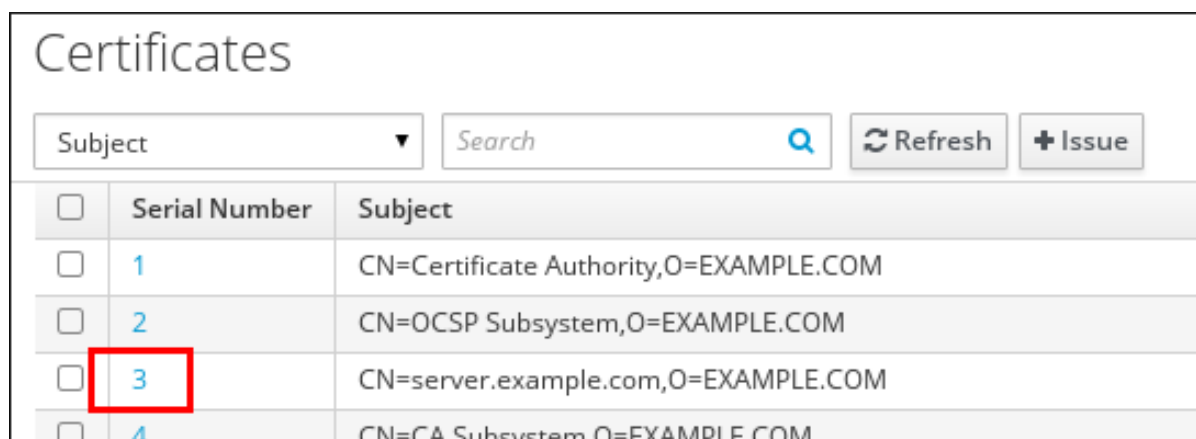


図24.2 証明書一覧

3. **Actions** → **Revoke Certificate** をクリックします。
4. 取り消し理由を選択して **Revoke** をクリックします。理由については、[表24.1「取り消しの理由」](#)を参照してください。

コマンドライン: 証明書の取り消し

ipa cert-revoke コマンドで以下を指定します。

- 証明書のシリアル番号
- 取り消し理由を特定する番号。理由については、[表24.1「取り消しの理由」](#)を参照してください。

たとえば、証明書シリアル番号が **1032** で、理由が「1: キーのセキュリティ侵害」である場合、以下を実行します。

```
$ ipa cert-revoke 1032 --revocation-reason=1
```

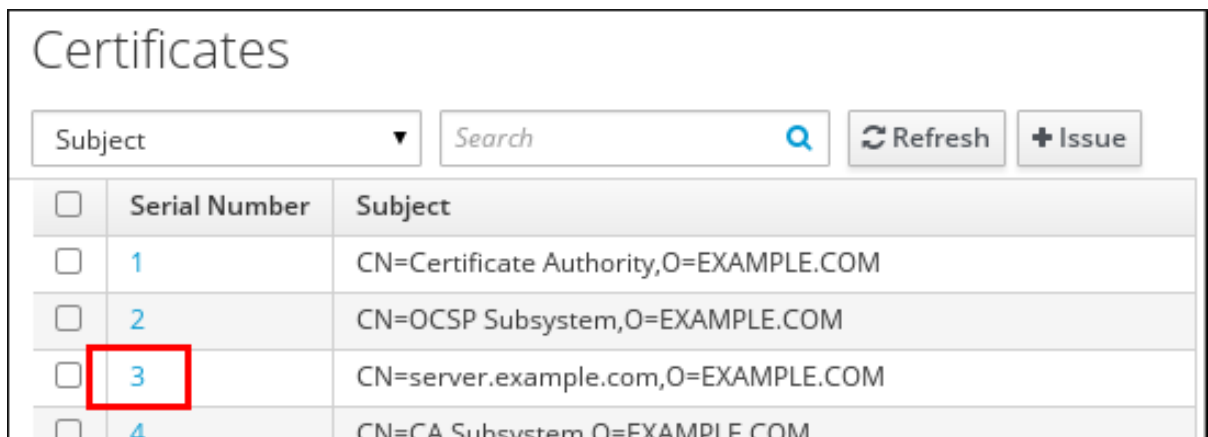
24.1.3. 統合 IdM CA での証明書の復元

「理由 6: 証明書の保留」で証明書を取り消した場合、これを以下の方法で復元することができます。

- IdM Web UI を使用の場合、[「Web UI: 証明書の復元」](#)
- コマンドラインを使用の場合、[「コマンドライン: 証明書の復元」](#)

Web UI: 証明書の復元

1. **Authentication** タブを開き **Certificates** サブタブを選択します。
2. 情報ページを開く証明書のシリアル番号をクリックします。



	Serial Number	Subject
<input type="checkbox"/>	1	CN=Certificate Authority,O=EXAMPLE.COM
<input type="checkbox"/>	2	CN=OCSP Subsystem,O=EXAMPLE.COM
<input type="checkbox"/>	3	CN=server.example.com,O=EXAMPLE.COM
<input type="checkbox"/>	4	CN=CA Subsystem O=EXAMPLE.COM

図24.3 証明書一覧

3. **Actions** → **Restore Certificate** をクリックします。

コマンドライン: 証明書の復元

以下のように **ipa cert-remove-hold** コマンドで証明書のシリアル番号を指定します。

```
$ ipa cert-remove-hold 1032
```

24.2. 外部 CA 発行の証明書の管理

24.2.1. コマンドライン: 外部 CA 発行の証明書の追加および削除

ユーザー、ホスト、またはサービスに証明書を追加するには、以下を実行します。

- **ipa user-add-cert**

- **ipa host-add-cert**
- **ipa service-add-cert**

ユーザー、ホスト、またはサービスから証明書を削除するには、以下を実行します。

- **ipa user-remove-cert**
- **ipa host-remove-cert**
- **ipa service-remove-cert**

外部 CA 発行の証明書は、IdM からこれを削除しても取り消されるわけではありません。これは、証明書が IdM CA データベースに存在しないためです。これらの証明書は、外部 CA 側から手動でしか取り消すことができません。

コマンドでは以下の情報を指定する必要があります。

- ユーザー、ホスト、またはサービスの名前
- Base64 でエンコードされた認証情報

このコマンドを対話型で実行するには、オプションなしで使します。

必須情報をコマンドで直接提供するには、以下のようにコマンドライン引数とオプションを使用します。

```
$ ipa user-add-cert user --certificate=MIQTPrajQAwg...
```

注記

証明書のコンテンツをコマンドラインにコピーする代わりに、証明書を DER 形式に変換して base64 に再度エンコードすることもできます。たとえば、**user_cert.pem** 証明書を **user** に追加するには、以下を実行します。

```
$ ipa user-add-cert user --certificate="$(openssl x509 -outform der -in user_cert.pem | base64 -w 0)"
```

24.2.2. Web UI: 外部 CA 発行の証明書の追加および削除

ユーザー、ホスト、またはサービスに証明書を追加するには、以下を実行します。

1. **Identity** タブを開き、**Users**、**Hosts**、または **Services** のサブタブを選択します。
2. 設定を開くユーザー、ホスト、またはサービス名をクリックします。
3. **Certificates** エントリーの横にある **Add** をクリックします。

User: demouser
demouser is a member of:

Settings | User Groups | Netgroups | Roles | HBAC Rules | Sudo Rules

Refresh | Revert | Save | Actions

Identity Settings

Job Title:

First name *:

Last name *:

Full name *:

Display name:

Initials:

GECOS:

Class:

Account Settings

User login: demouser

Password: *****

Password expiration: 2016-07-14 10:14:41Z

UID:

GID:

Principal alias: demouser@IDM.EXAMPLE.COM

Kerberos principal expiration: : : UTC

Login shell:

Home directory:

SSH public keys:

Certificates:

図24.4 ユーザーアカウントへの証明書の追加

4. Base64 または PEM エンコード形式で証明書をテキストフィールドに貼り付け、**Add** をクリックします。

5. **Save** ボタンをクリックして、変更を保存します。

ユーザー、ホスト、またはサービスから証明書を削除するには、以下を実行します。

1. **Identity** タブを開き、**Users**、**Hosts**、または **Services** のサブタブを選択します。
2. 設定を開くユーザー、ホスト、またはサービス名をクリックします。
3. 削除する証明書の横にある **Actions** をクリックして、**Delete** を選択します。
4. **Save** ボタンをクリックして、変更を保存します。

24.3. 証明書の一覧と表示

Web UI での証明書の一覧と表示

ユーザー、ホスト、またはサービスエントリーに割り当てられた証明書を一覧表示するには、以下の手順に従います。

1. **Identity** タブを開き、**Users**、**Hosts**、または **Services** のサブタブを選択します。
2. 設定を開くユーザー、ホスト、またはサービス名をクリックします。

Hosts

Search

Refresh

Delete

+ Add

Actions

<input type="checkbox"/>	Host name	Description	Enrolled
<input type="checkbox"/>	server.example.com		True

Showing 1 to 1 of 1 entries.

図24.5 ホストの一覧

3. 設定ページでエントリーに割り当てられた証明書がすべて一覧表示されます。また、**Show** をクリックすると特定の証明書が表示されます。

IdM サーバーで登録された全証明書を一覧表示するには、以下の手順に従います。

1. **Authentication** タブを開き **Certificates** サブタブを選択します。
2. **Certificates** セクションに全証明書が一覧表示されます。表示する証明書のシリアル番号をクリックします。

Certificates

Subject

Search

Refresh

Issue

<input type="checkbox"/>	Serial Number	Subject
<input type="checkbox"/>	1	CN=Certificate Authority,O=EXAMPLE.COM
<input type="checkbox"/>	2	CN=OCSP Subsystem,O=EXAMPLE.COM
<input type="checkbox"/>	3	CN=server.example.com,O=EXAMPLE.COM
<input type="checkbox"/>	4	CN=CA Subsystem O=EXAMPLE.COM

図24.6 証明書一覧

コマンドラインからの証明書の一覧表示

IdM データベースにある証明書すべてを一覧表示するには、**ipa cert-find** コマンドを実行します。

```
$ ipa cert-find
-----
10 certificates matched
-----
Serial number (hex): 0x1
Serial number: 1
Status: VALID
Subject: CN=Certificate Authority,O=EXAMPLE.COM
...
Number of entries returned 10
-----
```

発行日や有効期間などの特定の属性を指定すると、検索結果をフィルタリングすることができます。たとえば、発行日で検索するには **--issuedon-from** および **--issuedon-to** のオプションを使用して開始日と終了日や一定期間を指定します。

```
ipa cert-find --issuedon-from=2017-01-07 --issuedon-to=2017-02-07
```

証明書の検索のフィルタリングに使用可能なオプションの完全一覧については、**ipa cert-find** に **-help** オプションを付けて実行します。

コマンドラインからの証明書の表示

証明書を表示するには、**ipa cert-show** コマンドでシリアル番号を指定して実行します。

```
$ ipa cert-show 132
Serial number: 132
Certificate:
MIIDtzCCAp+gAwIBAgIBATANBgqhkiG9w0BAQsFADBBMR8wHQYDVQKExZMQUIu
...
LxIQjrEFtJmoBGB/TWRLwGEWy1ayr4iTEf1ayZ+RGNylLalEAtk9RLjEjg==
Subject: CN=Certificate Authority,O=EXAMPLE.COM
Issuer: CN=Certificate Authority,O=EXAMPLE.COM
Not Before: Sun Jun 08 05:51:11 2014 UTC
Not After: Thu Jun 08 05:51:11 2034 UTC
Serial number (hex): 0x132
Serial number: 132
```

ユーザー、ホスト、またはサービスエントリーに割り当てられた証明書を表示するには、**ipa cert-show** でエントリーを指定します。たとえば、ユーザーに割り当てられた証明書を表示するには、以下を実行します。

```
$ ipa user-show user
User login: user
...
Certificate: MIICfzCCAWcCAQA...
...
```

--out オプションを **ipa cert-show** に追加すると、証明書をファイルに保存することもできます。

```
$ ipa cert-show certificate_serial_number --out=path_to_file
```

ユーザー、ホスト、またはサービスに複数の証明書がある場合は、**--out** オプションではそれらすべてがエクスポートされます。証明書は、PEM オブジェクトとしてエクスポートされます。

24.4. 証明書のプロファイル

証明書プロファイルは、証明書発行に関する制限、登録方法、登録用の出入力形式のほかに特定のプロファイルに属する証明書のコンテンツを定義します。ある証明書プロファイルは特定のタイプの証明書の発行に関連付けられます。IdM のユーザー、サービス、およびホストでは、異なる証明書プロファイルを定義できます。

CA は証明書の署名で証明書プロファイルを使用して以下を判定します。

- CA が証明書署名リクエスト (CSR) を受け付けられるかどうか。
- 証明書にどの機能と拡張機能を記載するか。

IdM にはデフォルトで、**caIPAServiceCert** と **IECUserRoles** の 2 つの証明書プロファイルがあります。これに加えて、カスタムプロファイルをインポートすることもできます。

カスタムプロファイルを使用すると、特定かつ関連のない目的で証明書を発行することが可能になります。たとえば、あるプロファイルの使用を 1 ユーザーもしくは 1 グループに限定し、他のユーザーやグループがこのプロファイルを使用して認証目的で証明書を発行できないようにすることが可能です。

サポートされる証明書プロファイルの設定については、Red Hat Certificate System 『Administration Guide』の「[Defaults Reference](#)」と「[Constraints Reference](#)」を参照してください。



注記

証明書プロファイルと CA ACL 「[証明局の ACL ルール](#)」を結びつけることで、管理者はカスタム証明書プロファイルへのアクセスを定義、制御できるようになります。プロファイルと CA ACL を使用してユーザー証明書を発行する方法については、「[IdM CA での証明書プロファイルおよび ACL を使用したユーザー証明書の発行](#)」を参照してください。

24.4.1. コマンドラインからの証明書プロファイル管理

IdM プロファイル管理用の **certprofile** プラグインを使用すると、権限のあるユーザーは IdM 証明書プロファイルのインポート、修正、または削除ができるようになります。このプラグインがサポートするコマンドすべてを表示するには、**ipa certprofile** コマンドを実行します。

```
$ ipa certprofile
Manage Certificate Profiles
```

```
...
```

EXAMPLES:

Import a profile that will not store issued certificates:

```
ipa certprofile-import ShortLivedUserCert \
  --file UserCert.profile --desc "User Certificates" \
  --store=false
```

Delete a certificate profile:

```
ipa certprofile-del ShortLivedUserCert
```

```
...
```

certprofile 操作を実行するには、必要なパーミッションがあるユーザーとして操作する必要があります。IdM にはデフォルトで以下の証明書プロファイル関連のパーミッションがあります。

System: Read Certificate Profiles

ユーザーが全プロファイル属性を読み込むことを許可します。

System: Import Certificate Profile

ユーザーが証明書プロファイルを IdM にインポートすることを許可します。

System: Delete Certificate Profile

ユーザーが既存の証明書プロファイルを削除することを許可します。

System: Modify Certificate Profile

ユーザーがプロファイル属性を修正し、プロファイルを有効化、無効化することを許可します。

これらのパーミッションはすべて、デフォルトの **CA Administrator** 権限に含まれています。IdM のロールベースのアクセス制御およびパーミッションの管理についての詳細は、[「ロールベースのアクセス制御の定義」](#) を参照してください。



注記

証明書をリクエストする際には、**--profile-id** オプションを **ipa cert-request** コマンドに追加して使用するプロファイルを指定することができます。プロファイル ID が指定されない場合は、デフォルトの **caIPAServiceCert** プロファイルが使用されます。

本セクションでは、**ipa certprofile** コマンドを使用してプロファイルを管理する重要な側面のみを説明しています。このコマンドに関する完全な情報については、以下のように **--help** オプションを追加して実行してください。

```
$ ipa certprofile-mod --help
Usage: ipa [global-options] certprofile-mod ID [options]

Modify Certificate Profile configuration.
Options:
  -h, --help          show this help message and exit
  --desc=STR          Brief description of this profile
  --store=B00L        Whether to store certs issued using this profile
  ...
```

証明書プロファイルのインポート

IdM に新規の証明書プロファイルをインポートするには、**ipa certprofile-import** コマンドを使用します。このコマンドをオプションなしで実行すると対話型セッションが開始され、**certprofile-import** スクリプトが証明書のインポートに必要な情報を要求します。

```
$ ipa certprofile-import

Profile ID: smime
Profile description: S/MIME certificates
Store issued certificates [True]: TRUE
Filename of a raw profile. The XML format is not supported.: smime.cfg
-----
Imported profile "smime"
-----
  Profile ID: smime
  Profile description: S/MIME certificates
  Store issued certificates: TRUE
```

ipa certprofile-import コマンドは以下のようなオプションを取ります。

--file

このオプションは、プロファイル設定を含むファイルを直接 **ipa certprofile-import** に渡します。

```
$ ipa certprofile-import --file=smime.cfg
```


■

--store

このオプションは **Store issued certificates** 属性を設定します。以下の 2 つの値を受け付けます。

- **True**。この場合、発行された証明書がクライアントに配布され、ターゲットの IdM プリンシパルの **userCertificate** 属性に保存されます。
- **False**。この場合、発行された証明書がクライアントに配布されますが、IdM には保存されません。このオプションは、複数の短期証明書を発行する際によく使用されます。

ipa certprofile-import で指定されたプロファイル ID が既に使用されている場合、またはプロファイルコンテンツが正しくない場合は、インポートに失敗します。たとえば、必須の属性がない場合や、提供されたファイルで定義されているプロファイル ID が **ipa certprofile-import** で指定されたものと一致しない場合は、インポートに失敗します。

新規プロファイル用のテンプレートを取得するには、**ipa certprofile-show** コマンドを **--out** オプションと実行することで、指定された既存のプロファイルがファイルにエクスポートされます。例を示します。

```
$ ipa certprofile-show caIPAServiceCert --out=file_name
```

この後、エクスポートされたファイルを必要に応じて編集し、新規プロファイルとしてインポートできます。

証明書プロファイルの表示

IdM に保存されている証明書プロファイルすべてを表示するには、**ipa certprofile-find** コマンドを使用します。

```
$ ipa certprofile-find
-----
3 profiles matched
-----
Profile ID: caIPAServiceCert
Profile description: Standard profile for network services
Store issued certificates: TRUE

Profile ID: IECUserRoles
...
```

特定プロファイルの情報を表示するには、**ipa certprofile-show** コマンドを使用します。

```
$ ipa certprofile-show profile_ID
Profile ID: profile_ID
Profile description: S/MIME certificates
Store issued certificates: TRUE
```

証明書プロファイルの修正

既存の証明書プロファイルを修正するには、**ipa certprofile-mod** コマンドを使用します。**ipa certprofile-mod** にコマンドラインオプションで必要な修正を渡します。たとえば、プロファイルの内容を修正し、IdM が発行された証明書を保存するかどうかを変更するには、以下を実行します。

```
$ ipa certprofile-mod profile_ID --desc="New description" --store=False
```

```
-----
Modified Certificate Profile "profile_ID"
-----
```

```
Profile ID: profile_ID
Profile description: New description
Store issued certificates: FALSE
```

証明書プロファイル設定を更新するには、**--file** オプションを使用して更新された設定を含むファイルをインポートします。

```
$ ipa certprofile-mod profile_ID --file=new_configuration.cfg
```

証明書プロファイルの削除

IdM から既存の証明書プロファイルを削除するには、**ipa certprofile-del** コマンドを使用します。

```
$ ipa certprofile-del profile_ID
-----
Deleted profile "profile_ID"
-----
```

24.4.2. Web UI からの証明書プロファイル管理

IdM Web UI で証明書プロファイルを管理するには、以下の手順に従います。

1. **Authentication** タブを開き **Certificates** サブタブを選択します。
2. **Certificate Profiles** セクションを開きます。

IdentityPolicyAuthenticationNetwork ServicesIPA Server

CertificatesOTP TokensRADIUS Servers

Certificates

Certificates

Certificate Profiles >

CA ACLs

Certificate Profiles

Search

RefreshDelete

<input type="checkbox"/>	Profile ID	Profile description	Store issued certificates
<input type="checkbox"/>	IECUserRoles	User profile that includes IECUserRoles extension from request	TRUE
<input type="checkbox"/>	calPAserviceCert	Standard profile for network services	TRUE

Showing 1 to 2 of 2 entries.

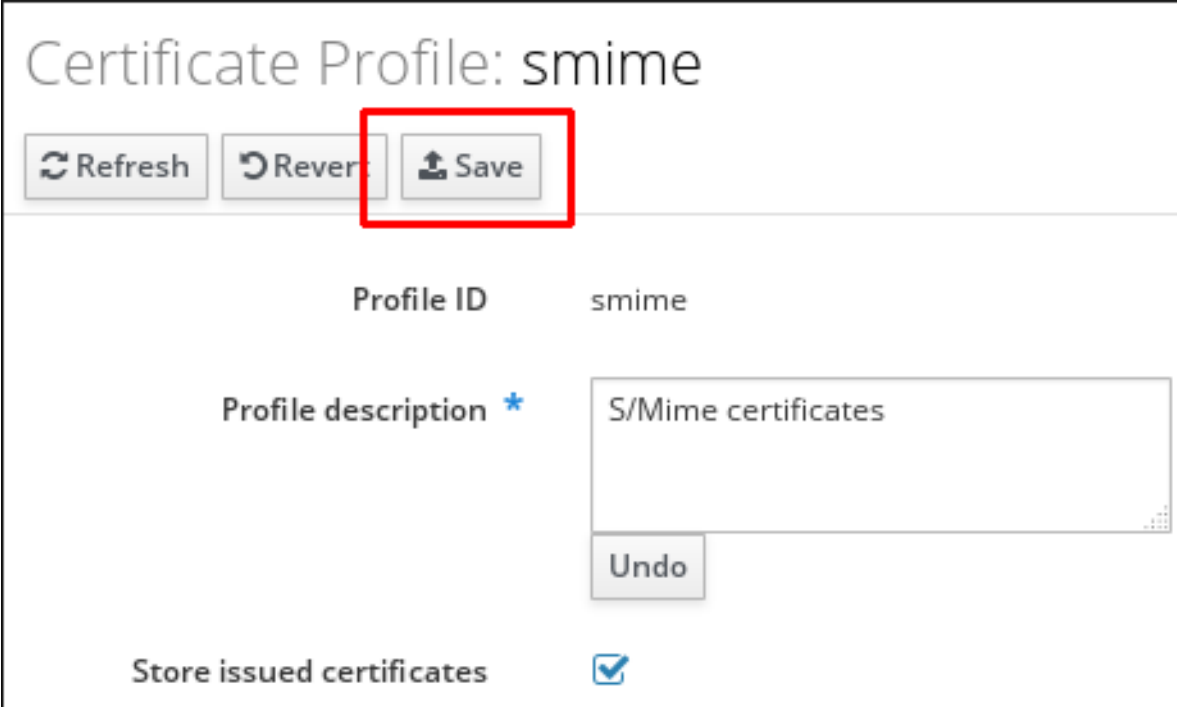
図24.7 Web UI での証明書プロファイル管理

Certificate Profiles セクションでは、既存のプロファイルについての情報を表示したり、属性を修正したり、選択したプロファイルを削除することができます。

たとえば、既存の証明書プロファイルを修正するには、以下の手順に従います。

1. プロファイル名をクリックして、プロファイルの設定ページを開きます。
2. 開いたページで必要な情報を入力します。

3. **Save** をクリックして新規設定を保存します。



Certificate Profile: smime

Refresh Revert Save

Profile ID smime

Profile description * S/Mime certificates

Undo

Store issued certificates ☒

図24.8 Web UI での証明書プロファイルの修正

Store issued certificates オプションを有効にすると、発行された証明書はクライアントに配布されるほか、ターゲットの IdM プリンシパルの **userCertificate** 属性に保存されます。このオプションを無効にすると、発行された証明書はクライアントに配布されますが、IdM には保存されません。複数の短期証明書の発行が必要な際には、証明書の保存は無効にされることがよくあります。

Web UI では現時点で、証明書プロファイル管理の以下の操作ができないことに注意してください。

- Web UI では証明書プロファイルのインポートができません。証明書をインポートするには、**ipa certprofile-import** コマンドを使用します。
- 属性と値のペアを設定、追加、または削除することができません。このペアを修正する場合は、**ipa certprofile-mod** コマンドを使用します。
- 更新された証明書プロファイルの設定をインポートすることはできません。これを実行するには、**ipa certprofile-mod --file=file_name** コマンドを使用します。

証明書プロファイル管理に使用するコマンドの詳細については、「[コマンドラインからの証明書プロファイル管理](#)」を参照してください。

24.4.3. 証明書プロファイルのある IdM サーバーのアップグレード

IdM サーバーをアップグレードすると、そこに含まれているプロファイルはすべてインポート、有効化されます。

複数のサーバーレプリカをアップグレードする場合は、最初にアップグレードされるレプリカのプロファイルがインポートされます。他のレプリカに関しては、IdM は残りのレプリカがあることを検出しますが、それらをインポートしたり、プロファイル間の競合を解決したりすることはありません。レプリカ上にカスタムプロファイルが定義されている場合は、アップグレード前に全レプリカ上のプロファイルで整合性がとれていることを確認してください。

24.5. 証明局の ACL ルール

証明局のアクセス制御リスト (CA ACL) ルールでは、どのユーザー、サーバー、またはホストにどのプロファイルを使って証明書を発行するかを定義します。CA ACL でプロファイル、プリンシパル、およびグループを関連付けることで、プリンシパルまたはグループが特定のプロファイルを使用した証明書をリクエストできるようになります。

- ACL は複数プロファイルへのアクセスを許可します
- ACL では複数のユーザー、サービス、ホスト、ユーザーグループ、およびホストグループに関連付けができます

たとえば、管理者は CA ACL を使用すると、ロンドンのオフィスから作業している社員向けのプロファイルの使用を、そのオフィスに関連するグループのメンバーとなっているホストに限定することができます。



注記

「[証明書のプロファイル](#)」で説明されている証明書プロファイルと CA ACL を結びつけることで、管理者はカスタム証明書プロファイルへのアクセスを定義、制御できるようになります。プロファイルと CA ACL を使用してユーザー証明書を発行する方法については、「[IdM CA での証明書プロファイルおよび ACL を使用したユーザー証明書の発行](#)」を参照してください。

24.5.1. コマンドラインからの CA ACL 管理

CA ACL ルール管理用の **caacl** プラグインを使用すると、権限のあるユーザーは指定された CA ACL の追加、表示、修正、または削除ができるようになります。このプラグインがサポートするコマンドすべてを表示するには、**ipa caacl** コマンドを実行します。

```
$ ipa caacl
Manage CA ACL rules.
```

```
...
```

EXAMPLES:

Create a CA ACL "test" that grants all users access to the "UserCert" profile:

```
ipa caacl-add test --usercat=all
ipa caacl-add-profile test --certprofiles UserCert
```

Display the properties of a named CA ACL:

```
ipa caacl-show test
```

Create a CA ACL to let user "alice" use the "DNP3" profile on "DNP3-CA":

```
ipa caacl-add alice_dnp3
ipa caacl-add-ca alice_dnp3 --cas DNP3-CA
ipa caacl-add-profile alice_dnp3 --certprofiles DNP3
ipa caacl-add-user alice_dnp3 --user=alice
```

```
...
```

caacl 操作を実行するには、必要なパーミッションがあるユーザーとして操作する必要があります。IdM にはデフォルトで以下の CA ACL 関連のパーミッションがあります。

System: Read CA ACLs

ユーザーが CA ACL の属性すべてを読み込むことを許可します。

System: Add CA ACL

ユーザーが新規 CA ACL を追加することを許可します。

System: Delete CA ACL

ユーザーが既存の CA ACL を削除することを許可します。

System: Modify CA ACL

ユーザーが CA ACL 属性を修正し、CA ACL を有効化、無効化することを許可します。

System: Manage CA ACL membership

ユーザーが CA ACL 内の CA、プロファイル、ユーザー、ホスト、およびサービスメンバーシップを管理することを許可します。

これらのパーミッションはすべて、デフォルトの **CA Administrator** 権限に含まれています。IdM のロールベースのアクセス制御およびパーミッションの管理についての詳細は、[「ロールベースのアクセス制御の定義」](#) を参照してください。

本セクションでは、**ipa caacl** コマンドを使用して CA ACL を管理する重要な側面のみを説明しています。このコマンドに関する完全な情報については、以下のように **--help** オプションを追加して実行してください。

```
$ ipa caacl-mod --help
Usage: ipa [global-options] caacl-mod NAME [options]

Modify a CA ACL.
Options:
  -h, --help                show this help message and exit
  --desc=STR                 Description
  --cacat=['all']           CA category the ACL applies to
  --profilecat=['all']      Profile category the ACL applies to
  ...
```

CA ACL の作成

新規の CA ACL を作成するには、**ipa caacl-add** コマンドを使用します。このコマンドをオプションなしで実行すると対話型セッションが開始され、**ipa caacl-add** スクリプトが新規 CA ACL について必要な情報を要求します。

```
$ ipa caacl-add
ACL name: smime_acl
-----
Added CA ACL "smime_acl"
-----
ACL name: smime_acl
Enabled: TRUE
```

新規 CA ACL はデフォルトで有効になります。

ipa caacl-add で使用可能なオプションのうち重要なものは、CA ACL を CA、証明書プロファイル、ユーザー、ホスト、またはサービスカテゴリーと関連付けるものです。

- **--cacat**
- **--profilecat**

- **--usercat**
- **--hostcat**
- **--servicecat**

IdM ではこれらのオプションで **all** の値のみを受け付けます。これは、CA ACL をすべての CA、プロファイル、ユーザー、ホスト、またはサービスと関連付けます。たとえば、CA ACL を全ユーザーおよびユーザーグループと関連付けるには、以下を実行します。

```
$ ipa caacl-add ca_acl_name --usercat=all
```

CA、プロファイル、ユーザー、ホスト、およびサービスのカテゴリーは、CA ACL に特定のオブジェクトやオブジェクトのグループを追加することの代わりとなるものです。これについては、「[CA ACL へのエントリー追加と CA ACL からのエントリー削除](#)」で説明しています。カテゴリーで使用するタイプと同じオブジェクトやグループを追加できないことに注意してください。たとえば、**--usercat=all** オプションを使った後に、**ipa caacl-add-user --users=user_name** コマンドでユーザーを CA ACL に追加することはできません。

注記

ユーザーもしくはグループが対応する CA ACL に追加されていないと、証明書プロファイルを使用しているそのユーザーもしくはグループの証明書をリクエストしても失敗します。例を示します。

```
$ ipa cert-request CSR-FILE --principal user --profile-id
profile_id
ipa: ERROR Insufficient access: Principal 'user' is not
permitted to use CA '.' with profile 'profile_id' for
certificate issuance.
```

この場合、「[CA ACL へのエントリー追加と CA ACL からのエントリー削除](#)」にあるように、CA ACL にユーザーもしくはグループを追加するか、この CA ACL を **all** ユーザーカテゴリーに関連付ける必要があります。

CA ACL の表示

すべての CA ACL を表示するには、**ipa caacl-find** コマンドを使用します。

```
$ ipa caacl-find
-----
2 CA ACLs matched
-----
ACL name: hosts_services_caIPAserviceCert
Enabled: TRUE
...
```

ipa caacl-find コマンドでは **--cacat**、**--profilecat**、**--usercat**、**--hostcat**、および **--servicecat** のオプションを使用して、これらに対応する CA、証明書プロファイル、ユーザー、ホスト、またはサービスカテゴリーがある CA ACL に検索結果を絞り込むことができます。ただし、IdM ではこれらのオプションで **all** カテゴリーしか受け付けられないことに注意してください。これらのオプションについての詳細は、「[CA ACL の作成](#)」を参照してください。

特定の CA ACL についての情報を表示するには、**ipa caacl-show** コマンドを実行します。

```
$ ipa caacl-show ca_acl_name
ACL name: ca_acl_name
Enabled: TRUE
Host category: all
...
```

CA ACL の修正

既存の CA ACL を修正するには、**ipa caacl-mod** コマンドを使用します。**ipa caacl-mod** にコマンドラインオプションで必要な修正を渡します。たとえば、CA ACL の記述内容を修正し、その CA ACL を全証明書プロファイルに関連付けるには、以下を実行します。

```
$ ipa caacl-mod ca_acl_name --desc="New description" --profilecat=all
-----
Modified CA ACL "ca_acl_name"
-----
ACL name: smime_acl
Description: New description
Enabled: TRUE
Profile category: all
```

ipa caacl-mod コマンドで利用できる重要なオプションは、**--cacat**、**--profilecat**、**--usercat**、**--hostcat**、および **--servicecat** です。これらのオプションについての説明は、「[CA ACL の作成](#)」を参照してください。

CA ACL の有効化および無効化

CA ACL を無効にするには、**ipa caacl-disable** コマンドを使用します。

```
$ ipa caacl-disable ca_acl_name
-----
Disabled CA ACL "ca_acl_name"
-----
```

無効になった CA ACL は適用されず、証明書のリクエストには使用できません。ただし、CA ACL を無効にしても、これが IdM から削除されるわけではありません。

CA ACL を有効にするには、**ipa caacl-enable** コマンドを使用します。

```
$ ipa caacl-enable ca_acl_name
-----
Enabled CA ACL "ca_acl_name"
-----
```

CA ACL の削除

既存の CA ACL を削除するには、**ipa caacl-del** コマンドを使用します。

```
$ ipa caacl-del ca_acl_name
```

CA ACL へのエントリー追加と CA ACL からのエントリー削除

ipa caacl-add-* と **ipa caacl-remove-*** コマンドを使用することで、それぞれ CA ACL に新規エントリーを追加したり、既存のエントリーを削除することができます。

ipa caacl-add-ca と **ipa caacl-remove-ca**

CA を追加または削除します。

ipa caacl-add-host と ipa caacl-remove-host

ホストもしくはホストグループを追加または削除します。

ipa caacl-add-profile と ipa caacl-remove-profile

プロファイルを追加または削除します。

ipa caacl-add-service と ipa caacl-remove-service

サービスを追加または削除します。

ipa caacl-add-user と ipa caacl-remove-user

ユーザーもしくはグループを追加または削除します。

以下に例を示します。

```
$ ipa caacl-add-user ca_acl_name --groups=group_name
```

CA ACL に追加するオブジェクトまたはオブジェクトグループと同じオブジェクトのカテゴリーを「[CA ACL の作成](#)」にあるように使用できないことに注意してください。たとえば、**ipa caacl-add-user --users=user_name** コマンドを **--usercat=all** オプションで指定した CA ACL で実行しようとする、これは失敗します。

```
$ ipa caacl-add-user ca_acl_name --users=user_name
ipa: ERROR: users cannot be added when user category='all'
```

注記

ユーザーもしくはグループが対応する CA ACL に追加されていないと、証明書プロファイルを使用しているそのユーザーもしくはグループの証明書をリクエストしても失敗します。例を示します。

```
$ ipa cert-request CSR-FILE --principal user --profile-id
profile_id
ipa: ERROR Insufficient access: Principal 'user' is not
permitted to use CA '.' with profile 'profile_id' for
certificate issuance.
```

この場合、CA ACL にユーザーもしくはグループを追加するか、「[CA ACL の作成](#)」にあるように、この CA ACL を **all** ユーザーカテゴリーに関連付ける必要があります。

これらのコマンドで必須の構文と利用可能なオプションについては、以下のようにコマンドに **--help** を追加して実行してください。

```
$ ipa caacl-add-user --help
```

24.5.2. Web UI での CA ACL 管理

IdM Web UI で CA ACL を管理するには、以下の手順に従います。

1. **Authentication** タブを開き **Certificates** サブタブを選択します。

2. CA ACLs セクションを開きます。

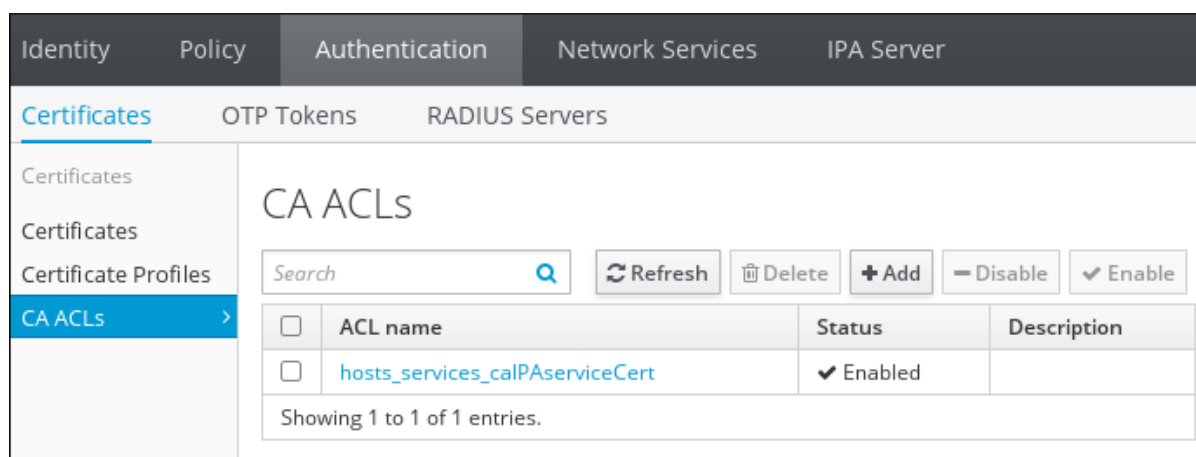


図24.9 Web UI での CA ACL ルールの管理

CA ACLs セクションでは、新規 CA ACL の追加、既存 CA ACL についての情報表示、属性の修正、選択した CA ACL の有効化と無効化、さらにはそれを削除することができます。

たとえば、既存の CA ACL を修正するには、以下を実行します。

1. CA ACL 名をクリックして、CA ACL の設定ページを開きます。
2. 開いたページで必要な情報を入力します。

Profiles と **Permitted to have certificates issued** のセクションでは、CA ACL を証明書プロファイル、ユーザーまたはユーザーグループ、ホストまたはホストグループ、もしくはサービスに関連付けることができます。**Add** ボタンでこれらのオブジェクトを追加するか、**Anyone** オプションを選択して CA ACL を全ユーザー、ホスト、またはサービスに関連付けます。

3. **Save** をクリックして新規設定を保存します。

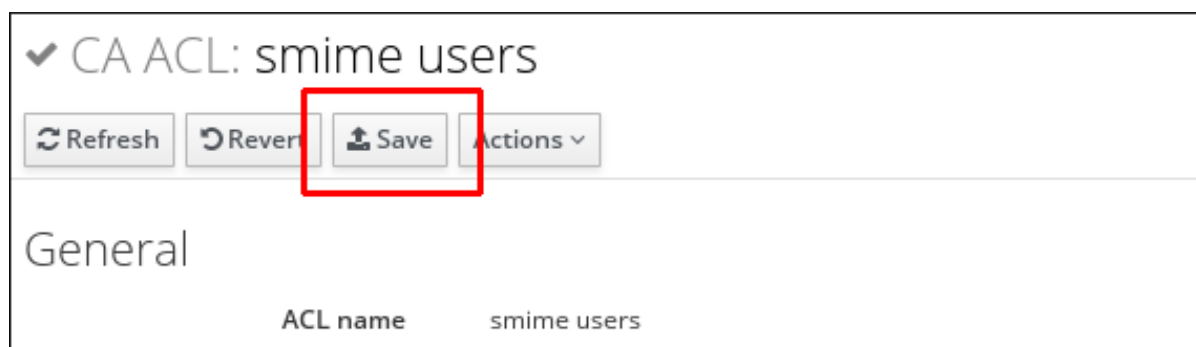


図24.10 Web UI での CA ACL ルールの修正

24.6. IDM CA での証明書プロファイルおよび ACL を使用したユーザー証明書の発行

証明局のアクセス制御リスト (CA ACL) で許可されている場合は、ユーザーは自分自身のための証明書をリクエストすることができます。以下の手順では、証明書プロファイルと CA ACL を使用します。これらはそれぞれ、「[証明書のプロファイル](#)」と「[証明局の ACL ルール](#)」で説明されているので、詳細はこれらのセクションを参照してください。

コマンドラインからのユーザーへの証明書発行

1. ユーザー証明書のリクエストを処理する新規のカスタム証明書プロファイルを作成またはインポートします。例を示します。

```
$ ipa certprofile-import certificate_profile --
file=certificate_profile.cfg --store=True
```

2. ユーザーエントリーの証明書のリクエストを許可するために使用される新規の証明局 (CA) ACL を追加します。

```
$ ipa caacl-add users_certificate_profile --usercat=all
```

3. カスタム証明書プロファイルを CA ACL に追加します。

```
$ ipa caacl-add-profile users_certificate_profile --
certprofiles=certificate_profile
```

4. ユーザー用の証明書リクエストを生成します。以下の例では、OpenSSL を使用します。

```
$ openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout
private.key -out cert.csr -subj '/CN=user'
```

5. **ipa cert-request** コマンドを実行して、IdM CA がユーザー用の新規証明書を発行するようにします。

```
$ ipa cert-request cert.csr --principal=user --profile-
id=certificate_profile
```

--ca sub-CA_name オプションを渡すと、root CA **ipa** ではなく、sub-CA からの証明書をリクエストすることもできます。

発行された証明書が当該ユーザーに割り当てられていることを確認するには、**ipa user-show** コマンドを実行します。

```
$ ipa user-show user
User login: user
...
Certificate: MIICfzCCAWcCAQA...
...
```

Web UI でのユーザーへの証明書発行

1. ユーザー証明書のリクエストを処理する新規のカスタム証明書プロファイルを作成またはインポートします。プロファイルのインポートは、コマンドラインからしかできません。例を示します。

```
$ ipa certprofile-import certificate_profile --
file=certificate_profile.txt --store=True
```

証明書プロファイルについての詳細は、「[証明書のプロファイル](#)」を参照してください。

2. Web UI の **Authentication** タブで、**CA ACLs** セクションを開きます。

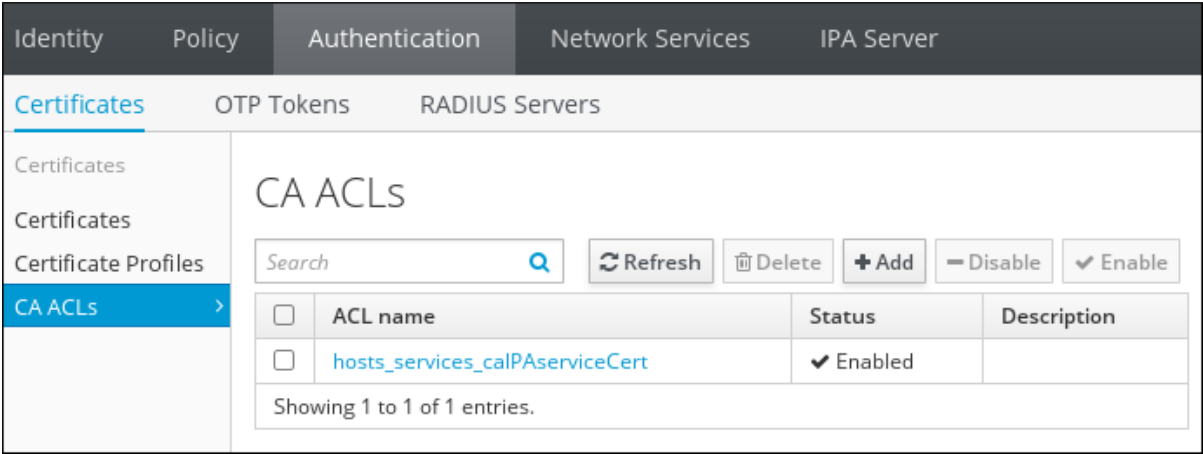


図24.11 Web UI での CA ACL ルールの管理

CA ACL リスト上部にある **Add** をクリックして、ユーザーエントリー用の証明書リクエストを許可する新規 CA ACL を追加します。

- a. **Add CA ACL** ウィンドウが開くので、新規 CA ACL についての必要な情報を入力します。

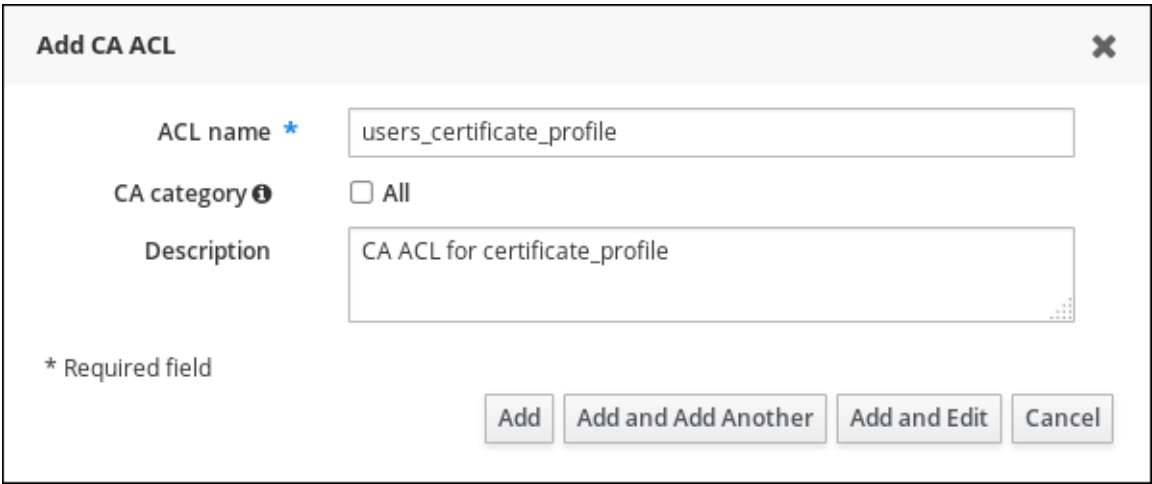


図24.12 新規 CA ACL の追加

Add and Edit をクリックして、CA ACL の設定ページに移動します。

- b. CA ACL 設定ページで **Profiles** セクションまでスクロールし、プロファイル一覧上部にある **Add** をクリックします。

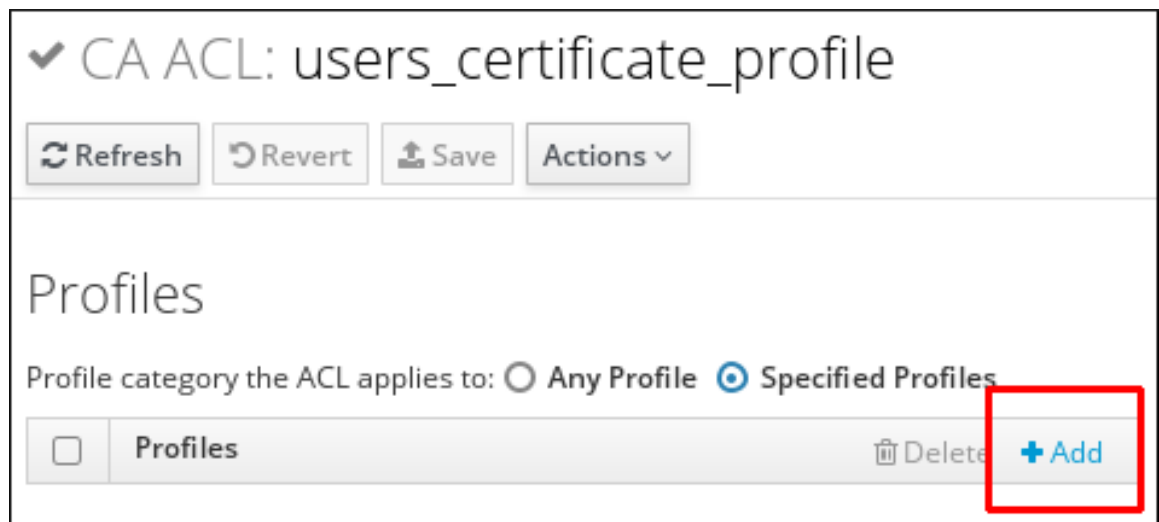


図24.13 CA ACL への証明書プロファイルの追加

- c. プロファイルを選択して **Prospective** コラムに移動し、カスタム証明書プロファイルを検査して CA ACL に追加します。

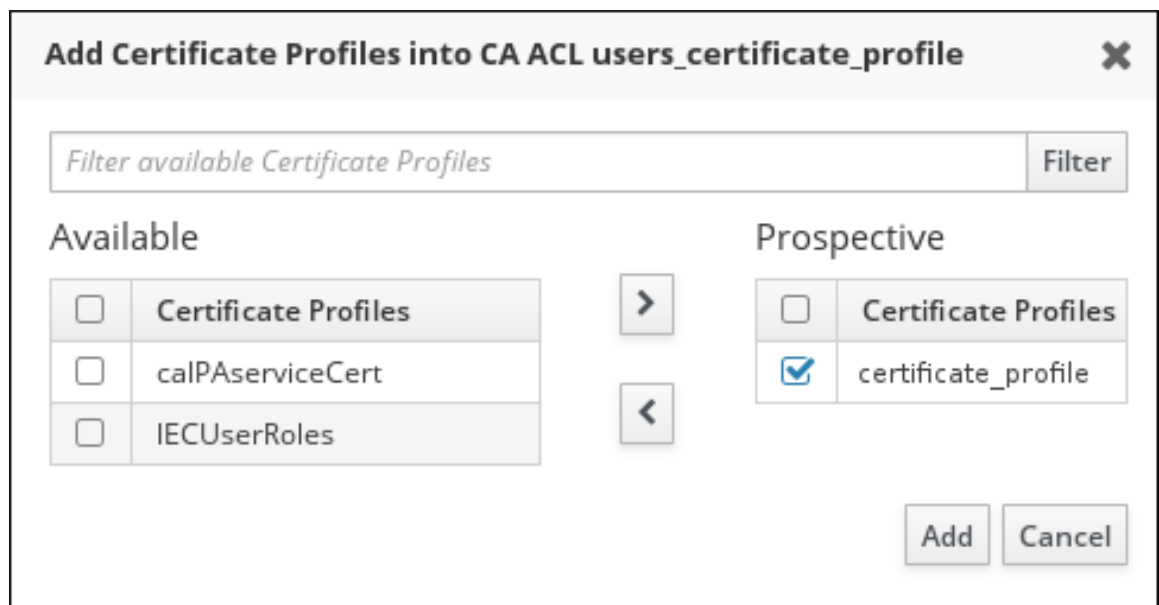


図24.14 証明書プロファイルの選択

Add をクリックします。

- d. **Permitted to have certificates issued** セクションまでスクロールします。CA ACL をユーザーもしくはユーザーグループに関連付けます。

Add ボタンでユーザーもしくはグループを追加するか、**Anyone** オプションを選択して CA ACL を全ユーザーに関連付けます。

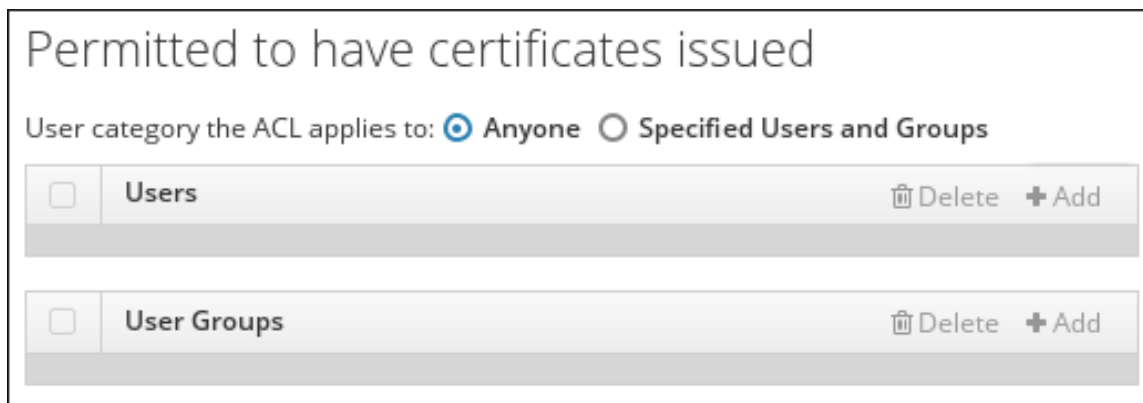


図24.15 ユーザーの CA ACL への追加

- e. **Permitted to have certificates issued** セクションでは、CA ACL を 1 つ以上の CA に関連付けることができます。

Add ボタンで CA を追加するか、**Any CA** オプションを選択して CA ACL を全 CA に関連付けることができます。

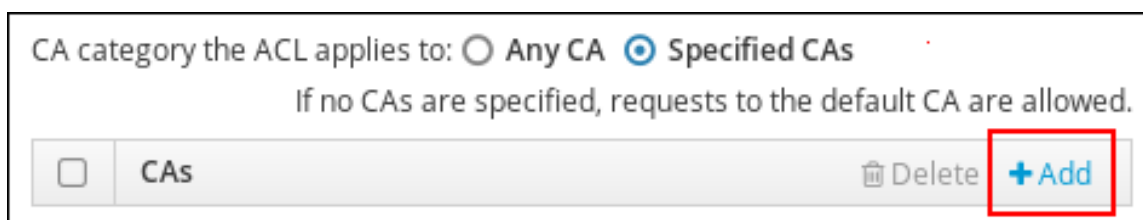


図24.16 CA を CA ACL に追加する

- f. CA ACL 設定ページ上部にある **Save** をクリックして、CA ACL の変更を保存します。
3. ユーザー用の新規証明書をリクエストします。
- a. **Identity** タブにある **Users** サブタブで、リクエストする証明書が必要とするユーザーを選択します。ユーザー名をクリックして、ユーザーエントリーの選択ページを開きます。
- b. 設定ページ上部にある **Actions** をクリックしてから **New Certificate** をクリックします。

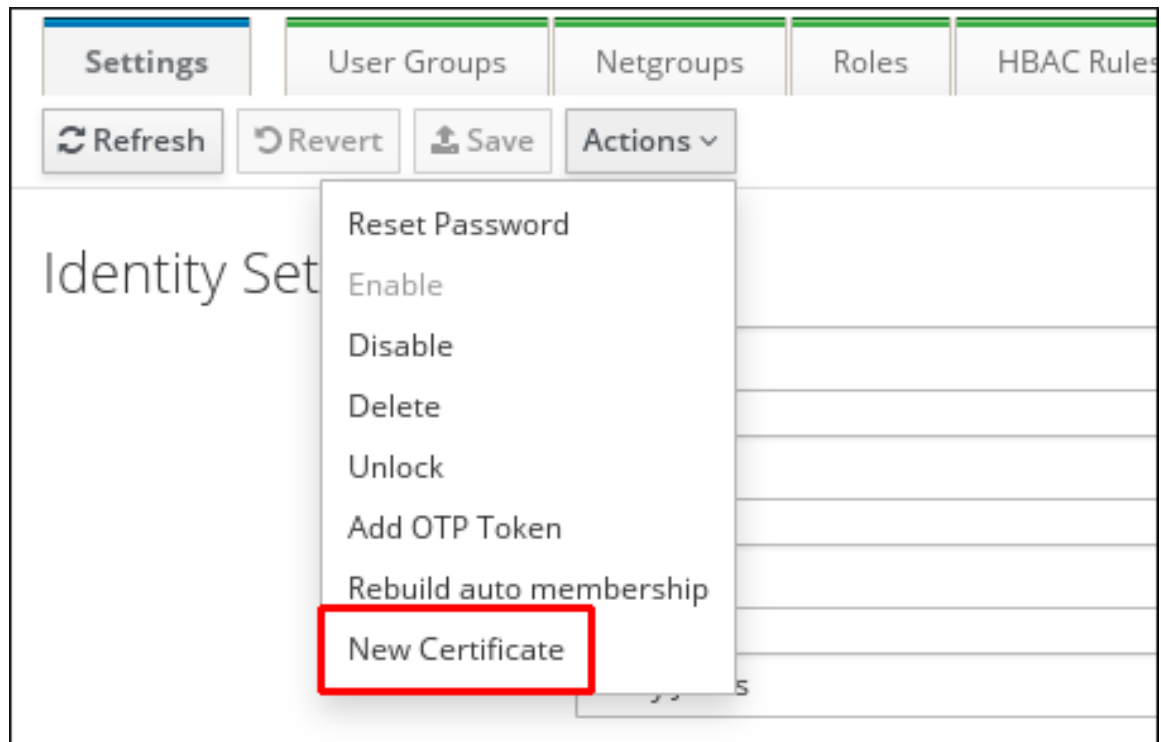


図24.17 ユーザー用証明書のリクエスト

c. 必要な情報を入力します。

Issue New Certificate for User user
✕

CA ★

Profile ID

- Create a certificate database or use an existing one. To create a new database:

```
# certutil -N -d <database path>
```
- Create a CSR with subject `CN=<uid>,O=<realm>`, for example:

```
# certutil -R -d <database path> -a -g <key size> -s 'CN=user,O=IDM.EXAMPLE.COM'
```
- Copy and paste the CSR (from `-----BEGIN NEW CERTIFICATE REQUEST-----` to `-----END NEW CERTIFICATE REQUEST-----`) into the text area below:

```
-----BEGIN CERTIFICATE REQUEST-----
MIICVTCCAT0CAQAwEDEOMAwGA1UEAwwFdHVzZXIwggEIMA0GCSqGSIb3DQEBAQUA
A4IBDwAwggEKAoIBAQDGH9WgTNE4Zbh8MCpsPx+MWvFyfk9tynyxpLTFg5x2r63
...
-----END CERTIFICATE REQUEST-----
```

Issue Cancel

図24.18 ユーザー用証明書の発行

最後に **Issue** をクリックします。

これで新規に発行された証明書がユーザーの設定ページで見えるようになります。

第25章 VAULT を使用した認証情報の秘密の保存

vault とは、秘密を保存、取得、共有、および復旧するための安全な場所です。秘密とは、限定されたグループのメンバーまたはエンティティのみがアクセスを許可される機密データで、以下のものがあります。

- パスワード
- PIN
- プライベート SSH キー

ユーザーとサービスは、Identity Management (IdM) ドメインに登録されているいずれのマシンからでも vault に保存されている秘密にアクセスすることができます。



注記

Vault はコマンドラインからのみ利用可能で、IdM web UI からは利用できません。

vault のユースケースには以下のものがあります。

ユーザーの個人秘密の保存

詳細は「[ユーザー個人の秘密の保存](#)」を参照してください。

サービスの秘密の保存

詳細は「[Vault でのサービスの秘密の保存](#)」を参照してください。

複数ユーザーが使用する共通の秘密の保存

詳細は「[複数ユーザー用の共通の秘密の保存](#)」を参照してください。

vault を使用するには、「[Vault 使用における前提条件](#)」に記載の条件を満たす必要があることに注意してください。

25.1. VAULT の仕組み

25.1.1. Vault の所有者、メンバー、および管理者

IdM は以下の vault ユーザータイプを区別します。

Vault 所有者

vault 所有者とは、vault の基本的な管理権限があるユーザーもしくはサービスです。たとえば、vault 所有者は vault プロパティを変更したり、新規の vault メンバーを追加することができます。

各 vault には最低 1 人の所有者が必要ですが、複数の所有者がいても構いません。

Vault メンバー

vault メンバーとは、別のユーザーやサービスが作成した vault にアクセスできるユーザーまたはサービスです。

Vault 管理者

Vault 管理者にはすべての vault に対する無制限のアクセスがあり、vault の全操作が許可されます。



注記

対称および非対称 vault はパスワードかキーで保護されており、特別なアクセス制御ルールを適用します (「[標準、対称、および非対称 Vault](#)」を参照)。管理者は以下を実行するためにこれらのルールに合致する必要があります。

- 対称および非対称 vault にある秘密へのアクセス
- vault パスワードまたはキーの変更もしくはリセット

vault 管理者とは、**Vault Administrators** 権限のあるユーザーのことです。ユーザー権限の定義に関する情報は、「[ロールベースのアクセス制御の定義](#)」を参照してください。

特定の所有者およびメンバー権限は、vault のタイプに依存します。詳細は「[標準、対称、および非対称 Vault](#)」を参照してください。

Vault ユーザー

`ipa vault-show` などのコマンド出力では、ユーザー vaults の **Vault user** も表示されます。

```
$ ipa vault-show my_vault
Vault name: my_vault
Type: standard
Owner users: user
Vault user: user
```

vault ユーザーは、vault が格納されているコンテナを所有しているユーザーを表します。vault コンテナとユーザー vaults についての詳細は、「[Vault コンテナ](#)」と「[ユーザー、サービス、共有 Vault](#)」を参照してください。

25.1.2. 標準、対称、および非対称 Vault

vault はセキュリティーレベルとアクセス制御を基に、以下のタイプに分けられます。

標準 vault

Vault 所有者と vault メンバーは、パスワードやキーなしで秘密をアーカイブおよび取得することができます。

対称 vault

この vault の秘密は、対称キーで保護されます。Vault メンバーと vault 所有者は秘密のアーカイブと取得ができますが、vault パスワードを使用する必要があります。

非対称 vault

この vault の秘密は、非対称キーで保護されます。ユーザーは公開キーを使って秘密をアーカイブし、プライベートキーを使って秘密を取得します。vault メンバーは秘密のアーカイブしかできませんが、vault 所有者はアーカイブと取得の両方ができます。

25.1.3. ユーザー、サービス、共有 Vault

vault は所有権を基に、以下のタイプに分けられます。

ユーザー vault: ユーザーのプライベート vault

所有者は単一のユーザーです。

ユーザーはだれでも 1 つ以上のユーザー vault を所有することができます。

サービス vault: サービスのプライベート vault

所有者は単一のサービスです。

サービスはどれでも 1 つ以上のサービス vault を所有することができます。

共有 vault

所有者: この vault を作成した vault 管理者。他の vault 管理者もこの vault への完全なアクセスがあります。

共有 vault は複数のユーザーやサービスが使用することが可能です。

25.1.4. Vault コンテナ

vault コンテナとは、複数の vault の集合です。

IdM にはデフォルトで以下の vault コンテナがあります。

ユーザーコンテナ: ユーザーのプライベートコンテナ

このコンテナには、特定ユーザーのユーザー vault が保存されます。

サービスコンテナ: サービスのプライベートコンテナ

このコンテナには、特定サービスのサービス vault が保存されます。

共有コンテナ

このコンテナには、複数のユーザーやサービスが共有可能な vault が保存されます。

IdM は、ユーザーまたはサービスの最初のプライベート vault が作成される際に、自動的にユーザーまたはサービス向けのユーザー/サービスコンテナを作成します。ユーザーまたはサービスが削除されると、IdM はそのコンテナとコンテンツを削除します。

25.2. VAULT 使用における前提条件

vault を有効にするには、Key Recovery Authority (KRA) Certificate System コンポーネントを IdM ドメイン内のいずれかのサーバーにインストールします。

```
# ipa-kra-install
```

25.3. VAULT コマンドのヘルプ

vault および vault コンテナの管理に使用する全コマンドを表示するには、以下を実行します。

```
$ ipa help vault
```

特定コマンドの詳細なヘルプを表示するには、そのコマンドに **--help** オプションを加えて実行します。

```
$ ipa vault-add --help
```

Vault コマンドで **vault not found** エラーが出て失敗する場合

コマンドによっては、以下のオプションを使用して、vault の所有者もしくはタイプを指定する必要があります。

- **--user** または **--service** で、表示する vault の所有者を指定します。

```
$ ipa vault-show user_vault --user user
```

- **--shared** は、表示する vault が共有 vault であることを指定します。

たとえば、別のユーザーの vault を **--user** を追加せずに表示使用すると、IdM は vault が見つからないと返します。

```
[admin@server ~]$ ipa vault-show user_vault  
ipa: ERROR: user_vault: vault not found
```

25.4. ユーザー個人の秘密の保存

本セクションでは、ユーザーが 1 つ以上のプライベート vault を作成して個人の秘密を安全に保存する方法を説明します。保存後は、ドメイン内のマシンであれば、いずれのものからも必要に応じて秘密を取得することができます。たとえば、vault に個人の証明書をアーカイブすれば、集中化された場所に安全に証明書を保存したことになります。

本セクションには以下の手順があります。

- 「ユーザー個人の秘密のアーカイブ化」
- 「ユーザー個人の秘密の取得」

これらの手順では、以下を想定しています。

- **user** とは、vault を作成するユーザーです。
- **my_vault** とは、ユーザーの証明書を保存するための vault です。
- vault タイプは **standard** で、アーカイブ化された証明書にアクセスする際にユーザーは vault パスワードが必要ありません。
- **secret.txt** とは、ユーザーが vault に保存する証明書を格納しているファイルです。
- **secret_exported.txt** とは、アーカイブ化された証明書のエクスポート先となるファイルです。

25.4.1. ユーザー個人の秘密のアーカイブ化

ユーザーのプライベート vault を作成して証明書をそこに保存します。vault タイプは標準とし、証明書にアクセスする際に認証が不要とするようにします。

1. **user** としてログインします。

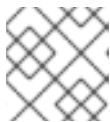
```
$ kinit user
```

2. **ipa vault-add** コマンドを使って標準 vault を作成します。

```
$ ipa vault-add my_vault --type standard
-----
Added vault "my_vault"
-----
Vault name: my_vault
Type: standard
Owner users: user
Vault user: user
```

3. **ipa vault-archive --in** コマンドを使って **secret.txt** ファイルを vault にアーカイブします。

```
$ ipa vault-archive my_vault --in secret.txt
-----
Archived data into vault "my_vault"
-----
```



注記

各 Vault が保存できるのは、シークレット 1 つのみです。

25.4.2. ユーザー個人の秘密の取得

プライベートの標準 vault から証明書をエクスポートします。

1. **user** としてログインします。

```
$ kinit user
```

2. **ipa vault-retrieve --out** コマンドを使って vault のコンテンツを取得し、それを **secret_exported.txt** ファイルに保存します。

```
$ ipa vault-retrieve my_vault --out secret_exported.txt
-----
Retrieved data from vault "my_vault"
-----
```

25.5. VAULT でのサービスの秘密の保存

本セクションでは、管理者が vault を使って集中化した場所にサービスの秘密を安全に保存する方法を説明します。サービスの秘密はサービスの公開キーで暗号化します。その後にサービスがこの秘密を取得する際は、ドメイン内のいずれかのマシンでサービスのプライベートキーを使用します。この秘密にアクセスできるのは、当該サービスと管理者のみです。

本セクションには以下の手順があります。

- 「サービスのパスワードを保存するユーザー Vault の作成」

- 「ユーザー Vault からサービスインスタンスへのサービスパスワードの提供」
- 「サービスインスタンス用のサービスパスワードの取得」
- 「サービス Vault のパスワード変更」

これらの手順では、以下を想定しています。

- **admin** とは、サービスのパスワードを管理する管理者です。
- **http_password** とは、管理者が作成しユーザーのプライベート vault の名前です。
- **password.txt** とは、サービスのパスワードを格納しているファイルです。
- **password_vault** とは、サービス用に作成された vault です。
- **HTTP/server.example.com** とは、そのパスワードがアーカイブ化されるサービスです。
- **service-public.pem** とは、**password_vault** に保存されているパスワードの暗号化に使用するサービスの公開キーです。

25.5.1. サービスのパスワードを保存するユーザー **Vault** の作成

管理者が所有するユーザー vault を作成し、サービスのパスワードを保存します。vault タイプは標準とし、管理者が vault のコンテンツにアクセスする際に認証が不要とするようにします。

1. 管理者としてログインします。

```
$ kinit admin
```

2. 標準のユーザー vault を作成します。

```
$ ipa vault-add http_password --type standard
-----
Added vault "http_password"
-----
Vault name: http_password
Type: standard
Owner users: admin
Vault user: admin
```

3. サービスパスワードを vault にアーカイブします。

```
$ ipa vault-archive http_password --in password.txt
-----
Archived data into vault "http_password"
-----
```

**警告**

パスワードを vault にアーカイブした後、システムから **password.txt** を削除します。

25.5.2. ユーザー **Vault** からサービスインスタンスへのサービスパスワードの提供

サービス用に作成された非対称 vault を使って、サービスパスワードをサービスインスタンスに提供します。

1. 管理者としてログインします。

```
$ kinit admin
```

2. サービスインスタンスの公開キーを取得します。たとえば、以下のように **openssl** ユーティリティーを使用します。

- a. **service-private.pem** プライベートキーを生成します。

```
$ openssl genrsa -out service-private.pem 2048
Generating RSA private key, 2048 bit long modulus
.+++
.....+++
e is 65537 (0x10001)
```

- b. このプライベートキーを基に、**service-public.pem** 公開キーを生成します。

```
$ openssl rsa -in service-private.pem -out service-public.pem -
pubout
writing RSA key
```

3. サービスインスタンスの vault として非対称 vault を作成し、公開キーを提供します。

```
$ ipa vault-add password_vault --service HTTP/server.example.com --
type asymmetric --public-key-file service-public.pem
-----
Added vault "password_vault"
-----
Vault name: password_vault
Type: asymmetric
Public key: LS0tLS1C...S0tLS0tCg==
Owner users: admin
Vault service: HTTP/server.example.com@EXAMPLE.COM
```

vault にアーカイブ化されたパスワードは、キーで保護されます。

4. 管理者のプライベート vault からサービスパスワードを取得し、これを新規サービスの vault にアーカイブします。

```
$ ipa vault-retrieve http_password --out password.txt
```

```
-----
Retrieved data from vault "http_password"
-----
```

```
$ ipa vault-archive password_vault --service HTTP/server.example.com
--in password.txt
```

```
-----
Archived data into vault "password_vault"
-----
```

これでパスワードは、サービスインスタンスの公開キーで暗号化されます。



警告

パスワードを vault にアーカイブした後、システムから **password.txt** を削除します。

パスワードを必要とするサービスインスタンスごとにこれらのステップを繰り返します。各サービスインスタンスごとに新たな非対称 vault を作成してください。

25.5.3. サービスインスタンス用のサービスパスワードの取得

サービスインスタンスは、ローカル保存のサービスのプライベートキーを使って、サービス vault のパスワードを取得することができます。

1. 管理者としてログインします。

```
$ kinit admin
```

2. サービス用の Kerberos チケットを取得します。

```
# kinit HTTP/server.example.com -k -t /etc/httpd/conf/ipa.keytab
```

3. サービス vault のパスワードを取得します。

```
$ ipa vault-retrieve password_vault --service
```

```
HTTP/server.example.com --private-key-file service-private.pem --out
password.txt
```

```
-----
Retrieved data from vault "password_vault"
-----
```

25.5.4. サービス **Vault** のパスワード変更

サービスインスタンスのセキュリティが侵害されたら、サービス vault のパスワードを変更し、セキュリティ侵害されていないサービスインスタンスにだけ新規パスワードを再度提供することで、これを切り離します。

1. 管理者のユーザー vault に新規パスワードをアーカイブします。

```
$ ipa vault-archive http_password --in new_password.txt
-----
Archived data into vault "http_password"
-----
```

これで vault に保存されている現行パスワードが上書きされます。

2. セキュリティが侵害されたインスタンス以外の各サービスインスタンスに新規パスワードを再提供します。

- a. 管理者の vault から新規パスワードを取得します。

```
$ ipa vault-retrieve http_password --out password.txt
-----
Retrieved data from vault "http_password"
-----
```

- b. 新規パスワードをサービスインスタンスの vault にアーカイブします。

```
$ ipa vault-archive password_vault --service
HTTP/server.example.com --in password.txt
-----
Archived data into vault "password_vault"
-----
```



警告

パスワードを vault にアーカイブした後、システムから **password.txt** を削除します。

25.6. 複数ユーザー用の共通の秘密の保存

本セクションでは、管理者が共有 vault を作成し、他のユーザーにこの vault 内の秘密へのアクセスを許可する方法を説明します。管理者は共通パスワードを vault にアーカイブし、他のユーザーはドメイン内のいずれかのマシンでもこのパスワードを取得できます。

本セクションには以下の手順があります。

- 「メンバーユーザーとして共有 Vault からの秘密の取得」
- 「共通の秘密がある共有 Vault の作成」

これらの手順では、以下を想定しています。

- **shared_vault** とは、共通パスワードを保存する vault です。
- **admin** とは、共有 vault を作成する管理者です。
- vault タイプは **standard** で、アーカイブ化されたパスワードにアクセスする際にユーザーは vault パスワードが必要ありません。
- **secret.txt** は、共通の秘密を格納しているファイルです。
- **user1** および **user2** は、vault へのアクセスを許可されているユーザーです。

25.6.1. 共通の秘密がある共有 **Vault** の作成

共有 vault を作成し、共通の秘密を保存します。この秘密にアクセスするユーザーを vault メンバーとして追加します。vault タイプは標準とし、秘密にアクセスする際に認証が不要とするようにします。

1. 管理者としてログインします。

```
$ kinit admin
```

2. 共有 vault を作成します。

```
$ ipa vault-add shared_vault --shared --type standard
-----
Added vault "shared_vault"
-----
Vault name: shared_vault
Type: standard
Owner users: admin
Shared vault: True
```

3. 秘密を vault にアーカイブします。 **--shared** オプションを追加して、vault が共有コンテナーにあるように指定します。

```
$ ipa vault-archive shared_vault --shared --in secret.txt
-----
Archived data into vault "shared_vault"
-----
```



注記

各 Vault が保存できるのは、シークレット 1 つのみです。

4. **user1** と **user2** を vault メンバーとして追加します。

```
ipa vault-add-member shared_vault --shared --users={user1,user2}
Vault name: shared_vault
Type: standard
Owner users: admin
Shared vault: True
Member users: user1, user2
```

```
-----  
Number of members added 2  
-----
```

25.6.2. メンバーユーザーとして共有 **Vault** からの秘密の取得

vault のメンバーユーザーとしてログインし、秘密のあるファイルを vault からエクスポートします。

1. **user1** メンバーユーザーとしてログインします。

```
$ kinit user1
```

2. 共有 vault から秘密を取得します。

```
$ ipa vault-retrieve shared_vault --shared --out secret_exported.txt  
-----  
Retrieved data from vault "shared_vault"  
-----
```

第26章 証明書と認証局の管理

26.1. 軽量のサブ証明局 (CA)

統合された Certificate System (CS) 証明局 (CA) で IdM システム環境が設定されている場合には、軽量のサブ CA を作成することができます。VPN (仮想プライベートネットワーク) など、特定のサブ CA が発行した証明書のみを受け入れるようにサービスを設定できます。同時に、別のサブ CA またはルート CA が発行した証明書だけを受け入れるように、他のサービスを設定することもできます。

サブ CA の中間証明書を破棄する場合には、このサブ CA で発行された証明書はすべて無効になります。

統合 CA で IdM を設定する場合は、自動的に作成された **ipa** CA が証明書システムのルート CA になります。作成するサブ CA はすべて、このルート CA に従属します。

26.1.1. 軽量のサブ証明局 (CA) の作成

サブ CA の作成に関する詳細は、以下を参照してください。

- [「Web UI からのサブ CA の作成」](#)
- [「コマンドラインからのサブ CA の作成」](#)

Web UI からのサブ CA の作成

`vpn-ca` という名前の新しいサブ CA を作成します。

1. **Authentication** タブを開き **Certificates** サブタブを選択します。
2. **Certificate Authorities** タブを選択します。**Add** をクリックします。
3. CA の名前とサブジェクト DN を入力します。



図26.1 CA の追加

サブジェクト DN は、IdM CA インフラストラクチャーで一意でなければなりません。

コマンドラインからのサブ CA の作成

`vpn-ca` という名前の新しいサブ CA を作成するには以下を入力します。

```
[root@ipaserver ~]# ipa ca-add vpn-ca --subject="CN=VPN,O=IDM.EXAMPLE.COM"
```

```
-----
Created CA "vpn-ca"
-----
```

```
Name: vpn-ca
Authority ID: ba83f324-5e50-4114-b109-acca05d6f1dc
Subject DN: CN=VPN,O=IDM.EXAMPLE.COM
Issuer DN: CN=Certificate Authority,O=IDM.EXAMPLE.COM
```

Name

CA の名前

認証局の ID

CA 用に自動的に作成される個別 ID

サブジェクト DN

サブジェクトの識別名 (DN)。IdM CA インフラストラクチャーで一意でなければなりません。

発行者 DN

サブ CA の証明書を発行した親 CA。サブ CA はすべて IdM ルート CA の子として作成されます。

新規の CA 署名証明書が正しく IdM のデータベースに追加されたことを確認するには、以下を実行します。

```
[root@ipaserver ~]# certutil -d /etc/pki/pki-tomcat/alias/ -L

Certificate Nickname           Trust
Attributes

SSL,S/MIME,JAR/XPI

caSigningCert cert-pki-ca      CTu,Cu,Cu
Server-Cert cert-pki-ca       u,u,u
auditSigningCert cert-pki-ca   u,u,Pu
caSigningCert cert-pki-ca ba83f324-5e50-4114-b109-acca05d6f1dc u,u,u
ocspSigningCert cert-pki-ca    u,u,u
subsystemCert cert-pki-ca      u,u,u
```



注記

証明書システムのインスタンスがインストールされると、新しい CA の証明書は、すべてのレプリカに自動的に転送されます。

26.1.2. 軽量のサブ CA の削除

サブ CA の削除に関する詳細は、以下を参照してください。

- [「Web UI からの サブ CA の削除」](#)
- [「コマンドラインからのサブ CA の削除」](#)

Web UI からの サブ CA の削除

1. **Authentication** タブを開き **Certificates** サブタブを選択します。
2. **Certificate Authorities** を選択します。
3. 削除するサブ CA を選択して **Delete** をクリックします。
4. **Delete** をクリックして確定します。

コマンドラインからのサブ CA の削除

サブ CA を削除するには、以下を入力します。

```
[root@ipaserver ~]# ipa ca-del vpn-ca
-----
Deleted CA "vpn-ca"
-----
```

26.2. 証明書の更新

以下の内容に関する説明は、ガイドを参照してください。

- 証明書の自動更新は「[証明書の自動更新](#)」を参照してください。
- 証明書の手動更新は「[手動での CA 証明書の更新](#)」を参照してください。

26.2.1. 証明書の自動更新

certmonger サービスは、期限が切れる 28 日前に、以下の証明書を自動的に更新します。

- ルート CA として IdM CA が発行した CA 証明書
- 内部 IdM サービスが使用する統合 IdM CA により発行されたサブシステムおよびサーバー証明書

サブ CA の CA 証明書を自動的に更新するには、**certmonger** のトラッキング一覧に追加されている必要があります。トラッキング一覧を更新するには、以下を実行します。

```
[root@ipaserver ~]# ipa-certupdate
trying https://idmserver.idm.example.com/ipa/json
Forwarding 'schema' to json server
'https://idmserver.idm.example.com/ipa/json'
trying https://idmserver.idm.example.com/ipa/json
Forwarding 'ca_is_enabled' to json server
'https://idmserver.idm.example.com/ipa/json'
Forwarding 'ca_find/1' to json server
'https://idmserver.idm.example.com/ipa/json'
Systemwide CA database updated.
Systemwide CA database updated.
The ipa-certupdate command was successful
```



注記

外部 CA をルート CA として使用する場合は、「[手動での CA 証明書の更新](#)」の説明のように、手動で証明書を更新する必要があります。**certmonger** サービスは、外部 CA が署名した証明書を自動的に更新できません。

certmonger が証明書の有効期限を監視する方法に関する情報は、『システムレベルの認証ガイド』の「[CERTMONGER を使った証明書の追跡](#)」を参照してください。

自動更新が予想どおりに機能していることを確認するには、`/var/log/messages` ファイルで **certmonger** のログメッセージを検証します。

- 証明書が更新されたら、**certmonger** には更新操作の成功または失敗を示す、以下のようなメッセージが記録されます。

```
Certificate named "NSS Certificate DB" in token "auditSigningCert
cert-pki-ca" in database "/var/lib/pki-ca/alias" renew success
```

- 証明書の有効期限が近づくと **certmonger** は以下のメッセージをログに記録します。

```
certmonger: Certificate named "NSS Certificate DB" in token
"auditSigningCert cert-pki-ca" in database "/var/lib/pki-ca/alias"
will not be valid after 20160204065136.
```

26.2.2. 手動での CA 証明書の更新

手作業で更新を行う場合は **ipa-cacert-manage** を使用することができます。

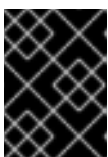
- 自己署名の IdM CA 証明書
- 外部署名の IdM CA 証明書

ipa-cacert-manage renew コマンドで更新された証明書は、以前の証明書と同じキーペアおよびサブジェクト名を使用します。証明書を更新しても、証明書のロールオーバーができるように、以前のバージョンは削除されません。

詳細は、`ipa-cacert-manage(1)` の man ページを参照してください。

26.2.2.1. 自己署名の IdM CA 証明書の手動更新

1. **ipa-cacert-manage renew** コマンドを実行します。証明書へのパスの指定は求められません。
2. 更新済み証明書が LDAP 証明書ストアと `/etc/pki/pki-tomcat/alias` NSS データベースに現れます。
3. 全サーバーおよびクライアントで **ipa-certupdate** ユーティリティーを実行して、LDAP からの新規証明書に関する情報でクライアントを更新します。全サーバーおよびクライアントで個別に **ipa-certupdate** を実行する必要があります。



重要

証明書を手動でインストールした後に **ipa-certupdate** を必ず実行します。実行しない場合には、証明書が他のマシンに配信されません。

更新した証明書が正しくインストールされていることを確認するには、**certutil** ユーティリティーを使用して、データベース内の証明書を表示します。以下に例を示します。

```
# certutil -L -d /etc/pki/pki-tomcat/alias
```

26.2.2.2. 外部署名の IdM CA 証明書の手動更新

1. **ipa-cacert-manage renew --external-ca** コマンドを実行します。
2. このコマンドでは **/var/lib/ipa/ca.crt** CSR ファイルを作成します。CSR を外部 CA に送信して、更新した CA 証明書を発行します。
3. もう一度 **ipa-cacert-manage renew** を実行し、今度は更新した認証局証明書と外部認証局の証明書チェーンファイルを **--external-cert-file** オプションを使って指定します。以下に例を示します。

```
# ipa-cacert-manage renew --external-cert-  
file=/tmp/servercert20110601.pem --external-cert-  
file=/tmp/cacert.pem
```

4. これで更新した認証局証明書と外部認証局のチェーンが LDAP 証明書ストア内と **/etc/pki/pki-tomcat/alias/** NSS データベースに表示されるようになります。
5. 全サーバーおよびクライアントで **ipa-certupdate** ユーティリティーを実行して、LDAP からの新規証明書に関する情報でクライアントを更新します。全サーバーおよびクライアントで個別に **ipa-certupdate** を実行する必要があります。



重要

証明書を手動でインストールした後に **ipa-certupdate** を必ず実行します。実行しない場合には、証明書が他のマシンに配信されません。

更新した証明書が正しくインストールされていることを確認するには、**certutil** ユーティリティーを使用して、データベース内の証明書を表示します。以下に例を示します。

```
# certutil -L -d /etc/pki/pki-tomcat/alias/
```

26.3. CA 証明書の手動インストール

新規の証明書を IdM にインストールするには、**ipa-cacert-manage install** コマンドを使用します。たとえば、このコマンドでは、有効期限に近づくとき現在の証明書を変更できるようになります。

1. **ipa-cacert-manage install** コマンドを実行して、証明書を含んでいるファイルへのパスを指定します。このコマンドは PEM 形式の証明書ファイルを受け付けます。

```
[root@server ~]# ipa-cacert-manage install /etc/group/cert.pem
```

これで証明書が LDAP 証明書ストアに置かれました。

2. 全サーバーおよびクライアントで **ipa-certupdate** ユーティリティーを実行して、LDAP からの新規証明書に関する情報でクライアントを更新します。全サーバーおよびクライアントで個別に **ipa-certupdate** を実行する必要があります。



重要

証明書を手動でインストールした後に **ipa-certupdate** を必ず実行します。実行しない場合には、証明書が他のマシンに配信されません。

ipa-cacert-manage install コマンドは、以下のオプションを取ります。

-n

これは証明書のニックネームを提供します。デフォルト値は、証明書のサブジェクト名になります。

-t

これは、**certutil** 形式で証明書の信頼フラグを指定します。デフォルト値は、**C,,** です。信頼フラグを指定する形式についての情報は、ipa-cacert-manage(1) man ページを参照してください。

26.4. 証明書チェーンの変更

証明書チェーンの変更は、**ipa-cacert-manage renew** を使用して CA を更新します。

自己署名の CA 証明書 → 外部署名の CA 証明書

--external-ca オプションを **ipa-cacert-manage renew** コマンドに追加します。これにより、自己署名の CA 証明書を外部署名の CA 証明書に更新します。

このオプションを使用したコマンドの詳細は、「[手動での CA 証明書の更新](#)」を参照してください。

外部 CA 証明書 → 自己署名 CA 証明書

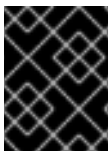
--self-signed オプションを **ipa-cacert-manage renew** に追加します。これにより、外部署名の CA 証明書を自己署名の CA 証明書に更新します。

26.5. IDM を有効期限の切れた証明書で起動できるようにする方法

IdM の管理サーバー証明書の期限が切れると、多くの IdM サービスにはアクセスできなくなります。基盤の Apache や LDAP サービスが証明書の期限が切れていても、サービスに SSL アクセスできるように設定できます。

期限切れの証明書でアクセスを制限できるようにする場合:

- Apache、Kerberos、DNS および LDAP サービスは継続して機能します。これらのサービスがアクティブな場合は、ユーザーは IdM ドメインにログインできます。
- アクセスに SSL が必要なクライアントサービスが依然として失敗します。たとえば、IdM クライアントで SSSD が必要で、SSSD は IdM に問い合わせるには SSL が必要であるため **sudo** に失敗します。



重要

この手順は一時的な回避策です。できるだけ早く必要な証明書を更新して、記載の変更を元に戻します。

1. Apache サーバーの **mod_nss** が有効な証明書を強制的に有効化しないように設定します。

a. **/etc/httpd/conf.d/nss.conf** ファイルを開きます。

b. **NSSEnforceValidCerts** パラメーターを **off** に設定します。

■


```
NSSEnforceValidCerts off
```

2. Apache を再起動します。

```
# systemctl restart httpd.service
```

3. LDAP ディレクトリーサーバーの有効性チェックが無効になっていることを確認します。これには、**nsslapd-validate-cert** 属性が **warn** に設定されていることを確認します。

```
# ldapsearch -h server.example.com -p 389 -D "cn=directory manager"
-w secret -LLL -b cn=config -s base "(objectclass=*)" nsslapd-
validate-cert
```

```
dn: cn=config
nsslapd-validate-cert: warn
```

属性が **warn** に設定されていない場合には、変更してください。

```
# ldapmodify -D "cn=directory manager" -w secret -p 389 -h
server.example.com
```

```
dn: cn=config
changetype: modify
replace: nsslapd-validate-cert
nsslapd-validate-cert: warn
```

4. ディレクトリーサーバーを再起動します。

```
# systemctl restart dirsrv.target
```

26.6. HTTP または LDAP 用のサードパーティー証明書のインストール

Apache Web サーバー、Directory Server または両サーバー用に新規 SSL サーバーをインストールすると、現在の SSL 証明書が新しい証明書に置き換えられます。これには、以下が必要です。

- SSL 秘密鍵 (以下の手順の **ssl.key**)
- SSL 証明書 (以下の手順の **ssl.crt**)

鍵および証明書の対応形式に関する一覧は ipa-server-certinstall(1) の man ページを参照してください。

前提条件

ssl.crt 証明書は、証明書を読み込むサーバーが認識している CA で署名されている必要があります。そうでない場合には、「[CA 証明書の手動インストール](#)」の記載どおりに **ssl.crt** を署名した CA の CA 証明書を IdM にインストールします。

これにより、IdM は CA を認識し、**ssl.crt** を受け入れます。

サードパーティー証明書のインストール

1. **ipa-server-certinstall** ユーティリティを使用して、証明書をインストールします。インストール先を指定します。

- **--http** は Apache Web サーバーに証明書をインストールします。
- **--dirsrv** は Directory Server に証明書をインストールします。

たとえば、SSL 証明書を両サーバーにインストールするには以下を実行します。

```
# ipa-server-certinstall --http --dirsrv ssl.key ssl.crt
```

2. サーバーが証明書のインストール先で立ち上がるように起動します。

- Apache Web サーバーの再起動

```
# systemctl restart httpd.service
```

- Directory Server の再起動

```
# systemctl restart dirsrv@REALM.service
```

3. 証明書が正しくインストールされていることを確認するには、証明書のデータベースに証明書が存在していることを確認します。

- Apache 証明書データベースを表示します。

```
# certutil -L -d /etc/httpd/alias
```

- Directory Server 証明書データベースを表示します。

```
# certutil -L -d /etc/dirsrv/slapd-REALM/
```

26.7. OCSF 応答の設定

IdM サーバーに統合されている CA すべては、内部のオンライン証明書状態プロトコル (OCSF: Online Certificate Status Protocol) レスポンダーを使用します。OCSF レスポンダーへのアクセスを許可する IdM サービスは **<http://ca-server.example.com/ca/ocsp>** で入手できます。クライアントは、この URL に接続して証明書の有効性を確認できます。



注記

OCSF の詳細は、Red Hat Certificate System のドキュメントを参照してください。たとえば、『Planning, Installation, and Deployment Guide』の「[2.2.4. Revoking Certificates and Checking Status](#)」を参照してください。

26.7.1. CRL 更新間隔の変更

デフォルトでは、IdM CA により CRL ファイルは 4 時間ごとに自動的に生成されます。この間隔を変更するには以下を行います。

1. 認証局サーバーを停止します。

```
# systemctl stop pki-tomcatd@pki-tomcat.service
```

2. `/var/lib/pki/pki-tomcat/conf/ca/CS.cfg` ファイルを開き、`ca.crl.MasterCRL.autoUpdateInterval` の値を新しい間隔設定に変更します。たとえば、CRL を 60 分ごとに生成するには以下を実行します。

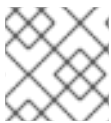
```
ca.crl.MasterCRL.autoUpdateInterval=60
```

3. CA サーバーを起動します。

```
# systemctl start pki-tomcatd@pki-tomcat.service
```

26.8. CA の既存の IDM ドメインへのインストール

IdM ドメインが証明局 (CA) なしでインストールされた場合には、後で CA サービスをインストールできます。環境によっては、IdM 証明書サーバーの CA をインストールすることも、外部の CA を使用することもできます。



注記

サポートされる CA 設定に関する詳細は「[CA 設定の決定](#)」を参照してください。

IdM 証明書サーバー

1. 以下のコマンドを使用して IdM 証明書サーバーの CA をインストールします。

```
[root@ipa-server ~] ipa-ca-install
```

2. 全サーバーおよびクライアントで **ipa-certupdate** ユーティリティを実行して、LDAP からの新規証明書に関する情報でクライアントを更新します。全サーバーおよびクライアントで個別に **ipa-certupdate** を実行する必要があります。



重要

証明書を手動でインストールした後に **ipa-certupdate** を必ず実行します。実行しない場合には、証明書が他のマシンに配信されません。

外部 CA

外部 CA を後でインストールする際には、複数の手順を踏む必要があります。

1. インストールを開始します。

```
[root@ipa-server ~] ipa-ca-install --external-ca
```

この手順を終えると、証明書署名要求 (CSR) が保存された旨の情報が表示されます。CSR を外部 CA に送信して、発行した証明書を IdM サーバーにコピーします。

2. 証明書と外部 CA への完全なパスを **ipa-ca-install** に渡して、インストールを続行します。

```
[root@ipa-server ~]# ipa-ca-install --external-cert-file=/root/master.crt --external-cert-file=/root/ca.crt
```

3. 全サーバーおよびクライアントで **ipa-certupdate** ユーティリティーを実行して、LDAP からの新規証明書に関する情報でクライアントを更新します。全サーバーおよびクライアントで個別に **ipa-certupdate** を実行する必要があります。



重要

証明書を手動でインストールした後に **ipa-certupdate** を必ず実行します。実行しない場合には、証明書が他のマシンに配信されません。

CA をインストールしても、LDAP および Web サーバーの既存のサービス証明書は、新しくインストールした CA の証明書には置き換えられません。証明書を置き換える方法は、「[Web サーバーおよび LDAP サーバーの証明書の置き換え](#)」を参照してください。

26.9. WEB サーバーおよび LDAP サーバーの証明書の置き換え

Web サーバーおよび LDAP サーバーのサービス証明書を置き換えます。

1. 以下を使用して、新しい証明書を要求します。

- 統合 CA: 詳細は「[ユーザー、ホスト、またはサービス向けの新規証明書のリクエスト](#)」を参照してください。
- 外部 CA: 秘密鍵と証明書署名要求 (CSR) を生成します。以下の例では、OpenSSL を使用します。

```
$ openssl req -new -newkey rsa:2048 -days 365 -nodes -keyout
new.key -out new.csr -subj
'/CN=idmserver.idm.example.com,O=IDM.EXAMPLE.COM'
```

CSR を外部の CA に送信します。このプロセスは、外部 CA として使用するサービスにより異なります。

2. Apache Web サーバーの秘密鍵と証明書を置き換えます。

```
[root@ipaserver ~]# ipa-server-certinstall -w --pin=password new.key
new.crt
```

3. LDAP サーバーの秘密鍵と証明書を置き換えます。

```
[root@ipaserver ~]# ipa-server-certinstall -d --pin=password new.key
new.cert
```

パート **VI.** ポリシーの管理

第27章 パスワードポリシーの定義

本章では、Identity Management (IdM) でのパスワードポリシーとその管理方法について説明します。

27.1. パスワードポリシーのその役割

パスワードポリシーとはパスワードが満たす必要のあるルールセットです。

たとえば、パスワードポリシーでは、パスワードの最小限の長さや有効期間を定義することができます。このポリシーに影響を受けるユーザーはすべて、この長さの要件を満たし、有効期間が過ぎる前にパスワードを変更する必要があります。

パスワードポリシーにより、ユーザーのパスワードがだれかに見つけられたり、不正使用されることを防ぐことができます。

27.2. IDM におけるパスワードポリシー

ユーザーはすべて、Identity Management (IdM) Kerberos ドメインに認証するためのパスワードが必要になります。IdM でのパスワードポリシーでは、これらのユーザーのパスワードが満たす必要のある要件を定義します。



注記

IdM パスワードポリシーは基礎となる LDAP ディレクトリーで設定されますが、Kerberos Key Distribution Center (KDC) でも強制実行されます。

27.2.1. サポートされるパスワードポリシーの属性

表27.1「パスワードポリシーの属性」では、IdM のパスワードポリシーで定義可能な属性を示しています。

表27.1 パスワードポリシーの属性

属性	説明	例
Max lifetime	パスワードのリセットが必要になるまでの、パスワードの最長有効期間を日数単位で設定します。	Max lifetime = 90 ユーザーのパスワードは 90 日間有効です。この後は、IdM がパスワード変更をユーザーにプロンプトします。
Min lifetime	一旦パスワードを変更してから再度変更可能となるまでのパスワードの最小有効期間を時間単位で設定します。	Min lifetime = 1 ユーザーは一旦パスワードを変更すると、次に変更可能となるまで少なくとも 1 時間待つ必要があります。
History size	保存する以前のパスワード数を設定します。ユーザーは、パスワード履歴にあるパスワードは使用できません。	History size = 0 ユーザーは以前のいずれのパスワードも使用できます。

属性	説明	例
Character classes	<p>パスワードで使用する必要のある文字クラスの数。文字クラスとは、以下を指します。</p> <ul style="list-style-type: none"> 大文字 小文字 数字 コンマ (,), ピリオド (.), アスタリスク (*) などの特殊文字 他の UTF-8 文字 <p>同じ文字を 3 つ以上続けて使用すると、文字クラス数は 1 つ減ることになります。たとえば、</p> <ul style="list-style-type: none"> Secret1 には、大文字、小文字、数字の 3 つの文字クラスがあります。 Secret111 では 1 を続けて使用したことから、大文字、小文字、数字の 3 つの文字クラスから -1 のペナルティーを課した 2 つの文字クラスとなります。 	<p>Character classes = 0</p> <p>デフォルトのクラス数は 0 です。この数字を設定するには、ipa pwpolicy-mod コマンドを --minclasses オプションと実行します。このコマンドは、必須文字クラス数を 1 に設定します。</p> <pre>\$ ipa pwpolicy-mod --minclasses=1</pre> <p>本テーブル下の 重要 も参照してください。</p>
Min length	パスワードで使用する最低文字数	<p>Min length = 8</p> <p>8 文字未満のパスワードは使用できません。</p>
Max failures	<p>実行可能なログインの最大試行回数。失敗回数がこの値を超えた場合には、ユーザーアカウントがロックされます。「ログイン失敗後のユーザーアカウントのロック解除」も参照してください。</p>	<p>Max failures = 6</p> <p>ユーザーが 7 回連続で間違ったパスワードを入力すると、IdM はこのユーザーアカウントをロックします。</p>
Failure reset interval	失敗したログイン試行回数を IdM がリセットするまでの時間を秒単位で設定します。	<p>Failure reset interval = 60</p> <p>Max failures で設定した回数のログイン試行に失敗した後に 1 分間以上待機すると、ユーザーはアカウントをロックすることなく再度ログイン試行することができます。</p>
Lockout duration	<p>Max failures で定義された回数のログイン試行に失敗した後、ユーザーアカウントがロックされる時間を秒単位で設定します。「ログイン失敗後のユーザーアカウントのロック解除」も参照してください。</p>	<p>Lockout duration = 600</p> <p>ユーザーはアカウントがロックされると、10 分間ログインできません。</p>



重要

国際文字や記号に対応していないハードウェアを使用している場合は、英数字および一般的な記号を文字クラスに使用してください。パスワードの文字クラスポリシーに関する詳細は、Red Hat ナレッジベースの「[What characters are valid in a password?](#)」を参照してください。

27.2.2. グローバルおよびグループ固有のパスワードポリシー

デフォルトのパスワードポリシーは、**グローバルパスワードポリシー** です。これ以外では、新たに **グループパスワードポリシー** を設定できます。

グローバルパスワードポリシー

最初の IdM サーバーをインストールすると、自動的にグローバルパスワードポリシーがデフォルト設定で作成されます。

グローバルポリシールールは、グループパスワードポリシーがない全ユーザーに適用されます。

グループパスワードポリシー

グループパスワードポリシーは、対応するユーザーグループの全メンバーに適用されます。

あるユーザーに一度に適用されるのは、1 つのパスワードポリシーのみです。ユーザーに複製のパスワードポリシーが割り当てられている場合は、優先度をベースにそのうちのいずれかが適用されます。「[パスワードポリシーの優先度](#)」を参照してください。

27.2.3. パスワードポリシーの優先度

グループパスワードポリシーには、**優先度** セットがあります。この値が低ければ低いほど、そのポリシーの優先度は高くなります。最低値は **0** です。

- 複数のパスワードポリシーがあるユーザーに適用可能な場合、優先度の値が最も低いポリシーが優先されます。他のポリシーで定義されているルールはすべて無視されます。
- 優先度の値が最も低いポリシーは、属性が定義されていないものでも、このポリシーが全パスワードポリシー属性に適用されます。

グローバルパスワードポリシーには優先度の値が設定されていません。これは、ユーザーにグループポリシーが設定されていない場合のフォールバックポリシーとして機能します。グローバルポリシーがグループポリシーよりも優先されることはありません。

表27.2「[優先度をベースにしたパスワードポリシー属性の適用例](#)」では、ポリシーが定義された 2 つのグループにユーザーが属する場合に、パスワードポリシーの優先度がどのように機能するかを示しています。

表27.2 優先度をベースにしたパスワードポリシー属性の適用例

	Max lifetime	Min length
グループ A のポリシー (優先度 0)	60	10
グループ B のポリシー (優先度 1)	90	0 (制限なし)

	Max lifetime	Min length
	↓	↓
ユーザー (グループ A とグループ B のメンバー)	60	10



注記

ipa pwpolicy-show --user=user_name コマンドでは、現在どのポリシーが特定のユーザーに適用されているかを示します。

27.3. 新規パスワードポリシーの追加

新規パスワードポリシーを追加する際には、以下を指定します。

- ポリシーの適用先となるユーザーグループ (「[グローバルおよびグループ固有のパスワードポリシー](#)」を参照)
- 優先度 (「[パスワードポリシーの優先度](#)」を参照)

新規パスワードポリシーは、以下のいずれかの方法で追加できます。

- web UI の場合は、「[Web UI: 新規パスワードポリシーの追加](#)」を参照してください。
- コマンドラインの場合は、「[コマンドライン: 新規パスワードポリシーの追加](#)」を参照してください。

Web UI: 新規パスワードポリシーの追加

1. **Policy → Password Policies** を選択します。
2. **Add** をクリックします。
3. ユーザーグループと優先度を定義します。
4. **Add** をクリックして確定します。

新規パスワードポリシーの属性の設定方法については、「[パスワードポリシー属性の編集](#)」を参照してください。

コマンドライン: 新規パスワードポリシーの追加

1. **ipa pwpolicy-add** コマンドで、ユーザーグループと優先度を指定します。

```
$ ipa pwpolicy-add
Group: group_name
Priority: priority_level
```

2. オプションで、**ipa pwpolicy-find** コマンドを使用してポリシーが正常に追加されたことを確認できます。

```
$ ipa pwpolicy-find
```

新規パスワードポリシーの属性の設定方法については、「[パスワードポリシー属性の編集](#)」を参照してください。

27.4. パスワードポリシー属性の編集



重要

パスワードポリシーを変更する際には、新規ルールは新たなパスワードにのみ適用されます。変更が遡及的に既存のパスワードに適用されることはありません。

変更が適用されるには、既存のパスワードを変更するか、管理者が他のユーザーのパスワードをリセットする必要があります。「[ユーザーパスワードの変更およびリセット](#)」を参照してください。



注記

安全なユーザーパスワードについての推奨例は、『セキュリティガイド』の[パスワードのセキュリティ](#)を参照してください。

パスワードポリシーは、以下のいずれかの方法で修正できます。

- web UI の場合は、「[Web UI: パスワードポリシーの修正](#)」を参照してください。
- コマンドラインの場合は、「[コマンドライン: パスワードポリシーの修正](#)」を参照してください。

パスワードポリシー属性を **0** に設定すると、属性の制限がなくなります。たとえば、maximum lifetime を **0** に設定すると、パスワードの有効期限がなくなります。

Web UI: パスワードポリシーの修正

1. **Policy → Password Policies** を選択します。
2. 変更するポリシーを修正します。
3. 必要な属性を更新します。利用可能な属性の詳細については、「[サポートされるパスワードポリシーの属性](#)」を参照してください。
4. **Save** をクリックして変更を保存します。

コマンドライン: パスワードポリシーの修正

1. **ipa pwpolicy-mod** コマンドを使用してポリシーの属性を変更します。
 - a. たとえば、グローバルパスワードポリシーを更新してパスワードの最小限の長さを **10** に設定するには、以下を実行します。

```
$ ipa pwpolicy-mod --minlength=10
```

- b. グループポリシーを更新するには、グループ名を **ipa pwpolicy-mod** に追加します。

```
$ ipa pwpolicy-mod group_name --minlength=10
```

2. オプションで、**ipa pwpolicy-show** コマンドを使用すると、新規ポリシー設定を確認することができます。

- a. グローバルポリシーを表示するには、以下を実行します。

```
$ ipa pwpolicy-show
```

- b. グループポリシーを表示するには、グループ名を **ipa pwpolicy-show** に追加します。

```
$ ipa pwpolicy-show group_name
```

27.5. パスワードの有効期限を変更して即座に反映させる

IdM は、既存のパスワードを変更するか、ユーザーが新規パスワードを入力する際にパスワードポリシールールを適用します。「[パスワードポリシー属性の編集](#)」を参照してください。

ユーザーパスワードの有効期限の変更を即座に反映させるには、LDAP で **krbPasswordExpiration** の属性値を変更します。単一ユーザーの場合、以下のようにします。

1. **ldapmodify** ユーティリティーを使用します。

```
# ldapmodify -D "cn=Directory Manager" -w secret -h
server.example.com -p 389 -vv

dn: uid=user_name,cn=users,cn=accounts,dc=example,dc=com
changetype: modify
replace: krbPasswordExpiration
krbPasswordExpiration: 20160203203734Z
```

krbPasswordExpiration フォーマットは以下のテンプレートに従っています。

- 。 年 (2016)
- 。 月 (02)
- 。 日 (03)
- 。 現在の時刻 (20:37:34)
- 。 タイムゾーン (Z)

2. **Ctrl+D** を押して変更をサーバーに送信します。

複数のエントリーを同時に編集するには、**-f** オプションを **ldapmodify** で使用して LDIF ファイルを参照します。

第28章 KERBEROS ドメインの管理

本章では Identity Management サーバーの Kerberos Key Distribution Center (KDC) コンポーネントの管理について説明します。



重要

Identity Management の Kerberos ポリシー管理には、**kadmin** または **kadmin.local** ユーティリティーを使用しないでください。本ガイドに記載のように、ネイティブの Identity Management コマンドツールを使用してください。

上記の Kerberos ツールを使用して Identity Management ポリシーの管理を試みると、操作によっては Directory Server インスタンスに保存されている Identity Management の設定に影響が及ばないものもあります。

28.1. KERBEROS チケットポリシーの管理

Identity Management の Kerberos チケットポリシーは、チケットの期間や更新に関する制約を設定します。以下の手順を使用して、Identity Management サーバー上で実行中の Kerberos Key Distribution Center (KDC) の Kerberos チケットポリシーを設定できます。

28.1.1. グローバルおよびユーザー固有の **Kerberos** チケットポリシー

グローバルの Kerberos チケットポリシーを再定義して、個人ユーザー専用の追加のポリシーを定義することができます。

グローバルの **Kerberos** チケットポリシー

グローバルポリシーは、Identity Management の Kerberos レルム内で発行された全チケットに適用されます。

ユーザー固有の **Kerberos** チケットポリシー

ユーザー固有のポリシーは、関連付けられたユーザーアカウントに対してのみ適用されます。たとえば、ユーザー固有のチケットポリシーにより、**admin** ユーザーのチケットの最大有効期間を延長するように定義できます。

ユーザー固有のポリシーは、グローバルポリシーよりも優先されます。

28.1.2. グローバルの **Kerberos** チケットポリシーの設定

グローバルの Kerberos チケットポリシーを設定するには、以下を使用できます。

- Identity Management の Web UI: 「[Web UI: グローバルのKerberos チケットポリシーの設定](#)」を参照してください。
- コマンドライン: 「[コマンドライン: グローバルのKerberos チケットポリシーの設定](#)」を参照してください。

表28.1 サポートされる **Kerberos** チケットポリシーの属性

属性	説明	例
Max renew	Kerberos チケットの有効期限が切れてからチケットを更新できる期間 (秒)。更新期間が過ぎると、ユーザーは kinit ユーティリティを使用してログインし、新規チケットを取得する必要があります。 チケットを更新するには kinit -R コマンドを使用します。	Max renew = 604800 チケットの期限が切れると、ユーザーは 7 日以内 (604,800 秒) であればチケットを更新できます。
Max life	Kerberos チケットの有効期間 (秒)。Kerberos チケットが有効な期間。	Max life = 86400 チケットは発行されてから 24 時間 (86,400 秒) で期限が切れます。

Web UI: グローバルのKerberos チケットポリシーの設定

1. **Policy** → **Kerberos Ticket Policy** を選択します。
2. 必須値を定義します。
 - a. **Max renew** フィールドで、Kerberos チケットの最大更新期間を入力します。
 - b. **Max life** フィールドで、Kerberos チケットの最大有効期間を入力します。

図28.1 グローバルの Kerberos チケットポリシーの設定

3. **保存** をクリックします。

コマンドライン: グローバルのKerberos チケットポリシーの設定

グローバルの Kerberos チケットポリシーを変更します。

- **ipa krbtpolicy-mod** コマンドを使用して、最低でも以下のオプションを 1 つ指定します。
 - **--maxrenew** は、Kerberos チケットの最大更新期間を定義します。
 - **--maxlife** は、Kerberos チケットの最大有効期間を定義します。

たとえば、最大有効期間を変更するには、以下を実行します。

```
$ ipa krbtpolicy-mod --maxlife=80000
Max life: 80000
Max renew: 604800
```

元のデフォルト値に、グローバルの Kerberos チケットポリシーをリセットします。

1. **ipa krbtpolicy-reset** コマンドを使用します。
2. オプションで、**ipa krbtpolicy-show** コマンドを使用して、現在の設定を確認します。

ipa krbtpolicy-mod および **ipa krbtpolicy-reset** の詳細は、コマンドの実行時に **--help** のオプションを指定します。

28.1.3. ユーザー固有の Kerberos チケットポリシーの設定

特定ユーザーの Kerberos チケットポリシーを変更します。

1. **ipa krbtpolicy-mod user_name** コマンドを使用して、最低でも以下のオプションを 1 つ 指定します。
 - **--maxrenew** は、Kerberos チケットの最大更新期間を定義します。
 - **--maxlife** は、Kerberos チケットの最大有効期間を定義します。

属性 1 つのみを定義した場合には、Identity Management は、他の属性にグローバルの Kerberos チケットポリシーを適用します。

たとえば **admin** ユーザーの最大有効期間を変更するには、以下を実行します。

```
$ ipa krbtpolicy-mod admin --maxlife=160000
Max life: 80000
Max renew: 604800
```

2. オプションで、**ipa krbtpolicy-show user_name** コマンドを使用して、特定のユーザーの 現在値を表示します。

新しいポリシーは、**kinit** ユーティリティーの使用時など、次の Kerberos チケットの要求時に即座にチケットに適用されます。

ユーザー固有の Kerberos チケットポリシーをリセットするには、**ipa krbtpolicy-reset user_name** コマンドを使用します。このコマンドは、Identity Management がグローバルポリシーの値を適用してからユーザーに定義された値を消去します。

ipa krbtpolicy-mod および **ipa krbtpolicy-reset** の詳細は、コマンドの実行時に **--help** のオプションを指定します。

28.2. KERBEROS プリンシパルの鍵の変更

Kerberos プリンシパルの **鍵**を変更すると、プリンシパルの Keytab に、現在のキーバージョン番号 (KVNO) よりも高い値を新規 Keytab エントリーとして追加します。

1. 指定の期間内に発行された keytab をすべて検索します。たとえば、以下のコマンドでは、**ldapsearch** ユーティリティーを使用して、グリニッジ標準時 (GMT) の 2016 年 1 月 1

日午前 12 時から 12 月 31 日午後 11 時 59 分までに作成された全ホストとサービスプリンシパルを表示します。

```
# ldapsearch -x -b "cn=computers,cn=accounts,dc=example,dc=com" "(&
(krblastpwdchange>=20160101000000)
(krblastpwdchange<=20161231235959))" dn krbprincipalname
```

```
# ldapsearch -x -b "cn=services,cn=accounts,dc=example,dc=com" "(&
(krblastpwdchange>=20160101000000)
(krblastpwdchange<=20161231235959))" dn krbprincipalname
```

- 検索ベース (**-b**) では、**ldapsearch** がプリンシパルを検索するサブツリーを定義します。
 - ホストプリンシパルは、**cn=computers,cn=accounts,dc=example,dc=com** サブツリーの配下に保存されます。
 - サービスプリンシパルは、**cn=services,cn=accounts,dc=example,dc=com** サブツリーの配下に保存されます。
 - **krblastpwdchange** パラメーターは、最終変更日で検索結果を絞り込みます。このパラメーターは、日付には YYYYMMDD の形式、日次には HHMMSS の形式に対応しています (グリニッジ標準時)。
 - **dn** および **krbprincipalname** 属性を指定すると、検索結果をエントリー名とプリンシパルのみに絞り込みます。
2. サービスおよびホストがプリンシパルの鍵の再生成が必要な場合には、**ipa-getkeytab** ユーティリティーを使用して新規 keytab エントリーを取得します。以下のオプションを渡します。
- **--principal (-p)**: プリンシパルを指定します。
 - **--keytab (-k)**: 元の keytab の場所を指定します。
 - **--server (-s)**: Identity Management サーバーのホスト名を指定します。

例を示します。

- デフォルトの場所である **/etc/krb5.keytab** にある keytab のホストプリンシパルの鍵を再生成するには、以下を実行します。

```
# ipa-getkeytab -p host/client.example.com@EXAMPLE.COM -s
server.example.com -k /etc/krb5.keytab
```

- デフォルトの場所である **/etc/httpd/conf/ipa.keytab** にある Apache サービスの keytab の鍵を再生成するには以下を実行します。

```
# ipa-getkeytab -p HTTP/client.example.com@EXAMPLE.COM -s
server.example.com -k /etc/httpd/conf/ipa.keytab
```



重要

NFS バージョン 4 などのサービスは、一部の暗号化タイプのみをサポートします。**ipa-getkeytab** コマンドに適切な引数を渡して keytab を設定します。

3. **オプション:** プリンシパルの鍵が正しく再生成されたことを確認します。**klist** ユーティリティーを使用して、すべての Kerberos チケットを表示します。たとえば、**/etc/krb5.keytab** の keytab エントリーを表示するには以下を実行します。

```
# klist -kt /etc/krb5.keytab
Keytab: WRFILE:/etc/krb5.keytab
KVNO Timestamp          Principal
-----
-----
1 06/09/16 05:58:47 host/client.example.com@EXAMPLE.COM(aes256-
cts-hmac-sha1-96)
2 06/09/16 11:23:01 host/client.example.com@EXAMPLE.COM(aes256-
cts-hmac-sha1-96)
1 03/09/16 13:57:16 krbtgt/EXAMPLE.COM@EXAMPLE.COM(aes256-cts-
hmac-sha1-96)
1 03/09/16 13:57:16 HTTP/server.example.com@EXAMPLE.COM(aes256-
cts-hmac-sha1-96)
1 03/09/16 13:57:16 ldap/server.example.com@EXAMPLE.COM(aes256-
cts-hmac-sha1-96)
```

この出力では、**client.example.com** の keytab エントリーが前の KVNO よりも大きい数値を使用して鍵が再生成されたことを示します。以前の KVNO が付いた元の keytab は、そのままデータベースに残ります。

以前の keytab に対して発行されたチケットは機能し続け、KVNO の値が最大の鍵で新規チケットが発行されるため、システム操作の中断を防ぎます。

28.3. KEYTAB の保護

サーバーにアクセスできる他のユーザーから kerberos の keytab を保護するには、keytab の所有者だけに keytab へのアクセスを制限します。keytab の取得後には keytab の権限を保護することを推奨します。

たとえば、**/etc/httpd/conf/ipa.keytab** の Apache keytab を保護するには以下を行います。

1. **apache** にファイルの所有者を設定します。

```
# chown apache /etc/httpd/conf/ipa.keytab
```

2. ファイルのパーミッションを **0600** に設定します。これは、所有者に読み取り、書き込み、実行の権限を割り当てます。

```
# chmod 0600 /etc/httpd/conf/ipa.keytab
```

28.4. KEYTAB の削除

ホストの登録解除や再登録、kerberos の接続エラーなどの場合には、keytab を削除して、新規 keytab を作成する必要があります。

ホストの全 keytab を削除するには **ipa-rmkeytab** ユーティリティを使用してこれらのオプションを渡します。

- **--realm (-r)**: Kerberos レalmを指定します。
- **--keytab (-k)**: keytab ファイルへのパスを指定します。

```
# ipa-rmkeytab --realm EXAMPLE.COM --keytab /etc/krb5.keytab
```

特定のサービスの keytab を削除するには、**--principal (-p)** オプションを使用して、サービスプリンシパルを指定します。

```
# ipa-rmkeytab --principal ldap/client.example.com --keytab  
/etc/krb5.keytab
```

28.5. その他のリソース

- Identity Management サーバーがホストする Kerberos KDC の概要は、[「IdM サーバーでホストするサービス」](#)を参照してください。
- Kerberos の Red Hat ドキュメントについては、『システムレベルの認証ガイド』の[「Kerberos の使用」](#)を参照してください。
- Kerberos の概念に関する詳細情報は、[「MIT Kerberos documentation」](#)を参照してください。

第29章 sudo の使用

Identity Management には、**sudo** ポリシーを IdM ドメイン全体に予測通りかつ一貫性を持って適用するメカニズムがあります。IdM ドメイン内のシステムはすべて、**sudo** クライアントとして設定することが可能です。

29.1. IDENTITY MANAGEMENT の sudo ユーティリティー

sudo ユーティリティーを使うと、指定されたユーザーに管理アクセスが与えられます。信頼できるユーザーが **sudo** を管理コマンドの前に付けると、**ユーザー自身**のパスワードが要求されます。ユーザーが認証され、コマンドが許可されると、管理コマンドは root ユーザーのように実行されます。**sudo** についての詳細情報は、『[システム管理者のガイド](#)』を参照してください。

29.1.1. sudo 向け Identity Management LDAP スキーマ

IdM には、**sudo** エントリ用の特別な LDAP スキーマがあります。これは以下をサポートしています。

- ホストグループおよび netgroup。 **sudo** がサポートしているのは netgroup のみであることに注意してください。
- **sudo** コマンドグループ。これには複数のコマンドが含まれます。



注記

sudo はホストグループやコマンドグループをサポートしないので、**sudo** ルールの作成時に IdM は IdM **sudo** 設定をネイティブの **sudo** 設定に変換します。たとえば、IdM は各ホストグループに対応するシャドウ netgroup を作成し、これにより IdM 管理者はホストグループを参照する **sudo** ルールを作成することができます。一方で、ローカルの **sudo** コマンドは対応する netgroup を使用します。

デフォルトでは **sudo** 情報は、LDAP で匿名で利用可能ではないので、IdM はデフォルトの **sudo** ユーザーを **uid=sudo,cn=sysaccounts,cn=etc,\$SUFFIX** で定義します。このユーザーは、**/etc/sudo-ldap.conf** にある LDAP **sudo** 設定ファイルで変更することが可能です。

29.1.2. NIS ドメイン名要件

netgroup と **sudo** が正常に機能するには、NIS ドメイン名を設定する必要があります。**sudo** の設定には、NIS 形式 netgroup と netgroup 用の NIS ドメイン名が必要になります。ただし、この NIS ドメイン自体が存在する必要はありません。また、NIS サーバーがインストールされている必要もありません。



注記

ipa-client-install ユーティリティーは、デフォルトで NIS ドメイン名を自動的に IdM ドメイン名に設定します。

29.2. IDENTITY MANAGEMENT での sudo ルール

sudo ルールを使用することで、**誰が何をどこで、および誰として**という要素を定義できます。

- **誰** は、**sudo** を使用できるユーザーです。

- 何 は、**sudo** で使用できるコマンドです。
- どこで は、ユーザーが **sudo** を使用できるターゲットホストです。
- 誰として は、ユーザーがタスクを実行する上で装うシステムまたはユーザー ID です。

29.2.1. sudo ルールにおける外部ユーザーとホスト

IdM は、**sudo** ルールで外部のエントリティーを受け付けます。外部のエントリティーとは、IdM ドメインの一部ではないユーザーやホストなど、IdM ドメイン外で保存されているエントリティーです。

たとえば、**sudo** ルールを使って IdM 内の IT グループのメンバーに root アクセスを付与することができます。この場合の root ユーザーは、IdM ドメインで定義されているユーザーではありません。別の例では、ネットワーク上にあるものの IdM ドメインの一部ではない特定ホストへのアクセスを管理者はブロックすることができます。

29.2.2. sudo ルールでのユーザーグループのサポート

sudo を使って、IdM のユーザーグループ全体にアクセスを付与することができます。IdM では、Unix および 非 POSIX グループの両方に対応しています。非 POSIX グループを作成すると、非 POSIX グループ内のユーザーはこのグループから非 POSIX パーミッションを継承するため、アクセス問題が発生する場合がありますことに注意してください。

29.2.3. sudoers オプションのサポート

IdM は **sudoers** オプションをサポートしています。利用可能な **sudoers** オプションの全一覧については、`sudoers(5)` man ページを参照してください。

IdM は **sudoers** オプションで空白や改行を受け付けられないことに注意してください。このため、複数のオプションはコンマ区切りリストではなく、個別のコマンドで追加してください。たとえば、**sudoers** オプション 2 つをコマンドラインから追加するには、以下のようにします。

```
$ ipa sudorule-add-option sudo_rule_name
Sudo Option: first_option
$ ipa sudorule-add-option sudo_rule_name
Sudo Option: second_option
```

同様に、長いオプションでも以下のように一行で提供します。

```
$ ipa sudorule-add-option sudo_rule_name
Sudo Option: env_keep="COLORS DISPLAY EDITOR HOSTNAME HISTSIZE INPUTRC
KDEDIR LESSSECURE LS_COLORS MAIL PATH PS1 PS2 XAUTHORITY"
```

29.3. sudo ポリシーをルックアップする場所の設定

sudo 設定用の中央 IdM データベースは、IdM で定義された **sudo** ポリシーをグローバルで全ドメインホストに利用可能とします。Red Hat Enterprise Linux 7.1 以降のシステムでは、SSSD を **sudo** 用のデータプロバイダーとして設定することで、システムが IdM 定義のポリシーを使用するよう、**ipa-server-install** と **ipa-client-install** のユーティリティーが自動的に設定します。

sudo ポリシーをルックアップする場所は、`/etc/nsswitch.conf` ファイルの **sudoers** の行で定義します。Red Hat Enterprise Linux 7.1 以降で実行している IdM では、`nsswitch.conf` での **sudoers** のデフォルト設定は以下のようになります。

-

```
sudoers: files sss
```

files オプションは、**/etc/sudoers** のローカル SSSD 設定ファイルで定義された **sudo** 設定をシステムが使用することを指定します。**sss** オプションは、IdM で定義された **sudo** 設定を使用することを指定します。

29.3.1. ホストが IdM の以前のバージョンの IdM sudo ポリシーを使用する設定

IdM 定義の **sudo** ポリシーを Red Hat Enterprise Linux 7.1 より前のバージョンで稼働する IdM システムに実装するには、ローカルマシンを手動で設定する必要があります。これは SSSD または LDAP を使用することで可能になります。Red Hat では、SSSD ベースの設定を使用することを強く推奨しています。

29.3.1.1. SSSD を使用した sudo ポリシーのホストへの適用

sudo ルールに SSSD を使用する必要があるシステムで、以下のステップを実行します。

1. **sudoers** ファイルで SSSD をルックアップするよう **sudo** を設定します。

```
# vim /etc/nsswitch.conf

sudoers: files sss
```

files オプションを残しておくと、**sudo** は IdM 設定を SSSD でチェックする前にローカル設定をチェックします。

2. **sudo** を、ローカルの SSSD クライアントが管理するサービス一覧に追加します。

```
# vim /etc/sss/sssd.conf

[sssd]
config_file_version = 2
services = nss, pam, sudo
domains = IPADOMAIN
```

3. **sudo** 設定で NIS ドメインの名前を設定します。**sudo** は NIS スタイルの netgroup を使用するので、**sudo** が IdM **sudo** 設定で使用されているホストグループを発見できるようにするには、NIS ドメイン名はシステム設定で設定する必要があります。

1. **rhel-domainname** サービスが有効になっていない場合はこれを有効にし、NIS ドメイン名が再起動後も維持されるようにします。

```
# systemctl enable rhel-domainname.service
```

2. **sudo** ルールで使用する NIS ドメイン名を設定します。

```
# nisdomainname example.com
```

3. NIS ドメイン名が維持されるようにシステム認証設定を設定します。例を示します。

```
# echo "NISDOMAIN=example.com" >> /etc/sysconfig/network
```

これで NIS ドメインを使って `/etc/sysconfig/network` および `/etc/yp.conf` のファイルが更新されます。

4. オプションで、SSSD 内のデバッグを有効にして、使用している LDAP 設定を表示することができます。

```
[domain/IPADOMAIN]
debug_level = 6
....
```

SSSD が使用する LDAP 検索ベースは、`sssd_DOMAINNAME.log` のログに記録されます。

29.3.1.2. LDAP を使用して **sudo** ポリシーをホストに適用する



重要

SSSD を使用しないクライアントでは、LDAP ベースの設定のみを使用してください。Red Hat では、他のクライアントに関しては SSSD ベースの設定を使用することを推奨しています。これについては、[「SSSD を使用した **sudo** ポリシーのホストへの適用」](#)を参照してください。

LDAP を使用した **sudo** ポリシー適用に関する情報は、[Identity Management Guide for Red Hat Enterprise Linux 6](#) を参照してください。

LDAP ベースの設定は、主に Red Hat Enterprise Linux 7 よりも前のバージョンの Red Hat Enterprise Linux 上のクライアントで使うことが想定されています。このため、これについては Red Hat Enterprise Linux 6 のドキュメントで説明されています。

29.4. sudo コマンド、コマンドグループ、およびルールの追加

29.4.1. sudo コマンドの追加

Web UI での **sudo** コマンドの追加

1. **Policy** タブで **Sudo** → **Sudo Commands** をクリックします。
2. リスト上部にある **Add** をクリックします。
3. コマンドについての情報を入力します。コマンド実行可能ファイルへの完全なシステムパスを入力します。

図29.1 新規 sudo コマンドの追加

4. **Add** をクリックします。別の方法では、**Add and Add Another** をクリックして、別のユーザーを追加するか、**Add and Edit** をクリックして新規エントリーの編集を開始します。

コマンドラインからの sudo コマンドの追加

sudo コマンドを追加するには、**ipa sudocmd-add** コマンドを使用します。コマンドの実行可能ファイルへの完全なシステムパスを提供します。たとえば、**/usr/bin/less** コマンドとその説明を追加するには、以下を実行します。

```
$ ipa sudocmd-add /usr/bin/less --desc="For reading log files"
-----
Added sudo command "/usr/bin/less"
-----
sudo Command: /usr/bin/less
Description: For reading log files
```

29.4.2. sudo コマンドグループの追加

Web UI での sudo コマンドグループの追加

1. **Policy** タブで **Sudo** → **Sudo Command Groups** をクリックします。
2. リスト上部にある **Add** をクリックします。
3. コマンドグループについての情報を入力します。

Add Sudo Command Group

Sudo Command *

Group

Description

* Required field

Add Add and Add Another Add and Edit Cancel

図29.2 新規 sudo コマンドグループの追加

4. **Add and Edit** をクリックしてコマンドグループを編集します。
5. **Sudo Commands** タブで **Add** をクリックして **sudo** コマンドをグループに追加します。該当するコマンドを選択し、> ボタンを使って **Prospective** コラムに移動させます。

Add Sudo Commands into Sudo Command Group files

Filter available Sudo Commands Filter

Available

<input type="checkbox"/>	Sudo Command	Description
<input checked="" type="checkbox"/>	/usr/bin/less	For reading log files.
<input checked="" type="checkbox"/>	/usr/bin/vim	For editing files.

> <

Prospective

<input type="checkbox"/>	Sudo Command	Description
--------------------------	--------------	-------------

Add Cancel

図29.3 sudo コマンドグループへのコマンドの追加

6. **Add** をクリックします。

コマンドラインからの sudo コマンドグループの追加

1. **ipa sudocmdgroup-add** コマンドを使ってコマンドグループを作成します。たとえば、**files** コマンドを作成してその説明を追加するには、以下を実行します。

```
$ ipa sudocmdgroup-add files --desc="File editing commands"
-----
Added sudo command group "files"
```

```
-----
sudo Command Group: files
Description: File editing commands
```

2. **ipa sudocmdgroup-add-member** コマンドを使用してグループに **sudo** コマンドを含めます。「[sudo コマンドの追加](#)」で説明されたように IdM に追加されたコマンドしか含められないことに注意してください。

```
$ ipa sudocmdgroup-add-member files --sudocmds "/usr/bin/vim"
sudo Command Group: files
Description: File editing commands
Member sudo commands: /usr/bin/vim
-----
Number of members added 1
-----
```

29.4.3. sudo ルールの追加

Web UI での sudo ルールの追加

1. **Policy** タブで **Sudo** → **Sudo Rules** をクリックします。
2. リスト上部にある **Add** をクリックします。
3. ルールの名前を入力します。

図29.4 新規 sudo ルールの名前入力

4. **Add** をクリックします。別の方法では、**Add and Add Another** をクリックして、別のユーザーを追加するか、**Add and Edit** をクリックして新規エントリーの編集を開始します。

新規 **sudo** ルールの編集方法については、「[sudo ルールの修正](#)」を参照してください。

コマンドラインからの sudo ルールの追加

新規の **sudo** ルールを追加するには、**ipa sudorule-add** コマンドを使用します。たとえば、**files-commands** という名前のルールを追加するには、以下を実行します。

```
$ ipa sudorule-add files-commands
-----
Added Sudo Rule "files-commands"
-----
Rule name: files-commands
Enabled: TRUE
```


■
ipa sudorule-add コマンドの使用法とこのコマンドが受け付けるオプションについての詳細は、コマンドに **--help** オプションを実行すると表示されます。

新規 **sudo** ルールの編集方法については、「[sudo ルールの修正](#)」を参照してください。

新規 **sudo** ルールの追加およびコマンドラインからこれを編集する完全な例については、[例29.1「コマンドラインからの新規 sudo ルール追加および修正」](#)を参照してください。

29.5. sudo コマンドとコマンドグループの編集

Web UI での sudo コマンドとコマンドグループの修正

1. **Policy** タブで **Sudo** → **Sudo Commands** または **Sudo** → **Sudo Command Groups** をクリックします。
2. 設定ページを表示するコマンドまたはコマンドグループの名前をクリックします。
3. 必要に応じて設定を変更します。ページによっては上部に **Save** ボタンがあるものもあります。その場合は、このボタンをクリックして変更を保存します。

コマンドラインからの sudo コマンドとコマンドグループの修正

コマンドもしくはコマンドグループを修正するには、それぞれ以下のコマンドを実行します。

- **ipa sudocmd-mod**
- **ipa sudocmdgroup-mod**

上記コマンドにコマンドラインオプションを追加して、**sudo** コマンドまたはコマンドグループの属性を更新します。たとえば、**/usr/bin/less** コマンドに新たな説明を追加するには、以下を実行します。

```
$ ipa sudocmd-mod /usr/bin/less --desc="For reading log files"
-----
Modified Sudo Command "/usr/bin/less"
-----
Sudo Command: /usr/bin/less
Description: For reading log files
Sudo Command Groups: files
```

これらのコマンドや対応のオプションに関する情報は、コマンドに **--help** オプションを追加して実行してください。

29.6. sudo ルールの修正

Web UI での sudo ルールの修正

1. **Policy** タブで **Sudo** → **Sudo Rules** をクリックします。
2. 設定ページを表示するルールの名前をクリックします。
3. 必要に応じて設定を変更します。ページによっては上部に **Save** ボタンがあるものもあります。その場合は、このボタンをクリックして変更を保存します。

sudo ルールの設定ページには、以下の設定エリアがあります。

General エリア

このエリアでは、ルールの説明と **sudo order** を修正します。**sudo order** フィールドには、IdM がルールを評価する順番を整数で入力します。**sudo order** の値の最も高いルールが最初に評価されます。

Options エリア

このエリアでは、**sudoers** オプションをルールに追加します。

1. オプション一覧の上部にある **Add** をクリックします。

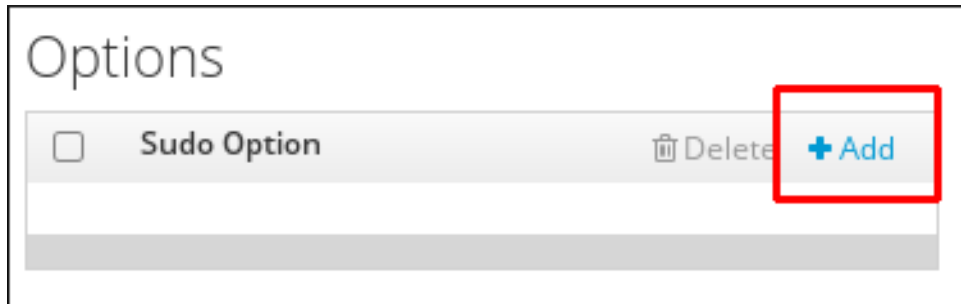


図29.5 sudo オプションの追加

2. **sudoers** オプションを入力します。たとえば、**sudo** でユーザー認証が要求されないようにするには、**!authenticate** を追加します。



図29.6 sudoers オプションの追加

sudoers オプションの詳細情報については、`sudoers(5) man` ページを参照してください。

3. **Add** をクリックします。

Who エリア

このエリアでは、**sudo** ルールが適用されるユーザーまたはユーザーグループを選択します。これらのユーザーは、ルールで定義されたように **sudo** を使用することができます。

すべてのシステムユーザーがルールで定義したように **sudo** を使用出来るようにするには、**Anyone** を選択します。

ルールを特定のユーザーまたはグループのみに適用するには、**Specified Users and Groups** を選択して、以下のステップに従います。

1. ユーザーまたはユーザーグループ一覧上部にある **Add** をクリックします。

Who

User category the rule applies to: ☐ Anyone ☒ Specified Users and Groups

<input type="checkbox"/>	Users	External	Delete	Add
<input type="checkbox"/>	manager			
<input type="checkbox"/>	employee			
<input type="checkbox"/>	helpdesk			

<input type="checkbox"/>	User Groups		Delete	Add
<input type="checkbox"/>	admins			

図29.7 sudo ルールへのユーザーの追加

2. ルールに追加するユーザーまたはユーザーグループを選択し、> ボタンをクリックして **Prospective** コラムに移動します。外部ユーザーを追加する場合には **External** フィールドでユーザーを指定してから > ボタンをクリックします。

Add Users into Sudo Rule files-commands

Filter available Users

Filter

Available

<input type="checkbox"/>	Users
<input type="checkbox"/>	xyz

External

>

<

Prospective

<input type="checkbox"/>	Users
<input checked="" type="checkbox"/>	employee
<input checked="" type="checkbox"/>	helpdesk
<input checked="" type="checkbox"/>	manager

Add

Cancel

図29.8 sudo ルール向けにユーザーを選択する

3. **Add** をクリックします。

Access This Host エリア

このエリアでは、**sudo** ルールを有効にするホストを選択します。これは、ユーザーに **sudo** パーミッションが付与されるホストになります。

全ホストでルールを有効にするには、**Anyone** を選択します。

ルールを特定のホストまたはホストグループのみに適用するには、**Specified Hosts and Groups** を選択して、以下のステップに従います。

1. ホスト一覧の上部にある **Add** をクリックします。

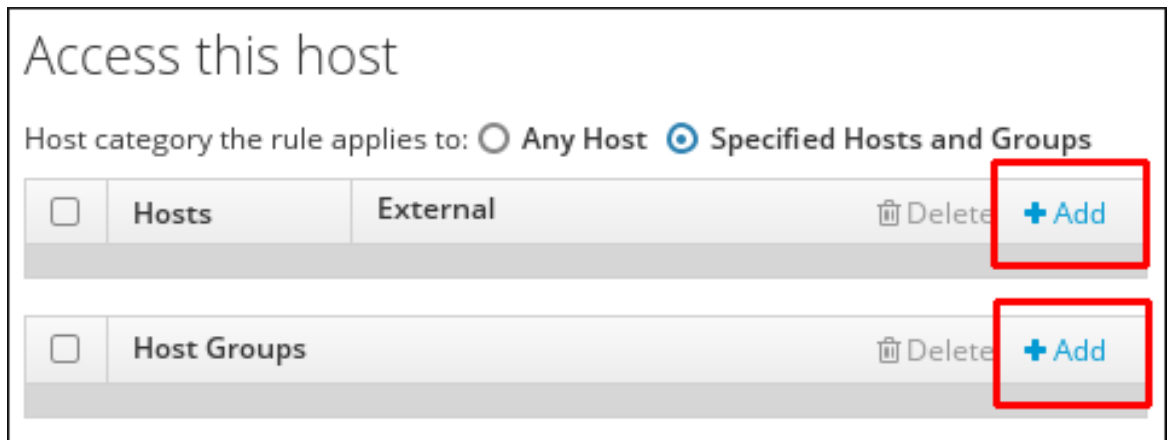


図29.9 sudo ルールへのホストの追加

2. ルールに含めるホストまたはホストグループを選択し、> ボタンをクリックして **Prospective** コラムに移動します。外部ホストを追加する場合には **External** フィールドでユーザーを指定してから > ボタンをクリックします。

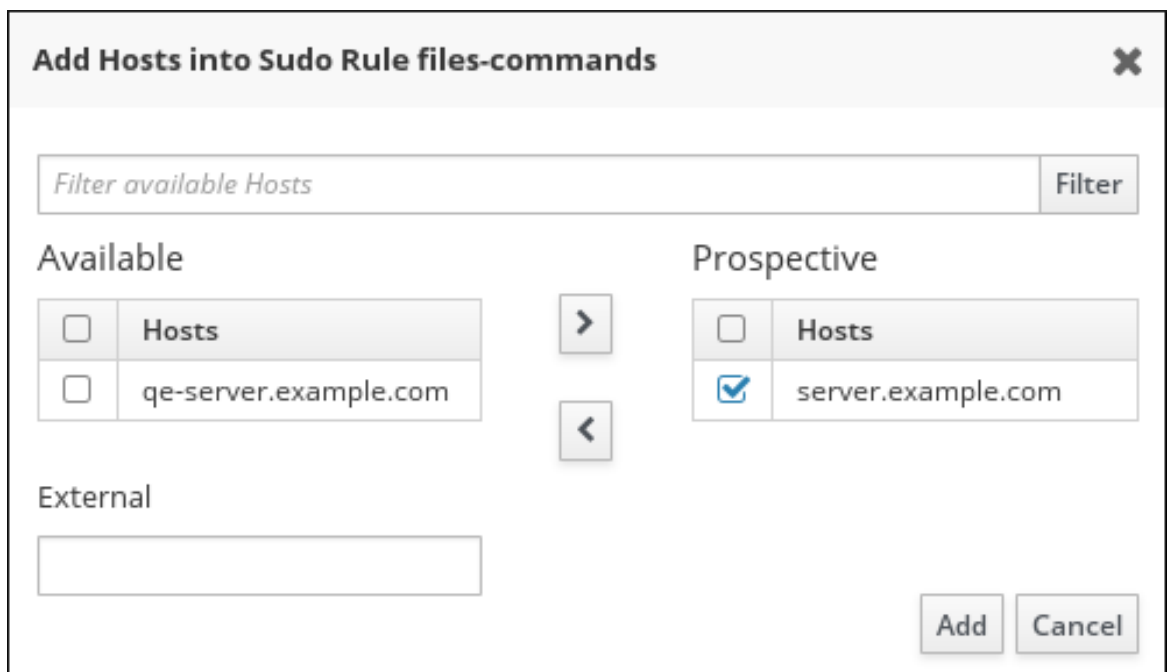


図29.10 sudo ルール用のホスト選択

3. **Add** をクリックします。

Run Commands エリア

このエリアでは、**sudo** ルールに含めるコマンドを選択します。特定コマンドの使用をユーザーに許可する、もしくは拒否することを指定できます。

ユーザーが **sudo** ですべてのコマンドを使用できるようにするには、**Any Command** を選択します。

ルールを特定のコマンドまたはコマンドグループに関連付けるには、**Specified Commands and Groups** を選択して、以下のステップに従います。

1. いずれかの **Add** ボタンをクリックして、コマンドまたはコマンドグループを追加します。

許可するコマンドまたはコマンドグループを指定するには、**Allow** エリアを使用します。
拒否するコマンドまたはコマンドグループを指定するには **Deny** エリアを使用します。

Run Commands

Command category the rule applies to: ☐ Any Command ☒ Specified Commands and Groups

Allow

<input type="checkbox"/>	Sudo Allow Commands	Delete	+ Add
<input type="checkbox"/>	Sudo Allow Command Groups	Delete	+ Add

Deny

<input type="checkbox"/>	Sudo Deny Commands	Delete	+ Add
<input type="checkbox"/>	Sudo Deny Command Groups	Delete	+ Add

図29.11 sudo ルールへのコマンドの追加

2. ルールに含めるコマンドまたはコマンドグループを選択し、> ボタンをクリックして **Prospective** コラムに移動させます。

Add Allow Sudo Commands into Sudo Rule files-commands

Filter available Sudo Commands Filter

Available

<input type="checkbox"/>	Sudo Commands
<input type="checkbox"/>	editing
<input type="checkbox"/>	log-files
<input type="checkbox"/>	login

> <

Prospective

<input type="checkbox"/>	Sudo Commands
<input checked="" type="checkbox"/>	files

Add Cancel

図29.12 sudo ルール向けにコマンドを選択する

3. **Add** をクリックします。

As Whom エリア

このエリアでは、特定の root 以外のユーザーとして特定コマンドを実行するよう **sudo** を設定します。

RunAs users のグループを追加すると、そのグループのメンバーの UID がそのコマンドの実行に使用されることに注意してください。また、RunAs group を追加すると、コマンドの実行にそのグループの GID が使用されます。

システム上のいずれのユーザーとしてルールを実行できるようにするには、**Anyone** を選択します。システム上のいずれのグループとしてルールを実行できるようにするには、**Any Group** を選択します。

1. ユーザー一覧上部にある **Add** をクリックします。

As Whom

RunAs User category the rule applies to: ☐ Anyone ☒ Specified Users and Groups

RunAs Users	External	Delete	+ Add
<input type="checkbox"/>			

RunAs Group category the rule applies to: ☐ Any Group ☒ Specified Groups

RunAs Groups	External	Delete	+ Add
<input type="checkbox"/>			

図29.13 特定ユーザーとしてコマンドを実行する **sudo** ルールの設定

2. ユーザーまたはグループを選択し、> ボタンをクリックして **Prospective** コラムに移動させます。外部のエンティティを追加する場合には **External** フィールドで指定してから > ボタンをクリックします。

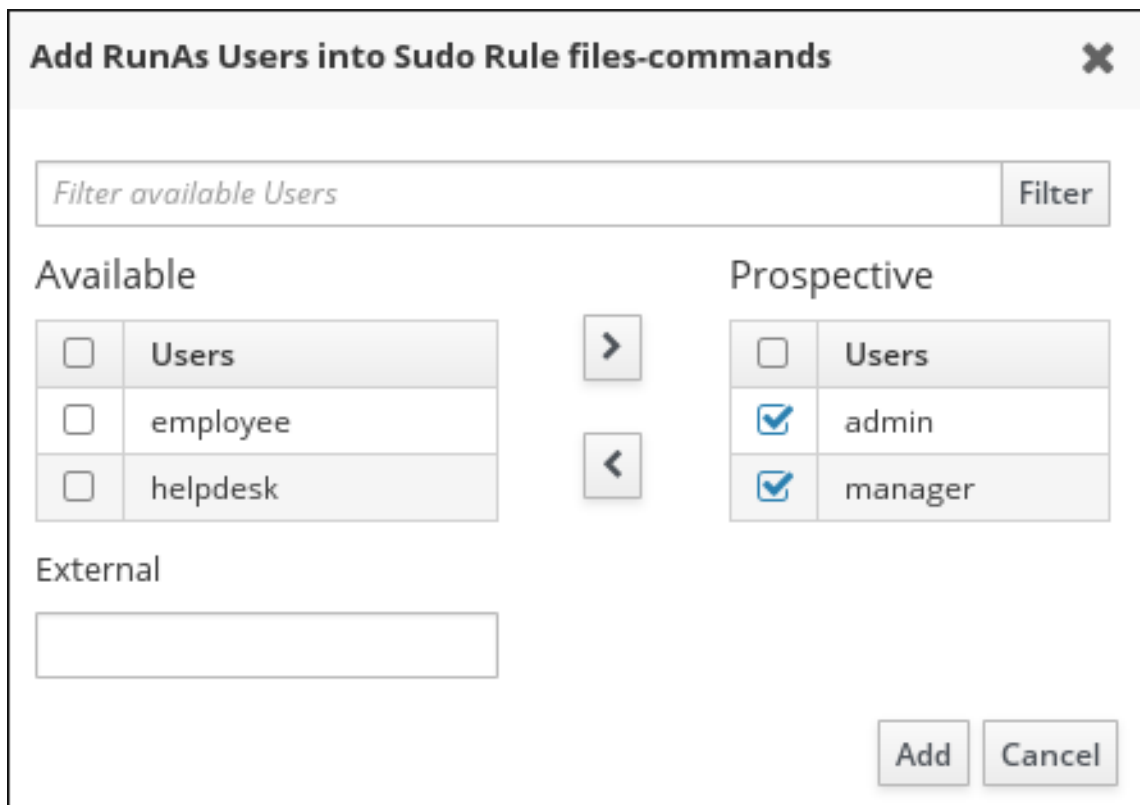


図29.14 コマンドのユーザー選択

3. **Add** をクリックします。

コマンドラインからの **sudo** ルールの修正

IdM コマンドラインでは、以下の **sudo** ルールエリアを設定できます。

全般的 **sudo** ルールの管理

sudo ルールの全般的設定を変更するには、**ipa sudorule-mod** コマンドを使用します。よく使用されるオプションには、以下のものがあります。

- **--desc** オプションでは、以下のように **sudo** ルールの説明を変更します。

```
$ ipa sudorule-mod sudo_rule_name --desc="sudo_rule_description"
```

- **--order** オプションでは、以下のように指定されたルールの順序を定義します。

```
$ ipa sudorule-mod sudo_rule_name --order=3
```

- エンティティーのカテゴリは以下のオプションで指定します: **--usercat** (ユーザーカテゴリ)、**--hostcat** (ホストカテゴリ)、**--cmdcat** (コマンドカテゴリ)、**--runasusercat** (run-as ユーザーカテゴリ)、および **--runasgroupcat** (run-as グループカテゴリ)。これらのオプションではすべて **all** の値を使用することが可能で、その場合は該当ルールをすべてのユーザー、ホスト、コマンド、run-as ユーザー、もしくは run-as グループに関連付けます。

たとえば、**sudo_rule** ルールで定義した **sudo** を全ユーザーが使用できるように指定するには、以下を実行します。

```
$ ipa sudorule-mod sudo_rule --usercat=all
```

ルールが既に特定エンティティと関連付けられている場合は、この関連付けを解除してからそのエンティティに対応する **all** カテゴリを定義する必要があります。たとえば、**sudo_rule** が **ipa sudorule-add-user** コマンドを使用して特定のユーザーと関連付けられている場合は、まず **ipa sudorule-remove-user** コマンドを使ってこのユーザーを削除する必要があります。

ipa sudorule-mod で使用可能なオプションの完全一覧と詳細情報については、コマンドに **--help** オプションを追加して実行してください。

sudo オプションの管理

sudoers オプションを追加するには、**ipa sudorule-add-option** コマンドを使用します。

たとえば、**files-commands** ルールをベースとした **sudo** を使用するユーザーの認証を不要とするには、**!authenticate** オプションを使用します。

```
$ ipa sudorule-add-option files-commands
Sudo Option: !authenticate
-----
Added option "!authenticate" to Sudo Rule "files-commands"
-----
```

sudoers オプションの詳細情報については、**sudoers(5) man** ページを参照してください。

sudoers オプションを削除するには、以下のように **ipa sudorule-remove-option** コマンドを使用します。

```
$ ipa sudorule-remove-option files-commands
Sudo Option: authenticate
-----
Removed option "authenticate" from Sudo Rule "files-commands"
-----
```

sudo 使用のパーミッション付与の管理

個別ユーザーを指定するには、**--users** オプションを **ipa sudorule-add-user** コマンドで使います。ユーザーグループを指定するには、**--groups** オプションを **ipa sudorule-add-user** に追加します。

たとえば、**user** と **user_group** を **files-commands** ルールに追加するには、以下のコマンドを実行します。

```
$ ipa sudorule-add-user files-commands --users=user --groups=user_group
...
-----
Number of members added 2
-----
```

個別のユーザーもしくはグループを削除するには、以下のように **ipa sudorule-remove-user** を使用します。

```
$ ipa sudorule-remove-user files-commands
[member user]: user
[member group]:
...
```



```
-----
Number of members removed 1
-----
```

ユーザーに **sudo** パーミッションを付与する場所の管理

ホストを指定するには、**--hosts** オプションを **ipa sudorule-add-host** コマンドで使用します。ホストグループを指定するには、**--hostgroups** オプションを **ipa sudorule-add-host** に追加します。

たとえば、**example.com** と **host_group** を **files-commands** ルールに追加するには、以下のコマンドを実行します。

```
$ ipa sudorule-add-host files-commands --hosts=example.com --
hostgroups=host_group
...
-----
Number of members added 2
-----
```

ホストまたはホストグループを削除するには、以下のように **ipa sudorule-remove-host** コマンドを使用します。

```
$ ipa sudorule-remove-host files-commands
[member host]: example.com
[member host group]:
...
-----
Number of members removed 1
-----
```

sudo と使用するコマンドの管理

特定コマンドの使用をユーザーに許可する、もしくは拒否することを指定できます。

許可するコマンドもしくはコマンドグループを指定するには、**--sudocmds** または **--sudocmdgroups** オプションを **ipa sudorule-add-allow-command** に追加します。拒否するコマンドもしくはコマンドグループを指定するには、**--sudocmds** または **--sudocmdgroups** オプションを **ipa sudorule-add-deny-command** コマンドに追加します。

たとえば、**/usr/bin/less** コマンドと **files** コマンドグループを許可するものとして **files-commands** ルールに追加するには、以下のコマンドを実行します。

```
$ ipa sudorule-add-allow-command files-commands --sudocmds=/usr/bin/less
--sudocmdgroups=files
...
-----
Number of members added 2
-----
```

コマンドまたはコマンドグループをルールから削除するには、以下のように **ipa sudorule-remove-allow-command** または **ipa sudorule-remove-deny-command** コマンドを実行します。

```
$ ipa sudorule-remove-allow-command files-commands
```

```
[member sudo command]: /usr/bin/less
[member sudo command group]:
...
-----
Number of members removed 1
-----
```

--sudocmds オプションは「[sudo コマンドの追加](#)」にあるように、IdM に追加されたコマンドしか受け付けないことに注意してください。

sudo コマンドの実行者の ID 管理

個別ユーザーもしくはグループ内のユーザーの UID をコマンド実行時の ID として使用するには、**-users** または **--groups** オプションを **ipa sudorule-add-runasuser** コマンドで使います。

ユーザーグループの GID をコマンド実行時の ID として使用するには、**ipa sudorule-add-runasgroup --groups** コマンドを使用します。

ユーザーやグループを指定しない場合は、**sudo** コマンドは root として実行されます。

たとえば、**user** の ID を使用して **sudo** ルール内のコマンドを実行するように指定するには、以下を実行します。

```
$ ipa sudorule-add-runasuser files-commands --users=user
...
RunAs Users: user
...
```

ipa sudorule-* コマンドの詳細については、**ipa help sudorule** コマンドの出力を確認するか、各コマンドに **--help** オプションを追加して実行します。

例29.1 コマンドラインからの新規 sudo ルール追加および修正

選択したサーバーで特定のユーザーグループが**sudo** ですべてのコマンドを使用できるようにするには、以下の手順を実行します。

1. **admin** ユーザーまたは **sudo** ルールの管理を許可されている他のユーザー用に Kerberos チケットを取得します。

```
$ kinit admin
Password for admin@EXAMPLE.COM:
```

2. 新規 **sudo** ルールを IdM に追加します。

```
$ ipa sudorule-add new_sudo_rule --desc="Rule for user_group"
-----
Added Sudo Rule "new_sudo_rule"
-----
Rule name: new_sudo_rule
Description: Rule for user_group
Enabled: TRUE
```

3. **who** を定義します。**sudo** ルールの使用が許可されるユーザーのグループを指定します。

```
$ ipa sudorule-add-user new_sudo_rule --groups=user_group
Rule name: new_sudo_rule
Description: Rule for user_group
Enabled: TRUE
User Groups: user_group
-----
Number of members added 1
-----
```

4. **where** を定義します。ユーザーに **sudo** パーミッションが付与されるホストのグループを指定します。

```
$ ipa sudorule-add-host new_sudo_rule --hostgroups=host_group
Rule name: new_sudo_rule
Description: Rule for user_group
Enabled: TRUE
User Groups: user_group
Host Groups: host_group
-----
Number of members added 1
-----
```

5. **what** を定義します。どの **sudo** コマンドもユーザーが実行することを許可するには、**all** コマンドカテゴリーをルールに追加します。

```
$ ipa sudorule-mod new_sudo_rule --cmdcat=all
-----
Modified Sudo Rule "new_sudo_rule"
-----
Rule name: new_sudo_rule
Description: Rule for user_group
Enabled: TRUE
Command category: all
User Groups: user_group
Host Groups: host_group
```

6. **sudo** コマンドを root として実行するには、run-as ユーザーまたはグループを指定しないでください。
7. **sudo** コマンド使用時にユーザー認証が要求されないようにするには、**!authenticate sudoers** を追加します。

```
$ ipa sudorule-add-option new_sudo_rule
Sudo Option: !authenticate
-----
Added option "!authenticate" to Sudo Rule "new_sudo_rule"
-----
Rule name: new_sudo_rule
Description: Rule for user_group
Enabled: TRUE
Command category: all
User Groups: user_group
Host Groups: host_group
Sudo Option: !authenticate
```

8. 新規の **sudo** ルール設定を表示して、内容を確認します。

```
$ ipa sudorule-show new_sudo_rule
Rule name: new_sudo_rule
Description: Rule for user_group
Enabled: TRUE
Command category: all
User Groups: user_group
Host Groups: host_group
Sudo Option: !authenticate
```

29.7. sudo コマンド、コマンドグループ、およびルールの表示

Web UI での sudo コマンド、コマンドグループおよびルールの表示

1. **Policy** タブで **Sudo** をクリックし、**Sudo Rules**、**Sudo Commands**、または **Sudo Command Groups** のいずれかを選択します。
2. 設定ページを表示するコマンド、コマンドグループまたはルールの名前をクリックします。

コマンドラインからの sudo コマンド、コマンドグループおよびルールの表示

全コマンド、全コマンドグループ、または全ルールを一覧表示するには、それぞれ以下のコマンドを使用します。

- **ipa sudocmd-find**
- **ipa sudocmdgroup-find**
- **ipa sudorule-find**

特定のコマンド、コマンドグループ、またはルールの情報を表示するには、それぞれ以下のコマンドを使用します。

- **ipa sudocmd-show**
- **ipa sudocmdgroup-show**
- **ipa sudorule-show**

たとえば、**/usr/bin/less** コマンドの情報を表示するには、以下を実行します。

```
$ ipa sudocmd-show /usr/bin/less
Sudo Command: /usr/bin/less
Description: For reading log files.
Sudo Command Groups: files
```

これらのコマンドや対応のオプションに関する情報は、コマンドに **--help** オプションを追加して実行してください。

29.8. sudo ルールの有効化および無効化

sudo ルールを無効にすると、これが一時的に非アクティブになります。無効になったルールは IdM から削除されるわけではなく、再度有効にすることができます。

Web UI での sudo ルールの有効化、無効化

1. **Policy** タブで **Sudo** → **Sudo Rules** をクリックします。
2. ルールを選択して、**Disable** または **Enable** をクリックします。

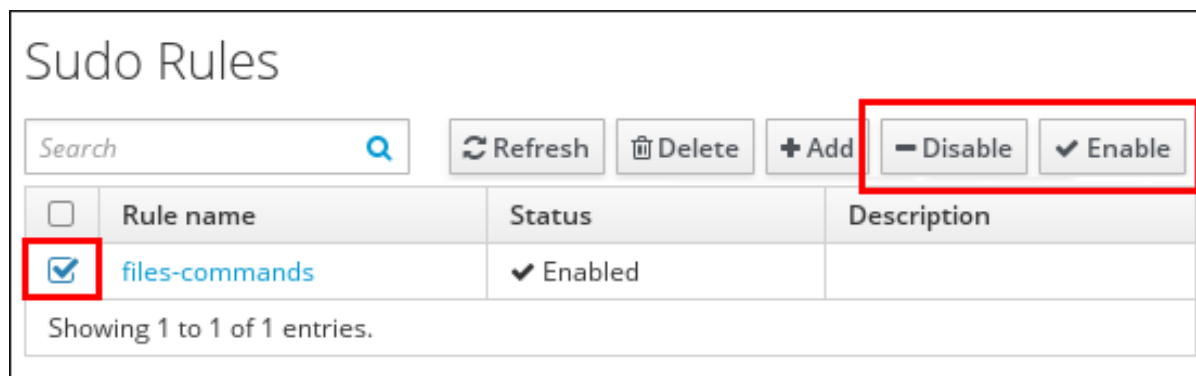


図29.15 sudo ルールの有効化および無効化

コマンドラインからの sudo ルールの有効化、無効化

ルールを無効にするには、**ipa sudo-rule-disable** コマンドを使用します。

```
$ ipa sudorule-disable sudo_rule_name
-----
Disabled Sudo Rule "sudo_rule_name"
-----
```

ルールを再度有効にするには、**ipa sudorule-enable** コマンドを使用します。

```
$ ipa sudorule-enable sudo_rule_name
-----
Enabled Sudo Rule "sudo_rule_name"
-----
```

29.9. sudo コマンド、コマンドグループ、およびルールの削除

Web UI での sudo コマンド、コマンドグループおよびルールの削除

1. **Policy** タブで **Sudo** をクリックし、**Sudo Rules**、**Sudo Commands**、または **Sudo Command Groups** のいずれかを選択します。
2. 削除するコマンド、コマンドグループまたはルールを選択して **Delete** をクリックします。

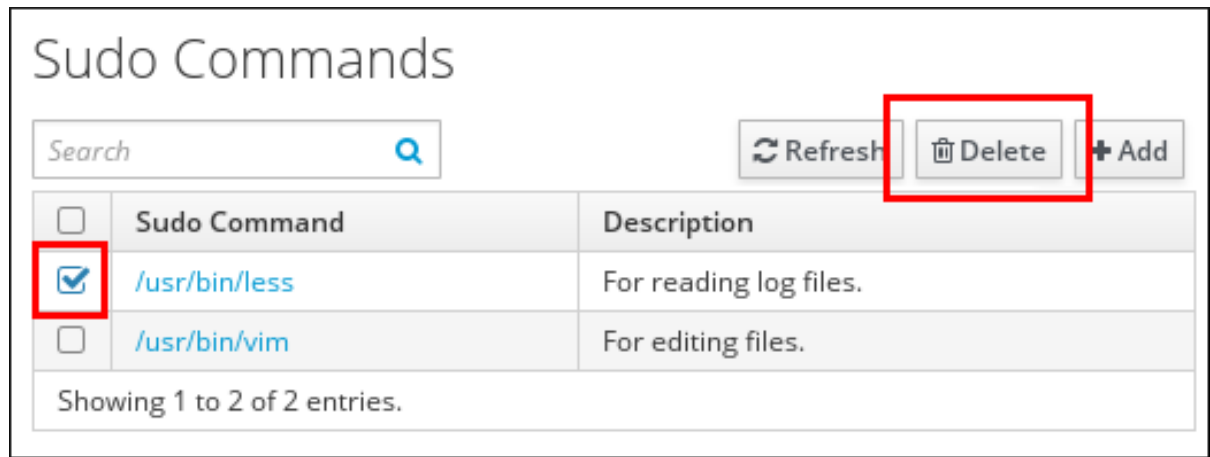


図29.16 sudo コマンドの削除

コマンドラインからの sudo コマンド、コマンドグループおよびルール削除

コマンド、コマンドグループまたはルールを削除するには、それぞれ以下のコマンドを使用します。

- **ipa sudocmd-del**
- **ipa sudocmdgroup-del**
- **ipa sudorule-del**

これらのコマンドや対応のオプションに関する情報は、コマンドに **--help** オプションを追加して実行してください。

第30章 ホストベースのアクセス制御の設定

本章では、Identity Management (IdM) での *host-based access control* (HBAC) や、IdM ドメインでのアクセス制御管理に HBAC を使用方法について説明します。

30.1. IDM での HOST-BASED ACCESS CONTROL の機能

Host-based access control は、指定のサービス (またはサービスグループ内のサービス) を使用して、指定のホスト (またはホストグループ) にアクセスできるユーザーを定義します。たとえば、以下が可能です。

- ドメイン内の指定のシステムへのアクセスを、特定のユーザーグループに所属するメンバーに制限すること
- ドメイン内のシステムにアクセスして特定のサービスだけを使用できるようにすること

管理者は、HBAC ルールと呼ばれる許可ルールを使用して、host-based access control を設定します。デフォルトでは、IdM には **allow_all** という名前のデフォルトの HBAC ルールで設定されています。このルールでは、IdM ドメイン全体にアクセスできます。

HBAC ルールのグループへの適用

アクセス制御管理を集約して簡素化するには、個別のユーザー、ホスト、またはサービスではなく、全ユーザー、ホスト、またはサービスグループに HBAC ルールを適用できます。

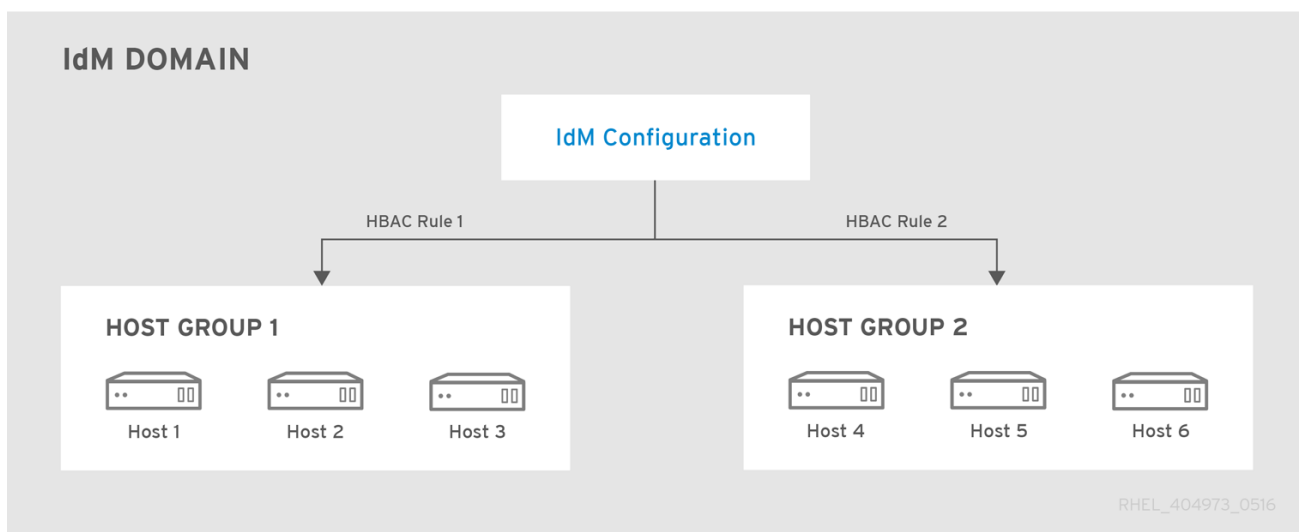


図30.1 ホストグループとホストベースのアクセス制御

グループに HBAC ルールを適用する際には、*automember* ルールの使用を検討してください。「[ユーザーおよびホストの自動グループメンバーシップの定義](#)」を参照してください。

30.2. IDM ドメインでの HOST-BASED ACCESS CONTROL の設定

Host-based access control 用にドメインを設定するには以下を行います。

1. [HBAC ルールの作成](#)
2. [新規 HBAC ルールのテスト](#)
3. [デフォルトの **allow_all** HBAC ルールの無効化](#)



重要

カスタムの HBAC ルールを作成する前に **allow_all** ルールを無効化しないでください。作成してしまうと、ユーザーはどのホストにもアクセスできなくなります。

30.2.1. HBAC ルールの作成

以下を使用して、HBAC ルールを作成できます。

- IdM web UI (「[Web UI: HBAC ルールの作成](#)」を参照してください)
- コマンドライン (「[コマンドライン: HBAC ルールの作成](#)」を参照してください)

サンプルについては「[HBAC ルールの例](#)」を参照してください。

Web UI: HBAC ルールの作成

1. **Policy** → **Host Based Access Control** → **HBAC Rules**を選択します。
2. **Add** をクリックして、新規ルールの追加を開始します。
3. ルールの名前を入力して **Add and Edit** をクリックし、HBAC ルールの設定ページに直接移動します。
4. **Who** エリアで対象ユーザーを指定します。
 - 特定のユーザーまたはグループのみに HBAC ルールを適用するには、**Specified Users and Groups** を選択してから、**Add** をクリックしてユーザーまたはグループを追加します。
 - 全ユーザーに HBAC ルールを適用するには、**Anyone** を選択します。

Who

User category the rule applies to: ☐ Anyone ☒ Specified Users and Groups

Users	Delete	+Add
<input type="checkbox"/> admin		

User Groups	Delete	+Add
<input type="checkbox"/>		

図30.2 HBAC ルールの対象ユーザーの指定

5. **Accessing** エリアで対象ホストを指定します。
 - 特定のホストまたはグループのみに HBAC ルールを適用するには、**Specified hosts and Groups** を選択してから、**Add** をクリックしてホストまたはグループを追加します。
 - 全ホストに HBAC ルールを適用するには、**Any Host** を選択します。
6. **Via Service** エリアでは、対象の HBAC サービスを指定します。

- 。特定のサービスまたはグループのみに HBAC ルールを適用するには、**Specified Services and Groups** を選択してから、**Add** をクリックしてサービスまたはグループを追加します。
- 。全サービスに HBAC ルールを適用するには、**Any Service** を選択します。



注記

デフォルトでは HBAC ルール用に、最も一般的なサービスおよびサービスグループのみが設定されています。

- 。現在利用可能なサービス一覧を表示するには、**Policy → Host-Based Access Control → HBAC Services** を選択します。
- 。現在利用可能なサービスグループ一覧を表示するのは **Policy → Host-Based Access Control → HBAC Service Groups** を選択します。

さらにサービスやサービスグループを追加するには、[「カスタムの HBAC サービス用に HBAC サービスエントリーの追加」](#) および [「HBAC サービスグループの追加」](#) を参照してください。

7. HBAC ルール設定ページで特定の設定を変更すると、ページの上部の **Save** ボタンがハイライトされます。ボタンがハイライトされたら、クリックして変更を確定します。

コマンドライン: HBAC ルールの作成

1. **ipa hbacrule-add** コマンドを使用して、ルールを追加します。

```
$ ipa hbacrule-add
Rule name: rule_name
-----
Added HBAC rule "rule_name"
-----
Rule name: rule_name
Enabled: TRUE
```

2. 対象のユーザーを指定します。

- 。指定のユーザーまたはグループのみに HBAC ルールを適用するには、**ipa hbacrule-add-user** コマンドを使用します。

たとえば、グループを追加するには以下を実行します。

```
$ ipa hbacrule-add-user
Rule name: rule_name
[member user]:
[member group]: group_name
Rule name: rule_name
Enabled: TRUE
User Groups: group_name
-----
Number of members added 1
-----
```

複数のユーザーまたはグループを追加するには、**--users** および **--groups** オプションを使用します。

```
$ ipa hbacrule-add-user rule_name --users=user1 --users=user2 --
users=user3
  Rule name: rule_name
  Enabled: TRUE
  Users: user1, user2, user3
  -----
Number of members added 3
-----
```

- 。HBAC ルールを全ユーザーに適用するには、**ipa hbacrule-mod** コマンドを使用して、**all** ユーザーカテゴリーを指定します。

```
$ ipa hbacrule-mod rule_name --usercat=all
-----
Modified HBAC rule "rule_name"
-----
  Rule name: rule_name
  User category: all
  Enabled: TRUE
```



注記

HBAC ルールが個別ユーザーまたはグループに関連付けられている場合には、**ipa hbacrule-mod --usercat=all** は失敗します。このような場合には、**ipa hbacrule-remove-user** コマンドを使用してユーザーとグループを削除します。

詳細は、**--help** オプションを指定して **ipa hbacrule-remove-user** を実行します。

3. 対象のホストを指定します。

- 。指定のホストまたはグループのみに HBAC ルールを適用するには、**ipa hbacrule-add-host** コマンドを使用します。

たとえば、単一のホストを追加するには、以下を実行します。

```
$ ipa hbacrule-add-host
Rule name: rule_name
[member host]: host.example.com
[member host group]:
  Rule name: rule_name
  Enabled: TRUE
  Hosts: host.example.com
  -----
Number of members added 1
-----
```

複数のホストまたはグループを追加するには、**--hosts** および **--hostgroups** オプションを使用します。

■

```
$ ipa hbacrule-add-host rule_name --hosts=host1 --hosts=host2 --
hosts=host3
Rule name: rule_name
Enabled: TRUE
Hosts: host1, host2, host3
-----
Number of members added 3
-----
```

- 。HBAC ルールを全ホストに適用するには、**ipa hbacrule-mod** コマンドを使用して、**all** ホストカテゴリーを指定します。

```
$ ipa hbacrule-mod rule_name --hostcat=all
-----
Modified HBAC rule "rule_name"
-----
Rule name: rule_name
Host category: all
Enabled: TRUE
```



注記

HBAC ルールが個別ホストまたはグループに関連付けられている場合には、**ipa hbacrule-mod --hostcat=all** は失敗します。このような場合には、**ipa hbacrule-remove-host** コマンドを使用してホストとグループを削除します。

詳細は、**--help** オプションを指定して **ipa hbacrule-remove-host** を実行します。

4. 対象の HBAC サービスを指定します。

- 。指定のサービスまたはグループのみに HBAC ルールを適用するには、**ipa hbacrule-add-service** コマンドを使用します。

たとえば、単一のサービスを追加するには以下を実行します。

```
$ ipa hbacrule-add-service
Rule name: rule_name
[member HBAC service]: ftp
[member HBAC service group]:
Rule name: rule_name
Enabled: TRUE
Services: ftp
-----
Number of members added 1
-----
```

複数のサービスまたはグループを追加するには、**--hbacsvcs** および **--hbacsvcgroups** オプションを使用します。

```
$ ipa hbacrule-add-service rule_name --hbacsvcs=su --
hbacsvcs=sudo
Rule name: rule_name
```

```

Enabled: TRUE
Services: su, sudo
-----
Number of members added 2
-----

```



注記

最も一般的なサービスおよびサービスグループのみが HBAC ルール用に設定されます。さらに追加するには「[カスタムの HBAC サービス用に HBAC サービスエントリーの追加](#)」および「[HBAC サービスグループの追加](#)」を参照してください。

- HBAC ルールを全サービスに適用するには、**ipa hbacrule-mod** コマンドを使用して、**all** サービスカテゴリーを指定します。

```

$ ipa hbacrule-mod rule_name --servicecat=all
-----
Modified HBAC rule "rule_name"
-----
Rule name: rule_name
Service category: all
Enabled: TRUE

```



注記

HBAC ルールが個別サービスまたはグループに関連付けられている場合には、**ipa hbacrule-mod --servicecat=all** は失敗します。このような場合には、**ipa hbacrule-remove-service** コマンドを使用してサービスとグループを削除します。

詳細は、**--help** オプションを指定して **ipa hbacrule-remove-service** を実行します。

5. オプション: HBAC ルールが正しく追加されたことを確認します。

- ipa hbacrule-find** コマンドを使用して、HBAC ルールが IdM に追加されたことを確認します。
- ipa hbacrule-show** コマンドを使用して、HBAC ルールのプロパティを確認します。

詳細は、**--help** オプションを指定してコマンドを実行します。

HBAC ルールの例

例30.1 サービスを使用して単一ユーザーに全ホストへのアクセス権を割り当てる手順

サービスを使用して **admin** ユーザーがドメイン内の全システムにアクセスできるようにするには、新規の HBAC ルールを作成して、以下を設定します。

- ユーザーを **admin** に設定します。
- ホストを **Any host** (web UI で) に設定します。または、**ipa hbacrule-add** (ルールの追加) または **ipa hbacrule-mod** を指定して **--hostcat=all** を実行します。

- サービスを **Any service** (web UI で) に設定します。または、**ipa hbacrule-add** (ルールの追加) または **ipa hbacrule-mod** を指定して **--servicecat=all** を実行します。

例30.2 特定のサービスのみを使用してホストにアクセスする手順

全ユーザーが **sudo** 関連のサービスを使用して、**host.example.com** という名前のホストにアクセスするには、新規の HBAC ルールを作成して、以下を設定します。

- ユーザーを **Anyone** (web UI で) に設定します。または、**ipa hbacrule-add** (ルールの追加) または **ipa hbacrule-mod** を指定して **--usercat=all** を実行します。
- ホストを **host.example.com** に設定します。
- HBAC サービスグループを **Sudo** に設定します。この Sudo は **sudo** と関連サービスのデフォルトグループです。

30.2.2. HBAC ルールのテスト

IdM では、シミュレーションのシナリオを使用して、さまざまな状況で HBAC 設定をテストすることができます。シミュレーションテストの実行を行うことで、HBAC ルールが実稼働環境にデプロイされる前に、設定ミスの問題や、セキュリティリスクを検出できます。



重要

カスタムの HBAC ルールを必ずテストしてから実稼働環境での使用を開始するようにしてください。

IdM は、信頼される Active Directory (AD) ユーザーに対して HBAC ルールが有効であるかどうかのテストは行わない点に注意してください。AD データは IdM LDAP ディレクトリーに保存されないのので、HBAC シナリオをシミュレーションする場合に、IdM は AD ユーザーのグループメンバーシップを解決できません。

以下を使用して、HBAC ルールをテストすることができます。

- IdM web UI (「[Web UI: HBAC ルールのテスト](#)」を参照してください)
- コマンドライン (「[コマンドライン: HBAC ルールのテスト](#)」を参照してください)

Web UI: HBAC ルールのテスト

1. **Policy → Host-Based Access Control → HBAC Test**を選択します。
2. **Who** 画面で、ID のテストを実行するユーザーを指定して、**Next** をクリックします。

Who

Who Accessing Via Service Rules Run Test

WHO

	User login	First name	Last name	Status
<input type="radio"/>	admin		Administrator	✓ Enabled
<input checked="" type="radio"/>	user1	user	user	✓ Enabled
<input type="radio"/>	user2	user	user	✓ Enabled
<input type="radio"/>	user3	user	user	✓ Enabled

Showing 1 to 4 of 4 entries.

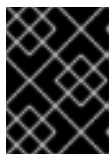
☐ Specify external User:

> Next

図30.3 HBAC テスト用の対象ユーザーの指定

3. **Accessing** 画面でユーザーがアクセスを試みるホストを指定して、**Next** をクリックします。
4. **Via Service** 画面で、ユーザーが使用を試みるサービスを指定して、**Next** をクリックします。
5. **Rules** 画面で、テストする HBAC ルールを選択して **Next** をクリックします。ルールを選択しない場合には、すべてのルールがテストされます。

Include Enabled を選択して、ステータスが **Enabled** の全ルールに対してテストを実行します。**Include Disabled** を選択して、ステータスが **Disabled** の全ルールにテストを実行します。HBAC ルールの表示やステータスの変更は、**Policy → Host Based Access Control → HBAC Rules** を選択します。



重要

複数のルールでテストが実行される場合には、選択したルールの 1 つでアクセスが許可されるとテストに成功します。

6. **Run Test** 画面で、**Run Test** をクリックします。

Run Test

Who Accessing Via Service Rules Run Test

⚙ Run Test

図30.4 HBAC テストの実行

7. テストの結果を確認します。

- **ACCESS DENIED** が表示されると、テストへのアクセス権が拒否されたことになります。
- **ACCESS GRANTED** が表示されると、ホストへのアクセスが正常に許可されたことになります。

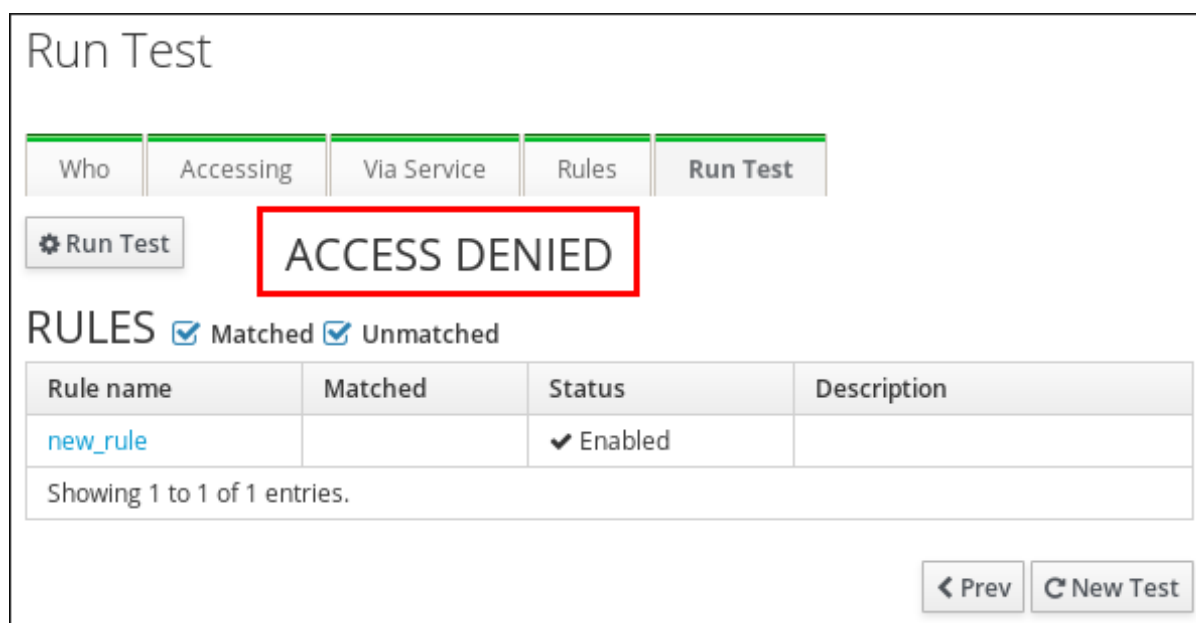


図30.5 HBAC テスト結果の確認

デフォルトでは、IdM はテスト結果を表示する際には、テスト済みの HBAC ルールをすべて表示します。

- アクセスを許可するルールを表示するには、**Matched** を選択してください。
- アクセスを拒否するルールを表示するには、**Unmatched** を選択してください。

コマンドライン: HBAC ルールのテスト

ipa hbactest コマンドを使用して、最低でも以下を指定します。

- ID のテストを実行するユーザー
- ユーザーがアクセスを試みるホスト
- ユーザーが使用を試みるサービス

たとえば、対話式に値を指定する場合には、以下のコマンドを実行します。

```
$ ipa hbactest
User name: user1
Target host: example.com
Service: sudo
-----
Access granted: False
-----
Not matched rules: rule1
```

デフォルトでは、IdM はステータスが **enabled** の HBAC ルールすべてでテストを実行します。別の HBAC ルールを指定するには、以下を実行します。

- HBAC ルールを 1 つまたは複数定義するには、**--rules** オプションを使用します。
- ステータスが **disabled** の HBAC ルールをすべてテストするには **--disabled** オプションを使用します。

HBAC ルールの現在のステータスを表示するには **ipa hbacrule-find** コマンドを使用します。

例30.3 コマンドラインからの HBAC ルールのテスト

以下のテストでは、**rule2** という名前の HBAC ルールを使用して、**user1** が **sudo** を使用して **example.com** にアクセスできないようにします。

```
$ ipa hbactest --user=user1 --host=example.com --service=sudo --
rules=rule1
-----
Access granted: False
-----
Not matched rules: rule1
```

例30.4 コマンドラインからの複数の HBAC ルールのテスト

複数の HBAC ルールをテストする場合は、ユーザーが正常にアクセスできるように許可するルールが 1 つ以上あればテストに合格します。

```
$ ipa hbactest --user=user1 --host=example.com --service=sudo --
rules=rule1 --rules=rule2
-----
Access granted: True
-----
Matched rules: rule2
Not matched rules: rule1
```

出力の内容:

- **Matched rules** では、正常なアクセスを許可するルールを表示します。
- **Not matched rules** はアクセスを拒否するルールを表示します。

30.2.3. HBAC ルールの無効化

HBAC ルールを無効にするとルールが無効になりますが、削除はされません。HBAC ルールを無効にする場合は、後でもう一度有効化することができます。



注記

たとえば、カスタムの HBAC ルールを初めて作成したあとに HBAC ルールを無効にする場合に便利です。また、新規設定がデフォルトの **allow_all** HBAC ルールで上書きされないようにするには **allow_all** を無効にする必要があります。

HBAC ルールを無効にするには、以下を使用します。

- IdM web UI (「[Web UI: HBAC ルールの無効化](#)」を参照)

- コマンドライン (「[コマンドライン: HBAC ルールの無効化](#)」を参照)

Web UI: HBAC ルールの無効化

1. **Policy** → **Host-Based Access Control** → **HBAC Rules**を選択します。
2. 無効化する HBAC ルールを選択して **Disable** をクリックします。

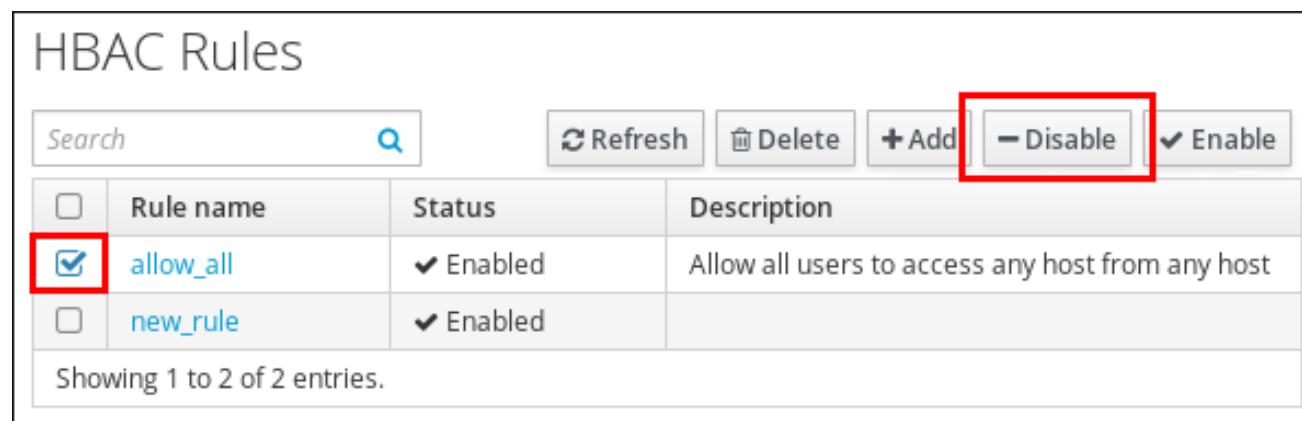


図30.6 allow_all の HBAC ルールの無効化

コマンドライン: HBAC ルールの無効化

ipa hbacrule-disable コマンドを使用します。たとえば、**allow_all** ルールを無効にするには、以下を実行します。

```
$ ipa hbacrule-disable allow_all
-----
Disabled HBAC rule "allow_all"
-----
```

30.3. カスタムの HBAC サービス用に HBAC サービスエントリーの追加

デフォルトでは、最も一般的なサービスおよびサービスグループのみが HBAC ルールとして設定されますが、他の Pluggable Authentication Module (PAM) サービスを HBAC サービスとして設定することもできます。これにより、HBAC ルールでカスタムの PAM サービスを定義することができます。



注記

HBAC サービスとしてのサービスを追加するのと、ドメインにサービスを追加するのは同じではありません。ドメインにサービスを追加すると (「[サービスエントリーおよび Keytab の追加と編集](#)」に記載)、そのサービスはドメインの他のリソースが利用できるようになりますが、HBAC ルールでこのサービスを利用できるわけではありません。

HBAC サービスエントリーを追加するには、以下を使用します。

- IdM web UI (「[Web UI: HBAC サービスエントリーの追加](#)」を参照)
- コマンドライン (「[コマンドライン: HBAC サービスエントリーの追加](#)」を参照)

Web UI: HBAC サービスエントリーの追加

1. **Policy** → **Host-Based Access Control** → **HBAC Services**を選択します。

2. **Add** をクリックして、HBAC サービスエントリーを追加します。

3. サービスの名前を入力して、**Add** をクリックします。

コマンドライン: HBAC サービスエントリーの追加

ipa hbacsvc-add コマンドを使用します。たとえば、**tftp** サービスのエントリーを追加するには以下を実行します。

```
$ ipa hbacsvc-add tftp
-----
Added HBAC service "tftp"
-----
Service name: tftp
```

30.4. HBAC サービスグループの追加

HBAC サービスグループを使用すると HBAC ルール管理を簡素化できます。HBAC ルールに個別のサービスを追加する代わりに、サービスグループ全体を追加することができます。

HBAC サービスグループを追加するには、以下を使用します。

- IdM web UI (「[Web UI: HBAC サービスグループの追加](#)」を参照)
- コマンドライン (「[コマンドライン: HBAC サービスグループの追加](#)」を参照)

Web UI: HBAC サービスグループの追加

1. **Policy → Host-Based Access Control → HBAC Service Groups**を選択します。
2. **Add** をクリックして、HBAC サービスグループを追加します。
3. サービスグループの名前を入力して、**Add and Edit** をクリックします。
4. サービスグループ設定ページで **Add** をクリックして、グループのメンバーとして HBAC サービスを追加します。

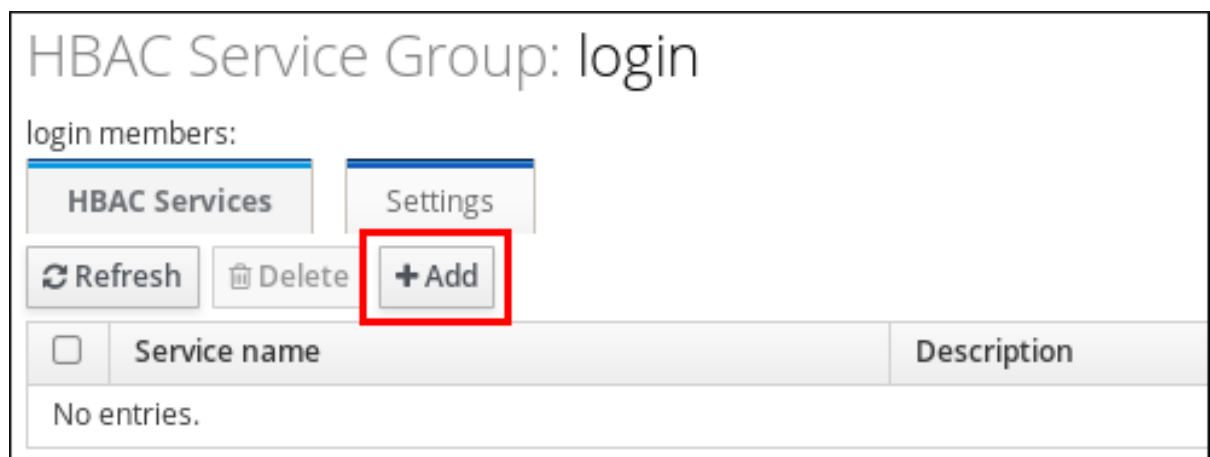


図30.7 HBAC サービスグループへの HBAC サービスの追加

コマンドライン: HBAC サービスグループの追加

1. **ipa hbacsvgroup-add** コマンドを使用して HBAC サービスグループを追加します。**login** という名前のグループを追加するには以下を実行します。

```
$ ipa hbacsvcgroupp-add
Service group name: login
-----
Added HBAC service group "login"
-----
Service group name: login
```

2. グループのメンバーとして HBAC サービスを追加するには **ipa hbacsvcgroupp-add-member** コマンドを使用します。たとえば、**sshd** サービスを **login** グループに追加するには、以下を実行します。

```
$ ipa hbacsvcgroupp-add-member
Service group name: login
[member HBAC service]: sshd
Service group name: login
Member HBAC service: sshd
-----
Number of members added 1
-----
```

第31章 SELINUX ユーザーマップの定義

Security-enhanced Linux (SELinux) は、システムユーザーがどのプロセス、ファイル、ディレクトリー、およびシステム設定にアクセスできるかを指定するルールを設定します。システム管理者とシステムアプリケーションの両方が、ユーザーアクセスと他のアプリケーションからのアクセスを許可、拒否する **セキュリティーコンテキスト** を定義することができます。

Identity Management ドメインでの集中化されたセキュリティーポリシー定義の一部として、Identity Management は IdM ユーザーを (既存の) SELinux ユーザーコンテキストにマッピングして、定義された SELinux ポリシーに基づいてホストごとに IdM ドメイン内のクライアントおよびサービスへのアクセスを許可もしくは制限します。

31.1. IDENTITY MANAGEMENT、SELINUX、およびユーザーのマッピング



注記

Identity Management は、システム上の SELinux コンテキストの作成や編集は行いません。既存のコンテキストをベースとして使用し、(ドメイン内の) IdM ユーザーを (システム上の) SELinux ユーザーにマッピングします。

Security-enhanced Linux は、ユーザー、プロセス、およびアプリケーションがシステム上の他のリソースと対話する方法に関するカーネルレベルの必須アクセス制御です。**コンテキスト** と呼ばれるこれらの対話のルールはシステム上の異なるオブジェクトのデータと動作の特徴を読み取り、各オブジェクトのセキュリティーに関する影響に基づいて **ポリシー** と呼ばれるルールを設定します。これは、データの重要度やアプリケーションの動作を勘案することなく、主にファイルの所有権やユーザー ID について憂慮する高レベルの任意アクセス制御と対照的です。システム上のリソース (ユーザー、アプリケーション、ファイル、プロセス) はすべてにコンテキストが割り当てられます。

システムユーザーは、SELinux **ロール** に関連付けられます。ロールには、多層セキュリティーコンテキスト (MLS) と複数カテゴリーセキュリティーコンテキスト (MCS) の両方が割り当てられます。MLS/MCS コンテキストは、ユーザーをシステム上でアクセス可能なプロセス、ファイル、およびオペレーションに **限定** します。

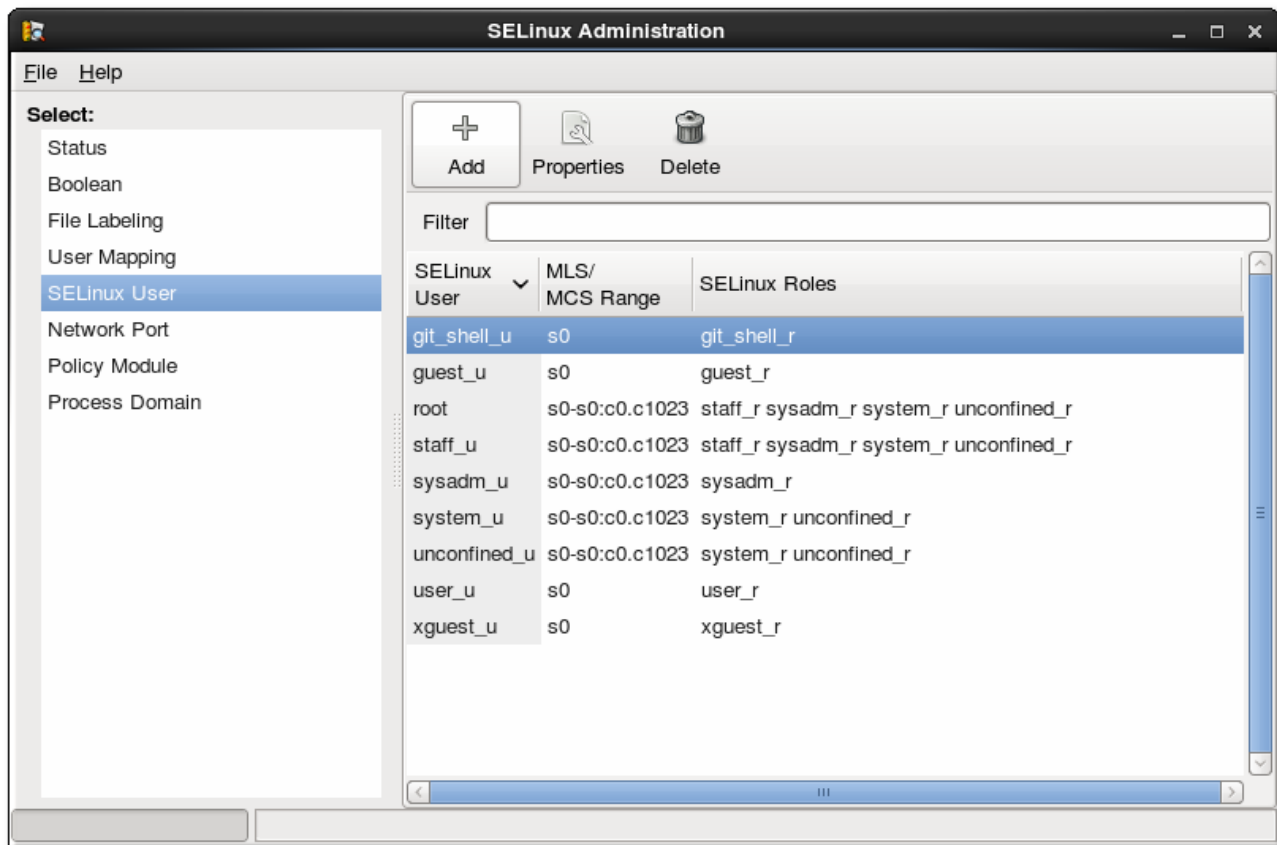


図31.1 SELinux マネージャーの SELinux ユーザー

これについてはすべて、『[Red Hat Enterprise Linux 6 Security-Enhanced Linux ユーザーガイド](#)』で詳述されています。

SELinux のユーザーとポリシーは、ネットワークレベルではなく、システムレベルで機能します。つまり、SELinux ユーザーは、各システムで個別に設定されます。SELinux には共通の定義済みシステムユーザーがあり、SELinux 対応サービスが独自のポリシーを定義しているので、これは多くの状況では受け入れられるものですが、リモートユーザーやローカルリソースにアクセスするシステムについては問題があります。リモートのユーザーとサービスは、それらの本来の SELinux ユーザーやロールについての情報が十分にない状態で、デフォルトのゲストコンテキストにシャッフルされる場合があります。

Identity Management が ID ドメインをローカルの SELinux サービスにうまく統合できるのはこのような状態です。Identity Management は IdM ユーザーを **ホストごとの**設定済み SELinux ロールにマッピングします。SELinux と IdM のユーザーをマッピングすることで、ユーザー管理が改善されます。

- リモートユーザーは、自身の IdM グループ割り当てに基づいて、適切な SELinux ユーザーコンテキストが提供されます。これにより管理者は、ローカルアカウントを作成したり SELinux を再構築することなく一貫して同じポリシーを同じユーザーに適用することもできるようになります。
- ホストが IT 環境に追加されたり、ユーザーが追加、削除、変更されたりすると、ローカルシステムを編集することなく、SELinux ユーザーは自動的に更新されます。
- SELinux ポリシーは、IdM ホストベースのアクセス制御ルールのようなドメイン全体のセキュリティポリシーと関連付けて計画することができます。
- 管理者は環境全体にわたる可視性を持ち、SELinux でユーザーやシステムが割り当てられる方法を制御します。

SELinux ユーザーマップは、システムにおける SELinux ユーザー、IdM ユーザー、および IdM ホスト、という 3 つの部分で構成されています。これらは 2 つの別個の関係を定義します。1 つめは、特定ホスト (ローカルまたはターゲットシステム) 上の SELinux ユーザーのマップを定義します。2 つ目は、SELinux ユーザーと IdM ユーザーのマップを定義します。

この組み合わせにより、管理者はアクセスするホストによって、同一の IdM ユーザーに異なる SELinux ユーザーを設定することが可能になります。

SELinux ユーザーマップは、System Security Services Daemon (SSSD) および **pam_selinux** モジュールと機能します。リモートユーザーがマシンにログインを試みると、SSSD はその IdM ID プロバイダーをチェックして、SELinux マップを含むユーザー情報を収集します。すると PAM モジュールはこのユーザーを処理し、適切な SELinux ユーザーコンテキストを割り当てます。

SELinux マッピングルールの中心となるのは、SELinux システムユーザーです。各マップは、まず SELinux ユーザーに関連付けられます。マッピングに利用可能な SELinux ユーザーは IdM サーバーで設定されるので、集中化された共通のリストがあることになります。これらの SELinux ユーザーは、IdM ドメイン内のそれぞれのホストで設定されたものです。デフォルトでは、以下の 5 つの共通 SELinux ユーザーが定義されています。

- `unconfined_u` (IdM ユーザーにデフォルトとして使用)
- `guest_u`
- `xguest_u`
- `user_u`
- `staff_u`

IdM サーバー設定では、各 SELinux ユーザーは ユーザー名と MLS/MCS の範囲で **SELinux_username:MLS[:MCS]** と設定され、マップ設定時にはこの形式を使用して SELinux ユーザーを識別します。

IdM ユーザーとホストの設定は、非常に柔軟性があります。ユーザーとホストは、明示的かつ個別に SELinux ユーザーマップに割り当てることができます。また、ユーザーグループもしくはホストグループを明示的にマップに割り当てすることもできます。

ホストベースのアクセス制御ルールを使用することで、新たなセキュリティー層が追加されます。ホストベースのアクセス制御ルールでユーザーとホストが定義されていれば、これを SELinux ユーザーマップに使用することが可能です。(「[30章 ホストベースのアクセス制御の設定](#)」で説明しているように) ホストベースのアクセス制御ルールは、SELinux ユーザーマップと IdM 内の他のアクセス制御の統合に役立ち、ローカルセキュリティーのコンテキストを定義するほか、リモートユーザーにおけるホストベースのユーザーアクセスの制限や許可にも役立ちます。



注記

ホストベースのアクセス制御ルールが SELinux ユーザーマップに関連付けられている場合、このルールが SELinux ユーザーマップ設定から除かれるまで削除することはできません。

31.2. SELINUX ユーザーマップの順序とデフォルト値の設定

名前が示すように、SELinux ユーザーマップは SELinux ユーザーと IdM ユーザーの関連付けを作成します。この関連付けを確立する前に、IdM サーバーは、管理対象となるサーバー上の基本的な SELinux ユーザー設定を認識する必要があります。

利用可能な システム SELinux ユーザーマップは、IdM サーバー設定の一部です。これは、制限の強さの順に並んだ SELinux ユーザーの一覧です。SELinux ユーザーエントリー自体は、以下の形式となっています。

```
SELinux_username:MLS[:MCS]
```

個別のユーザーエントリーは、ドル記号 (\$) で区切ります。

ユーザーエントリーにおける SELinux マップの要件はないことから、多くのエントリーはマッピングされていません。IdM サーバー設定では、デフォルトの SELinux ユーザー (SELinux マップ一覧すべてのユーザーの 1 人) がマッピングされていない IdM ユーザーエントリーを使用するように設定します。これにより、マッピングされていない IdM ユーザーでさえも、実用的な SELinux コンテキストを持つことになります。



注記

この設定は、利用可能なシステム SELinux ユーザーのマップ順序を定義します。これは、IdM ユーザーの SELinux ポリシーを定義するものではありません。IdM ユーザーと SELinux ユーザーのマップは、[「SELinux ユーザーの IdM ユーザーへのマッピング」](#)にあるように定義され、それからそのマップにユーザーを追加する必要があります。

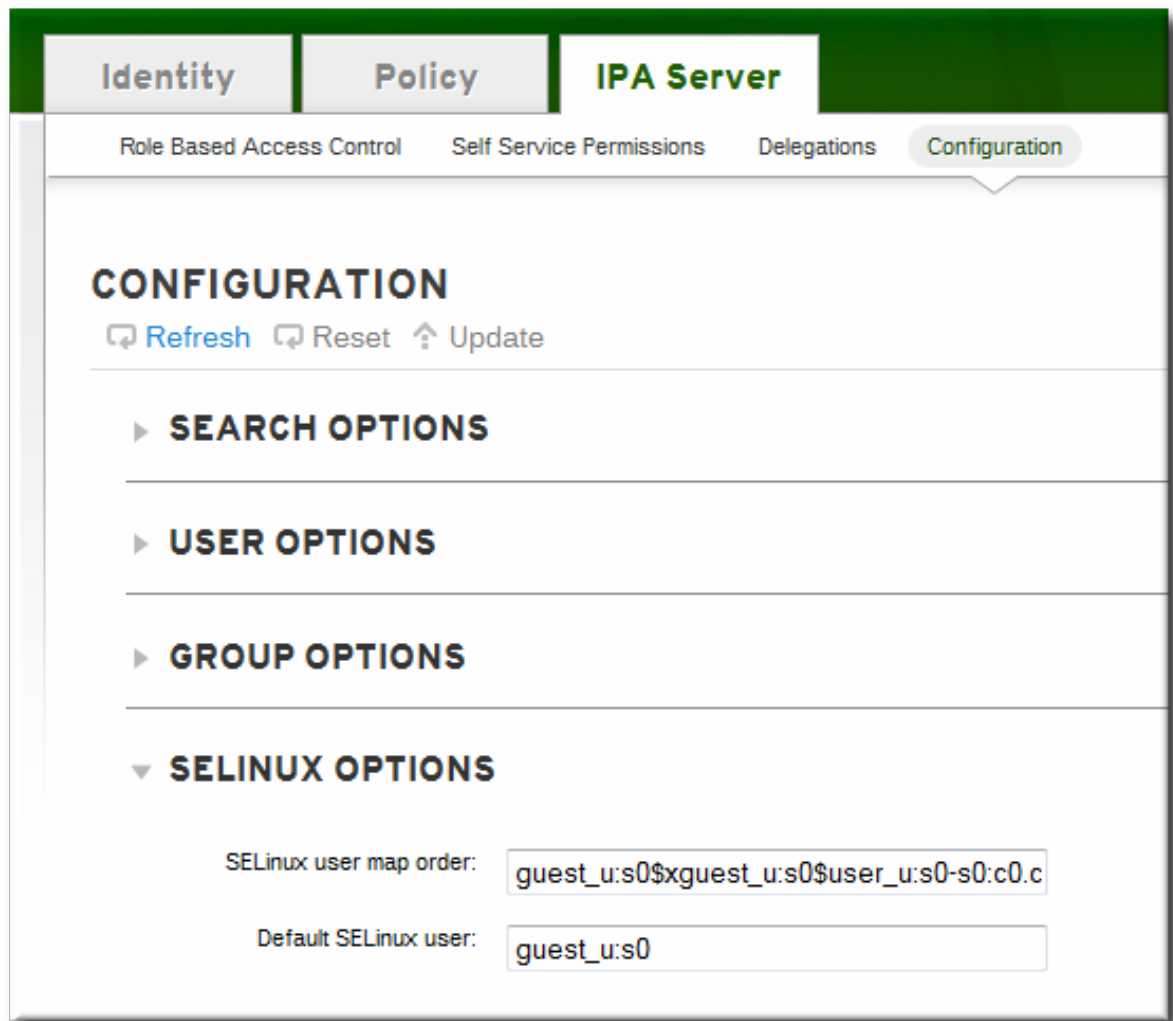
31.2.1. Web UI での設定

1. トップメニューで **IPA Server** メインタブをクリックし、**Configuration** サブタブをクリックします。
2. **SELINUX OPTIONS** まで、サーバー設定エリア一覧でスクロールダウンします。
3. SELinux ユーザー設定を行います。

編集可能なのは、SELinux ユーザーの優先度リストと、マッピングされていない IdM ユーザーに使用するデフォルト SELinux ユーザーの 2 つのエリアです。

SELinux user map order では、ローカルの Linux システムで定義された SELinux ユーザー一覧を提供します。この一覧は、マッピングルールの設定に使用可能です。これは制限の高いものから低いものの順に並んだ優先度の一覧です。各 SELinux ユーザーは、**SELinux_user:MLS** という形式になります。

Default SELinux user フィールドでは、マッピングされていない IdM ユーザーが使用する SELinux ユーザーを設定します。



4. ページ上部にある **Update** をクリックして変更を保存します。

31.2.2. コマンドラインでの設定

SELinux マッピングルールを作成する前に、マッピングに利用可能となる SELinux ユーザーの定義済みかつ汎用の一覧が必要になります。これは、IdM サーバー設定で設定されます。

```
[jsmith@server ~]$ ipa config-show
...
SELinux user map order: guest_u:s0$guest_u:s0$user_u:s0$staff_u:s0-
s0:c0.c1023$unconfined_u:s0-s0:c0.c1023
Default SELinux user: unconfined_u:s0-s0:c0.c1023
```

SELinux ユーザーの設定は、**config-mod** コマンドで編集できます。

例31.1 SELinux ユーザーの一覧

--ipaselinlinuxusermaporder オプションで、SELinux ユーザーの完全一覧を渡します。この一覧では、ユーザーを制限の高い方から低い方の順序で、優先度の順序を設定します。

SELinux ユーザーエントリ自体は、以下の形式となります。

```
SELinux_user:MLS:MCS
```

個別のユーザーエントリは、ドル記号 (\$) で区切ります。

例を示します。

```
[jsmith@server ~]$ ipa config-mod --
ipaselinuusermaporder="unconfined_u:s0-
s0:c0.c1023$guest_u:s0$xguest_u:s0$user_u:s0-s0:c0.c1023$staff_u:s0-
s0:c0.c1023"
```



注記

マッピングされていないエントリーに使用するデフォルトの SELinux ユーザーをユーザーマッピング一覧に含めないと、編集操作は失敗します。同様に、デフォルトを編集する際は、SELinux マッピング一覧にあるユーザーに変更する必要がある、そうでない場合はマッピング一覧を先に更新する必要があります。

例31.2 デフォルトの SELinux ユーザー

IdM ユーザーは、特定の SELinux ユーザーを自分のアカウントにマッピングする必要はありません。ただし、ローカルシステムは、IdM ユーザーアカウントに使用する SELinux ユーザーの IdM エントリーをチェックします。デフォルトの SELinux ユーザーは、マッピングされていない IdM ユーザーエントリーに使用するフォールバックユーザーを設定します。これはデフォルトで、Red Hat Enterprise Linux 上のシステムユーザーのデフォルトの SELinux ユーザーである **unconfined_u** です。

このデフォルトユーザーは、**--ipaselinuusermapdefault** で変更できます。例を示します。

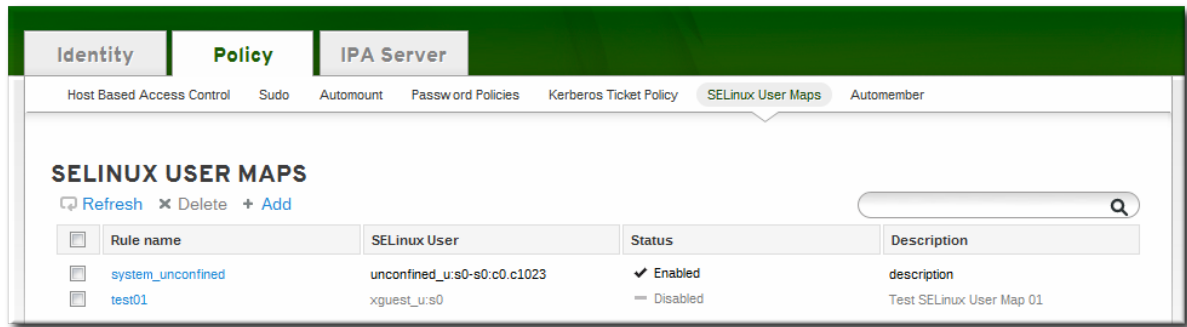
```
[jsmith@server ~]$ ipa config-mod --
ipaselinuusermapdefault="guest_u:s0"
```

31.3. SELINUX ユーザーの IDM ユーザーへのマッピング

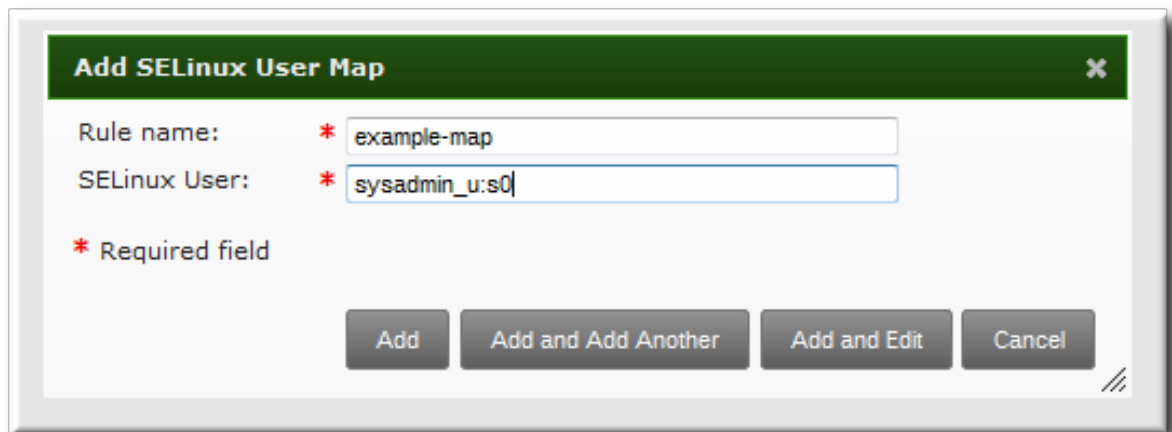
SELinux マップは、ローカルシステム上の SELinux ユーザーコンテキストをドメイン内の (単一または複数の) IdM ユーザーに関連付けます。SELinux マップは、SELinux ユーザーコンテキストと IdM ユーザー/ホストのペアという 3 つの部分で構成されています。この IdM ユーザー/ホストのペアは、以下のいずれかの方法で定義できます。明示的なホスト上の明示的なユーザー (またはユーザーおよびホストグループ) に設定するか、ホストベースのアクセス制御ルールを使って定義できます。

31.3.1. Web UI での設定

1. トップメニューで **Policy** メインタブをクリックし、**SELinux User Maps** サブタブをクリックします。
2. マッピングのリストで **Add** をクリックして新規マップを作成します。



3. IdM サーバー設定で表示されるものと全く同一になるようにマップと SELinux ユーザーの名前を入力します。SELinux ユーザーの形式は、**SELinux_username:MLS[:MCS]** となります。



4. **Add and Edit** をクリックして IdM ユーザー情報を追加します。
5. ホストベースのアクセス制御ルールを設定するには、設定の **General** エリアでドロップダウンメニューからルールを選択します。ホストベースのアクセス制御ルールを使用すると、リモートユーザーがターゲットマシンにアクセスする際に使用するホストでアクセス制御が導入されます。割り当て可能なホストベースのアクセス制御ルールは、**1 つのみ**です。



注記

ホストベースのアクセス制御ルールには、サービスだけでなく、ユーザーとホストも含める必要があります。

The screenshot shows the SELinux User Maps configuration interface. At the top, the breadcrumb is 'SELinux User Maps » user'. The main title is 'SELINUX USER MAP: user'. Below the title is a 'Settings' tab. Under the tab are three buttons: 'Refresh', 'Reset', and 'Update'. A section titled 'GENERAL' is expanded, showing the following fields: 'Rule name' is 'user'; 'Description' is an empty text area; 'SELinux User' is 'unconfined_u:s0-s0:c0.c1023'; 'HBAC Rule' is 'web_admin' with an 'undo' button; and 'Status' is 'Enabled' (selected) with a 'Disabled' option.

SELinux User Maps » user

SELINUX USER MAP: user

Settings

Refresh Reset Update

GENERAL

Rule name: user

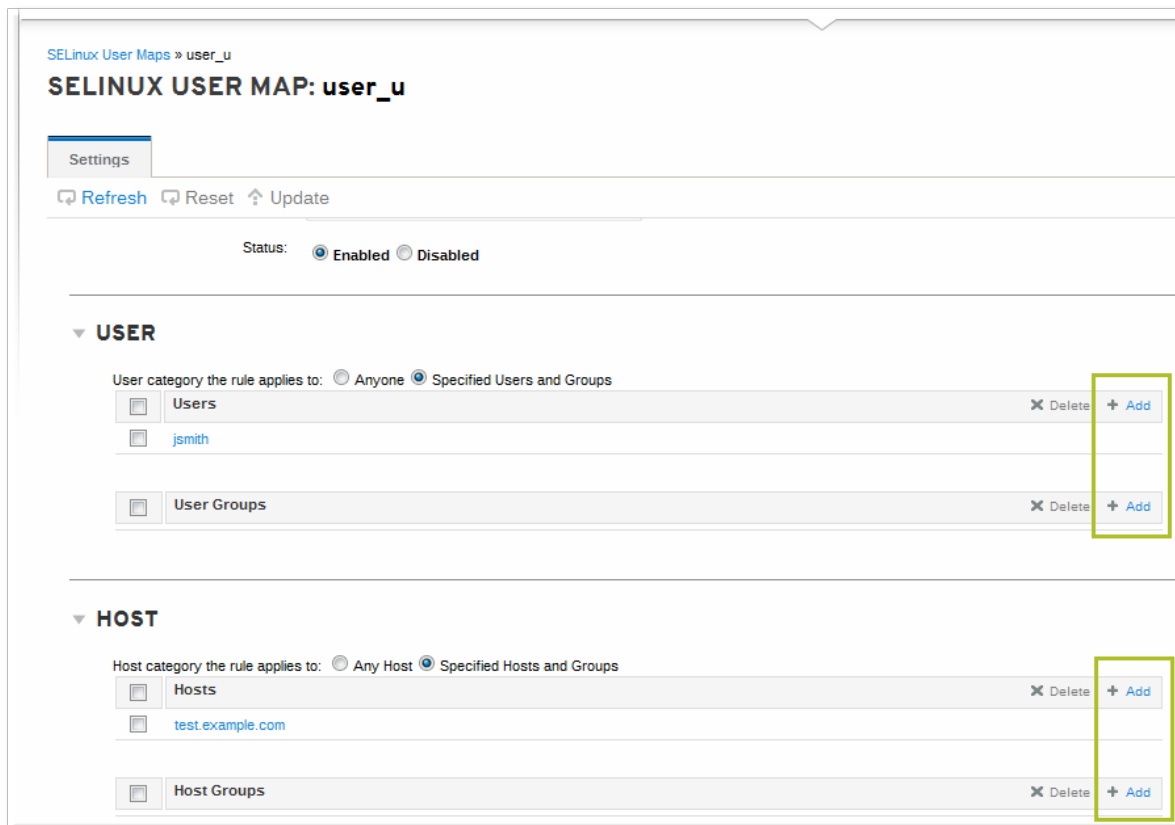
Description:

SELinux User: * unconfined_u:s0-s0:c0.c1023

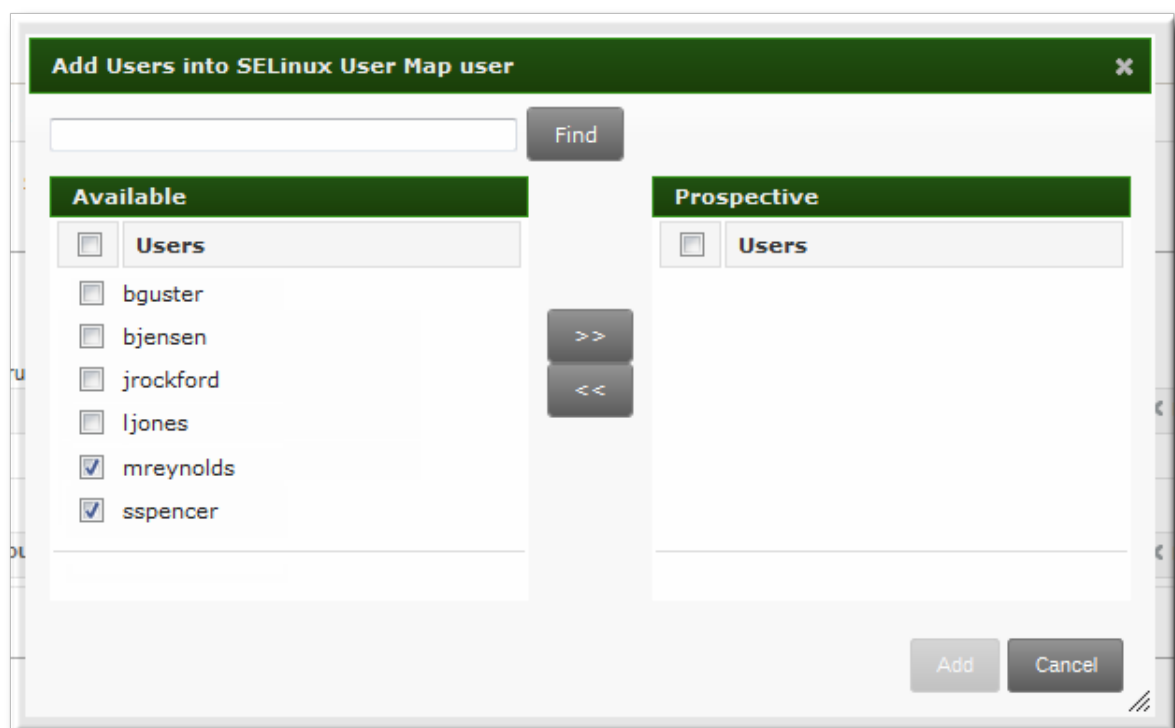
HBAC Rule: web_admin undo

Status: ☒ Enabled ☐ Disabled

別の方法では、**Users** と **Hosts** のエリアでスクロールダウンし、**Add** をクリックしてユーザー、ユーザーグループ、ホスト、もしくはホストグループを SELinux マップに割り当てます。



左側のユーザー（またはホストもしくはグループ）を選択し、右矢印 >> をクリックして **Prospective** コラムに移動します。**Add** をクリックして、それらをルールに追加します。



注記

ホストベースのアクセス制御ルールを提供するか、ユーザーおよびホストを手動で設定するかのどちらになります。両方のオプションを同時に使用することはできません。

6. 上部の **Update** をクリックして、SELinux ユーザーマップへの変更を保存します。

31.3.2. コマンドラインでの設定

SELinux マップルールには、以下の 3 つの基礎的部分があります。

- SELinux ユーザー (**--selinuxuser**)
- SELinux ユーザーに関連付けられたユーザーもしくはユーザーグループ (**--users** または **--groups**)
- SELinux ユーザーに関連付けられたホストもしくはホストグループ (**--hosts** または **--hostgroups**)
- 代替方法として、ホストおよびユーザーを指定しているホストベースのアクセス制御ルール (**-hbacrule**)

selinuxusermap-add コマンドを使うと、すべての情報があるルールを 1 度で作成できます。**selinuxusermap-add-user** および **selinuxusermap-add-host** のコマンドを使用すると、ルール作成後にユーザーとホストをそれぞれ追加できます。

例31.3 新規 SELinux マップの作成

--selinuxuser の値は、IdM サーバー設定で表示される SELinux ユーザー名と全く同一にする必要があります。SELinux ユーザーの形式は、**SELinux_username:MLS[:MCS]** となります。

SELinux のマッピングが有効になるには、ユーザーおよびホストの両方 (または適切なグループ) を指定する必要があります。ユーザー、ホスト、およびグループのオプションは複数回使用するか、**-option={val1,val2,val3}** のように中括弧内でコンマ区切りにして 1 回の使用とすることができます。

```
[jsmith@server ~]$ ipa selinuxusermap-add --users=jsmith --users=bjensen
--users=jrockford --hosts=server.example.com --hosts=test.example.com --
selinuxuser="xguest_u:s0" selinux1
```

例31.4 ホストベースのアクセス制御ルールでの SELinux マップ作成

--hbacrule の値は、マッピングに使用するホストベースのアクセス制御ルールを特定します。ホストベースのアクセス制御ルールを使用すると、リモートユーザーがターゲットマシンにアクセスする際に使用するホストでアクセス制御が導入されます。また、リモートユーザーがターゲットマシンにログインすると、SELinux コンテキストが適用されます。

アクセス制御ルールでユーザーとホストの両方が適切に指定されると、SELinux マップは SELinux ユーザー、IdM ユーザー、およびホストの 3 つを構築できます。

指定可能なホストベースのアクセス制御ルールは、1 つのみです。

```
[jsmith@server ~]$ ipa selinuxusermap-add --hbacrule=webserver --
selinuxuser="xguest_u:s0" selinux1
```

ホストベースのアクセス制御ルールは、「[30章 ホストベースのアクセス制御の設定](#)」で説明しています。

例31.5 ユーザーを SELinuxマッピングに追加する

ユーザーとホストはマップの作成時に追加できますが、これらはルール作成後にも追加できます。これを行うには、**selinuxusermap-add-user** または **selinuxusermap-add-host** のコマンドを使用します。

```
[jsmith@server ~]$ ipa selinuxusermap-add-user --users=jsmith selinux1
```

追加するホストベースのアクセス制御ルールは 1 つのみなので、ルール設定後にこれを追加するために、別のコマンドを使用する必要はありません。**selinuxusermap-mod** コマンドを **--hbacrule** オプションと使用すると、ホストベースのアクセス制御ルールが追加されるか、既存のものが上書きされます。

例31.6 ユーザーの SELinuxマッピングからの削除

特定のユーザーもしくはホストを SELinux マップから削除するには、**selinuxusermap-remove-host** または **selinuxusermap-remove-user** のコマンドを実行します。例を示します。

```
[jsmith@server ~]$ ipa selinuxusermap-remove-user --users=jsmith  
selinux1
```

パート **VII.** ネットワークサービスの管理

第32章 DNS の管理

Identity Management サーバーは統合 DNS サービスなしでインストールすることが可能なので、外部 DNS サービスを使用したり、DNS が設定された状態で使用することが可能です。詳細は、「[IdM サーバーのインストール: はじめに](#)」および「[統合 DNS 使用の判断](#)」を参照してください。

DNS サービスがドメイン内で設定される場合、IdM では管理者に多大な柔軟性と DNS 設定に関する制御がもたらされます。たとえば、ホストエントリー、場所、レコードなどのドメインの DNS エントリーをネイティブの IdM ツールで管理したり、クライアントが自身の DNS レコードを動的に更新したりできるようになります。

BIND と IdM ではほとんどの設定オプションが同じ方法で機能することから、BIND バージョン 9.9 で利用可能なドキュメント資料およびチュートリアルは IdM DNS に適用できます。本章では主に、BIND と IdM の注意すべき違いについて説明します。

32.1. IDENTITY MANAGEMENT における BIND

IdM は、BIND DNS サーバーのバージョン 9.9 をデータ複製に使用する LDAP データベースおよび、GSS-TSIG プロトコルを使用した DNS 更新署名用に Kerberos と統合します。^[3] これにより、IdM ツールを使用した DNS 管理が可能になり、同時に回復性が高まります。これは、IdM と統合された DNS サーバーは複数のマスター操作をサポートするので、IdM 統合の DNS サーバーはすべて、単一障害点を持つことなく、クライアントからの DNS 更新を受け付けることが可能になるからです。

デフォルトの IdM DNS 設定は、公開インターネットからアクセスできない内部ネットワークに適しています。IdM DNS サーバーに公開インターネットからアクセスできる場合は、[Red Hat Enterprise Linux ネットワークガイド](#)の説明にあるように、通常のセキュリティー機能を BIND に適用することを Red Hat では推奨しています。



注記

IdM 統合の BIND を **chroot** 環境内で実行することはできません。

IdM 統合の BIND は、**bind-dyndb-ldap** プラグインを使って Directory Server と通信します。IdM は BIND 向けに `/etc/named.conf` ファイル内に **dynamic-db** 設定セクションを作成します。これは、BIND **named-pkcs11** サービス用の **bind-dyndb-ldap** プラグインを設定するものです。

標準の BIND と IdM DNS との最も大きな違いは、IdM は DNS 情報を LDAP エントリーとして保存するという点です。ドメイン名はそれぞれ LDAP エントリーとして示され、リソースレコードはすべて LDAP エントリーの LDAP 属性として保存されます。たとえば、以下の **client1.example.com.** ドメイン名には 3 つの A レコードと AAAA レコードが 1 つ含まれています。

```
dn: idnsname=client1,idnsname=example.com.,cn=dns,dc=idm,dc=example,dc=com
objectclass: top
objectclass: idnsrecord
idnsname: client1
Arecord: 192.0.2.1
Arecord: 192.0.2.2
Arecord: 192.0.2.3
AAAArecord: 2001:DB8::ABCD
```




重要

DNS データや BIND 設定を編集する場合には、常に本章記載の IdM ツールを使用してください。

32.2. サポートされる DNS ゾーンタイプ

IdM は *master* と *forward* の 2 つの DNS ゾーンタイプをサポートします。



注記

本ガイドではゾーンタイプの BIND 用語に、Microsoft Windows DNS で使用されている用語とは別のものを使用しています。BIND における Master ゾーンは、Microsoft Windows DNS の *前方参照ゾーン* および *逆引き参照ゾーン* の用途と同じものになります。BIND の Forward ゾーンは、Microsoft Windows DNS の *条件付きフォワーダー* と同じ用途になります。

Master DNS ゾーン

Master DNS ゾーンには権威 DNS データが含まれており、動的な DNS 更新を受け付けることができます。この動作は、標準の BIND 設定における **type master** 設定と同等のものです。Master ゾーンは、**ipa dnszone-*** コマンドを使って管理します。

標準の DNS ルールに従い、各 master ゾーンには SOA と NS のレコードを含める必要があります。DNS ゾーンの作成時に IdM は自動的にこれらのレコードを生成しますが、NS レコードは手動で親ゾーンにコピーして適切な委任を作成する必要があります。

標準の BIND 動作に従い、master ゾーンに指定された転送設定は、サーバーに権威のない名前への影響します。

例32.1 DNS 転送の例

IdM サーバーも **test.example.** master ゾーンが含まれているとします。このゾーンには、**sub.test.example.** 名の NS 委任レコードが含まれています。また、**test.example.** ゾーンでは、**192.0.2.254** というフォワーダー IP アドレスが設定されています。

nonexistent.test.example. という名前をクエリーするクライアントは **NXDomain** という応答を受け取り、転送は発生しません。これはこの名前に対して IdM サーバーが権威があるためです。

一方、**sub.test.example.** という名前に対するクエリーは、設定された **192.0.2.254** というフォワーダーに転送されます。これは、IdM サーバーがこの名前に対して権限がないためです。

正引き DNS ゾーン

正引き DNS ゾーンには権威 DNS データが含まれていません。正引き DNS ゾーンに属する名前へのクエリーはすべて、指定されたフォワーダーに転送されます。この動作は、標準の BIND 設定における **type forward** 設定と同等のものです。Forward ゾーンは、**ipa dnsforwardzone-*** コマンドを使って管理します。

32.3. DNS 設定の優先順位

DNS 設定オプションの多くは 3 つの異なるレベルで設定ができます。

ゾーン固有の設定

IdM 内で定義された特定ゾーンに固有の設定レベルが、最も高い優先順位になります。ゾーン固有の設定は、**ipa dnszone-*** および **ipa dnsforwardzone-*** コマンドを使って管理します。

グローバル DNS 設定

ゾーン固有の設定が定義されていない場合、IdM は LDAP に保存されているグローバル DNS 設定を使用します。グローバル DNS 設定は、**ipa dnsconfig-*** コマンドを使って管理します。グローバル DNS 設定で定義されている設定は、すべての IdM DNS サーバーに適用されます。

/etc/named.conf での設定

各 IdM DNS サーバーにある **/etc/named.conf** ファイルで定義されている設定は、優先順位が一番低くなります。これは各サーバーに固有のもので、手動で編集する必要があります。

/etc/named.conf ファイルは通常、ローカル DNS キャッシュへの DNS 転送を指定するためにのみ使用されます。他のオプションは、上記のゾーン固有およびグローバル DNS 設定のコマンドを使用して管理します。

DNS オプションは複数のレベルで同時に設定することが可能です。その場合、より高い優先度の設定が低いものよりも優先されます。

32.4. MASTER DNS ゾーンの管理

32.4.1. Master DNS ゾーンの追加と削除

Web UI での Master DNS ゾーンの追加

1. **Network Services** タブから **DNS** サブタブを開き、**DNS Zones** セクションを選択します。

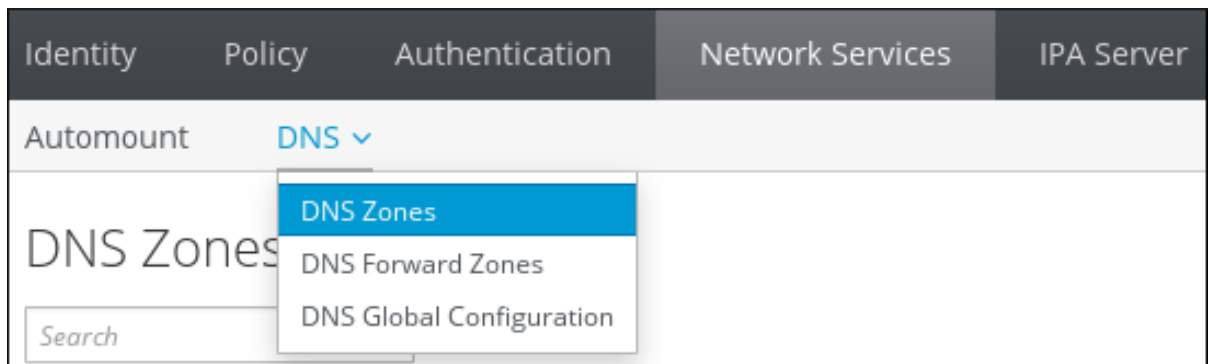


図32.1 Master DNS ゾーンの管理

2. 新規の master ゾーンを追加するには、全ゾーン一覧上部にある **Add** をクリックします。

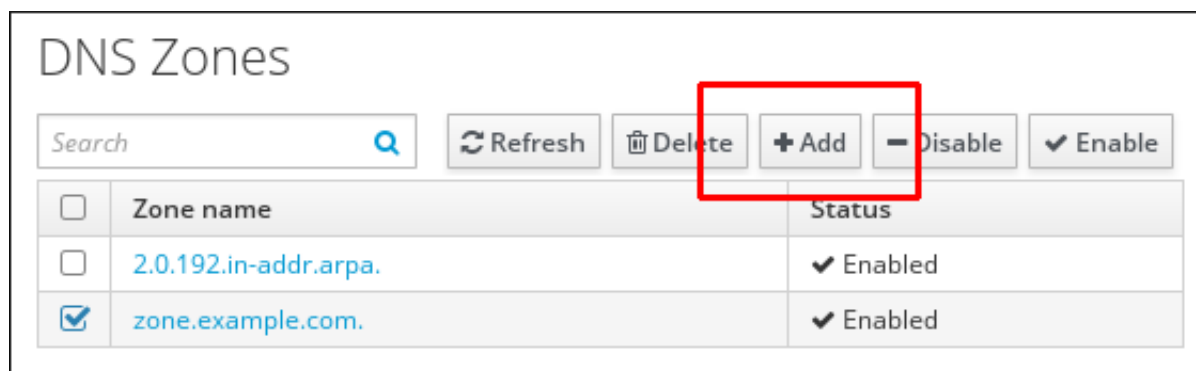


図32.2 Master DNS ゾーンの追加

3. ゾーン名を入力して **Add** をクリックします。

Add DNS Zone

☒ Zone name *

☐ Reverse zone

IP network

* Required field

図32.3 新規 Master ゾーンの入力

コマンドラインからの Master DNS ゾーンの追加

ipa dnszone-add コマンドで、新規ゾーンが DNS ドメインに追加されます。新規ゾーンを追加するには、新規サブドメインの名前を指定する必要があります。下記のようにこのコマンドでサブドメイン名を直接渡すことができます。

```
$ ipa dnszone-add newserver.example.com
```

ipa dnszone-add で名前を渡さない場合は、スクリプトが自動的にこれを要求します。

ipa dnszone-add コマンドは各種のコマンドラインオプションも受け付けます。オプションの完全一覧を確認するには、**ipa dnszone-add --help** コマンドを実行してください。

Master DNS ゾーンの削除

Web UI でマスター DNS ゾーンを削除するには、該当するゾーン名を選択して **Delete** をクリックします。

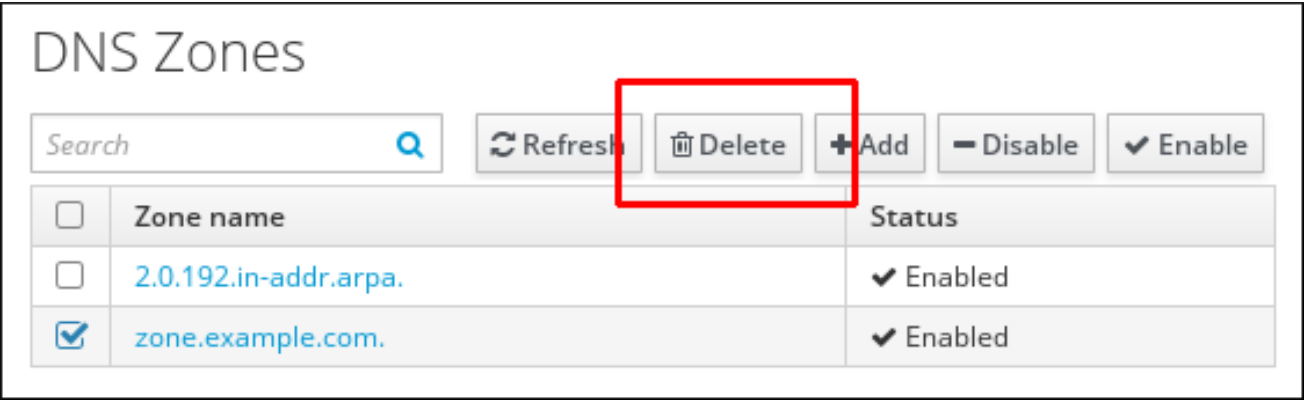


図32.4 Master DNS ゾーンの削除

コマンドラインからマスター DNS ゾーンを削除するには、以下のように **ipa dnszone-del** コマンドを実行します。

```
$ ipa dnszone-del server.example.com
```

32.4.2. マスター DNS ゾーンの追加設定

IdM は、リフレッシュの期間や転送設定、キャッシュ設定など、特定のデフォルト設定による新規ゾーンを作成します。

DNS ゾーン設定の属性

表32.1「ゾーン属性」には、使用可能なゾーン設定が記載されています。ここではゾーンの実際の情報を設定するほか、DNS サーバーが *start of authority* (SOA) レコードエントリーを処理する方法と、DNS ネームサーバーからの記録を更新する方法を定義します。

表32.1 ゾーン属性

属性	コマンドラインオプション	説明
Authoritative name server	--name-server	マスター DNS ネームサーバーのドメイン名 (SOA MNAME と呼ぶ) を設定します。 デフォルトでは、各 IdM サーバーが自身を SOA MNAME フィールドに公開します。この結果、 --name-server を使用して LDAP に保存された値は無視されます。
Administrator e-mail address	--admin-email	ゾーン管理者が使用する電子メールアドレスを設定します。デフォルトでは、ホストの root アカウントになります。
SOA serial	--serial	Sets a serial number in the SOA レコードのシリアル番号を設定します。IdM はバージョン番号を自動的に設定し、これはユーザーが編集しないことになっています。
SOA refresh	--refresh	セカンダリー DNS サーバーがプライマリ DNS サーバーから更新を要求するまでの待機時間を秒単位で設定します。

属性	コマンドラインオプション	説明
SOA retry	--retry	失敗したりフレッシュ動作を再試行するまでの待機時間を秒単位で設定します。
SOA expire	--expire	セカンダリー DNS サーバーがリフレッシュ更新を試行して、その動作を停止するまでの時間を秒単位で設定します。
SOA minimum	--minimum	RFC 2308 に従って、ネガティブキャッシュの有効時間 (TTL) の値を秒単位で設定します。
SOA time to live	--ttl	ゾーン apex のレコードの TTL を秒単位で設定します。たとえば、ゾーン example.com では、 example.com 名の全レコード (A、NS、または SOA) が設定されますが、 test.example.com のような他のドメイン名は影響を受けません。
Default time to live	--default-ttl	個別の TTL 値が設定されたことのないゾーンの全値について、ネガティブキャッシュの有効時間 (TTL) のデフォルト値を秒単位で設定します。変更を反映するには、全 IdM DNS サーバーで named-pkcs11 サービスを再起動する必要があります。
BIND update policy	--update-policy	DNS ゾーンでクライアントに許可されるパーミッションを設定します。 更新ポリシーの構文については、 Dynamic Update Policies in the 『BIND 9 Administrator Reference Manual』 を参照してください。
Dynamic update	--dynamic-update=TRUE FALSE	クライアントの DNS レコードへの動的更新を有効にします。 これを false に設定すると、IdM クライアントマシンは IP アドレスの追加や更新ができなくなります。詳細情報は、「 動的 DNS 更新の有効化 」を参照してください。
Allow transfer	--allow-transfer=string	指定されたゾーンの転送が可能な IP アドレスまたはネットワーク名をセミコロン区切りのリストで提供します。 ゾーン転送はデフォルトでは無効になっています。- allow-transfer のデフォルト値は none です。
Allow query	--allow-query	DNS クエリーの発行が可能な IP アドレスまたはネットワーク名をセミコロン区切りのリストで提供します。
Allow PTR sync	--allow-sync-ptr=1 0	ゾーンの A または AAAA レコード (正引きレコード) が自動的に PTR (逆引き) レコードと同期されるかどうかを設定します。

属性	コマンドラインオプション	説明
Zone forwarders	-- forwarder =IP_address	DNS ゾーン向けに特別に設定されたフォワーダーを指定します。これは、IdM ドメインで使用するグローバルのフォワーダーとは別のものです。複数のフォワーダーを指定するには、このオプションを複数回使用します。
Forward policy	-- forward-policy =none only first	転送ポリシーを指定します。サポートされるポリシーについての情報は、「 転送ポリシー 」を参照してください。

Web UI でのゾーン設定編集

Web UI で DNS マスターゾーンを管理するには、**Network Services** タブから **DNS** サブタブを開き、**DNS Zones** セクションを選択します。

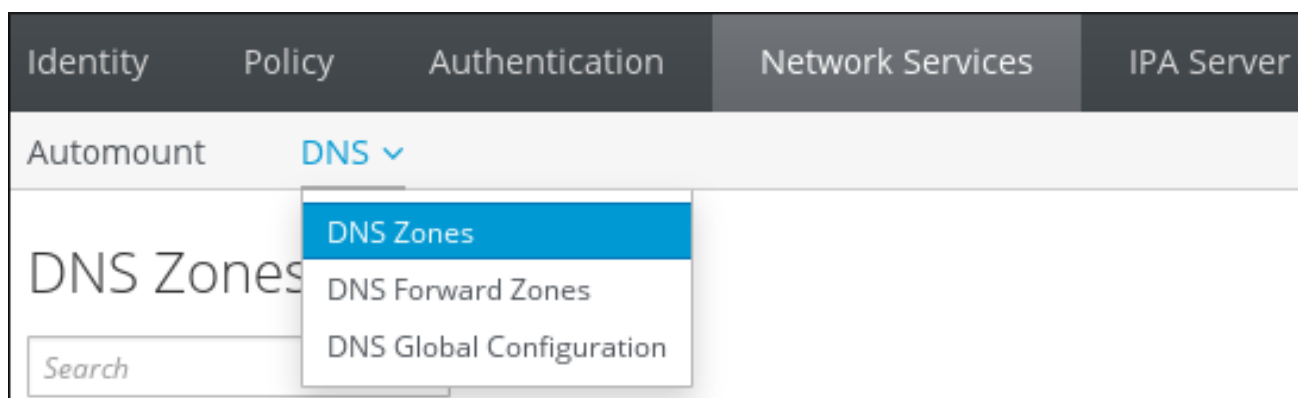


図32.5 Master DNS ゾーンの管理

DNS Zones セクションで既存のマスターゾーンを編集するには、以下の手順に従います。

1. ゾーンの全一覧からゾーン名をクリックして DNS ゾーンページを開きます。

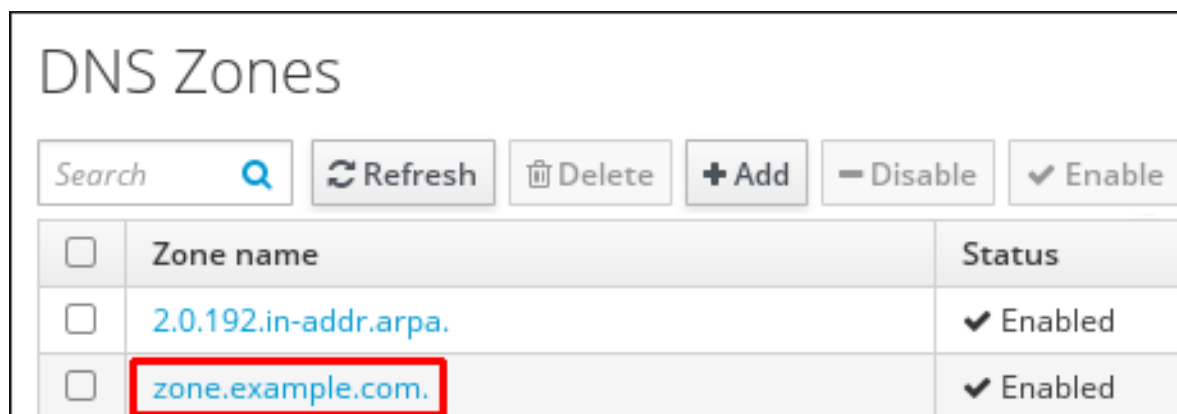


図32.6 マスターゾーンの編集

2. **Settings** をクリックして、ゾーン設定を変更します。

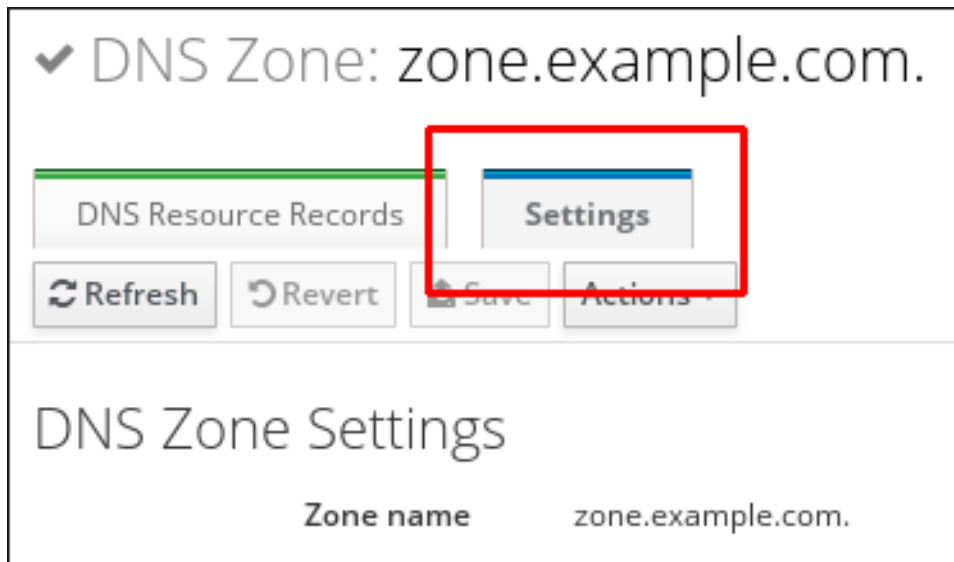


図32.7 マスターゾーン編集ページの **Settings** タブ

利用可能な設定については、[表32.1「ゾーン属性」](#) を参照してください。

3. **Save** をクリックして新規設定を保存します。



注記

ゾーンの有効時間 (TTL) のデフォルト値を変更して反映させるには、全 IdM DNS サーバーで **named-pkcs11** サービスを再起動する必要があります。その他の設定は、即座に有効になります。

コマンドラインでのゾーン設定編集

コマンドラインから既存のマスター DNS ゾーンを変更するには、**ipa dnszone-mod** コマンドを使用します。利用可能な設定については、[表32.1「ゾーン属性」](#) を参照してください。

DNS ゾーンエントリー内に属性がない場合は、**ipa dnszone-mod** コマンドは属性を追加します。属性がある場合は、指定された値で現行の値を上書きします。

ipa dnszone-mod コマンドおよびそのオプションについての詳細は、**ipa dnszone-mod --help** コマンドを実行してください。



注記

ゾーンの有効時間 (TTL) のデフォルト値を変更して反映させるには、全 IdM DNS サーバーで **named-pkcs11** サービスを再起動する必要があります。その他の設定は、即座に有効になります。

32.4.3. ゾーン転送の有効化

ネームサーバーは、ゾーンに対して権限のあるデータを維持します。ゾーンに変更が加えられると、この変更は DNS ドメインのネームサーバーに送信、配布される必要があります。ゾーン転送では、リソースレコードすべてがあるサーバーから別のサーバーにコピーされます。

IdM では、[RFC 5936](#) (AXFR) および [RFC 1995](#) (IXFR) 標準に準拠したゾーン転送をサポートしています。



重要

IdM で統合された DNS には複数のマスターがあります。IdM ゾーン内の SOA シリアル番号は、IdM サーバー間では同期されません。このため、DNS スレーブサーバーが 1 つの IdM マスターサーバーのみを使用するように設定します。これにより、SOA シリアル番号が同期されないことでゾーン転送が失敗するということが回避されます。

UI でのゾーン転送の有効化

「[Web UI でのゾーン設定編集](#)」にあるように DNS ゾーンページを開き、**Settings** タブに切り替えます。

Allow transfer セクションで、ゾーンレコードの転送先隣るネームサーバーを指定します。

図32.8 ゾーン転送の有効化

DNS ゾーンページ上部にある **Save** をクリックして新規設定を保存します。

コマンドラインでのゾーン転送の有効化

コマンドラインでゾーン転送を有効にするには、**--allow-transfer** オプションを **ipa dnszone-mod** コマンドに追加します。以下のように、**--allow-transfer** を使ってゾーンレコードを転送するネームサーバーの一覧を指定します。

```
[user@server ~]$ ipa dnszone-mod --allow-transfer=192.0.2.1;198.51.100.1;203.0.113.1 example.com
```

bind サービス内でゾーン転送を有効にすると、**dig** ユーティリティーのようなクライアントで IdM DNS ゾーンを名前転送できるようになります。

```
[root@server ~]# dig @ipa-server zone_name AXFR
```

32.4.4. DNS ゾーンへのレコードの追加

IdM では多くのレコードタイプをサポートしていますが、よく使われるのは以下の 4 つのタイプです。

A

これはホスト名および通常の IPv4 アドレスの基本的なマップです。A レコードのレコード名は、**www** などのホスト名です。**IP Address** の値は、**192.0.2.1** などの標準的な IPv4 アドレスになります。

A レコードに関する詳細情報は、[RFC 1035](#) を参照してください。

AAAA

これはホスト名および IPv6 アドレスの基本的なマップです。AAAA レコードのレコード名は、**www** などのホスト名です。**IP Address** の値は、**2001:DB8::1111** などの 16 進数の標準的な IPv6 アドレスになります。

AAAA レコードに関する詳細情報は、[RFC 3596](#) を参照してください。

SRV

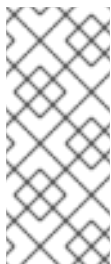
サービス (SRV) リソースレコードは、サービス名を、その特定サービスを提供するサーバーの DNS 名にマッピングします。たとえば、このタイプのレコードは LDAP ディレクトリーのようなサービスを管理するサーバーに、このサービスをマッピングします。

SRV レコードのレコード名は **_service._protocol** という形式になり、たとえば **_ldap._tcp** となります。SRV レコードの設定オプションには、優先順位、加重、ポート番号、およびターゲットサービスのホスト名などがあります。

SRV レコードに関する詳細情報は、[RFC 2782](#) を参照してください。

PTR

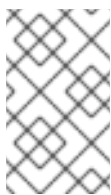
ポインター (PTR) レコードは、IP アドレスをドメイン名にマッピングする逆引き DNS レコードを追加します。



注記

IPv4 アドレスの逆引き DNS ルックアップはすべて、**in-addr.arpa.** ドメインで定義される逆引きエントリーを使用します。ヒューマンリーダブルな形式の逆アドレスは、通常の IP とまったく逆のもので、**in-addr.arpa.** ドメインが最後に付いています。たとえば、ネットワークアドレス **192.0.2.0/24** の逆引きゾーンは、**2.0.192.in-addr.arpa** になります。

PTR レコードのレコード名は、[RFC 1035](#)、[RFC 2317](#)、および [RFC 3596](#) に指定されている標準形式である必要があります。ホスト名の値は、レコードを作成するホストの正規のホスト名である必要があります。詳細は [例32.8「PTR レコード」](#) を参照してください。



注記

.ip6.arpa. ドメイン内のゾーンについても、IPv6 アドレスの逆引きゾーンを設定できます。IPv6 逆引きゾーンについての詳細情報は、[RFC 3596](#) を参照してください。

DNS リソースレコードを追加する際には、多くのレコードで異なるデータが必要になることに留意してください。たとえば、CNAME レコードではホスト名が必要ですが、A レコードでは IP アドレスが必要になります。Web UI では、新規レコード追加のフォームにおけるフィールドでは、選択しているレコードタイプに必要なデータが自動的に反映されます。

DNS のワイルドカードのサポート

IdM は、DNS ゾーン内の特別レコード * をワイルドカードとしてサポートします。

例32.2 DNS ワイルドカードの例

1. DNS ゾーン **example.com** で以下を設定します。

- 。ワイルドカード A レコード *.example.com
 - 。mail.example.com の MX レコード。ただし、このホストでは A レコードなしとします。
 - 。demo.example.com ではレコードなしとします。
2. 既存および存在しない DNS レコードとタイプをクエリーシマス。以下の結果が返されます。

```
# host -t MX mail.example.com.
mail.example.com mail is handled by 10 server.example.com.

# host -t MX demo.example.com.
demo.example.com. has no MX record.

# host -t A mail.example.com.
mail.example.com has no A record

# host -t A demo.example.com.
random.example.com has address 192.168.1.1
```

詳細は、[RFC1034](#) を参照してください。

Web UI での DNS リソースレコードの追加

1. 「Web UI でのゾーン設定編集」にあるように DNS ゾーンページを開きます。
2. DNS Resource Records セクションで、Add をクリックして新規レコードを追加します。

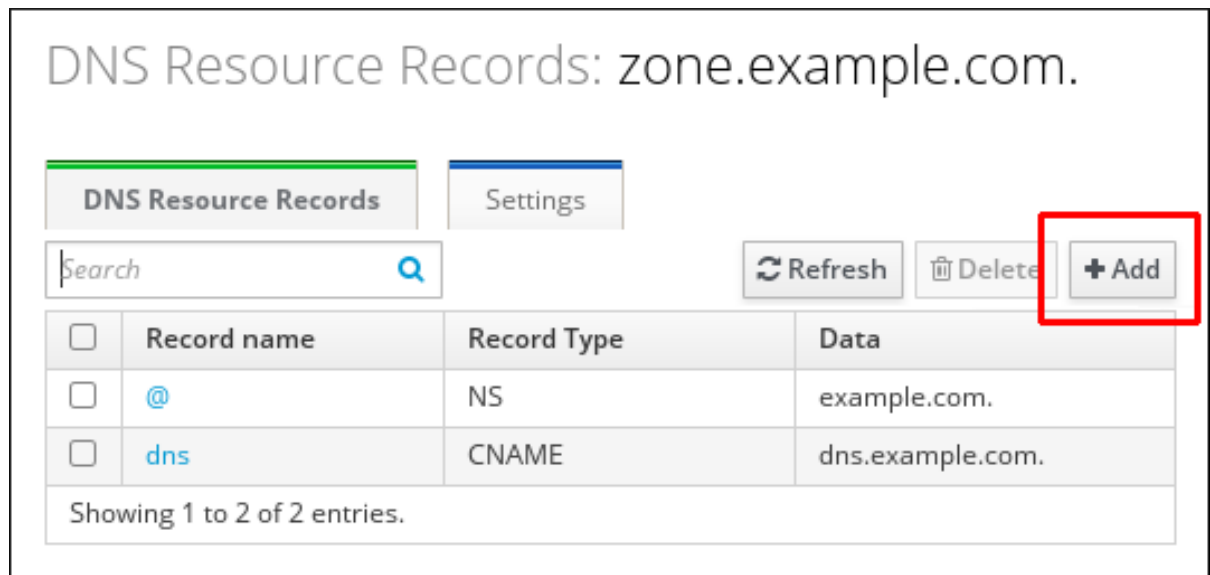


図32.9 新規 DNS リソースレコードの追加

3. 作成するレコードタイプを選択し、必要なフィールドに入力します。

Add DNS Resource Record

Record name *

dns

Record Type

CNAME

Hostname *

dns.example.com.

* Required field

Add

Add and Add Another

Add and Edit

Cancel

図32.10 新規 DNS リソースレコードの定義

4. **Add** をクリックして新規レコードを保存します。

コマンドラインでの DNS リソースレコードの追加

コマンドラインから DNS リソースレコードを追加するには、**ipa dnsrecord-add** コマンドを以下の構文で使

```
$ ipa dnsrecord-add zone_name record_name --record_type_option=data
```

zone_name は、レコードを追加する DNS ゾーンの名前です。*record_name* は、新規 DNS リソースレコードの識別子です。

表32.2「一般的な **ipa dnsrecord-add** オプション」では、A (IPv4)、AAAA (IPv6)、SRV、および PTR という一般的なリソースレコードのタイプのオプションを示しています。エントリーのリストは、同一コマンドでオプションを複数回使用して設定するか、Bash では **--option={val1,val2,val3}** のように中括弧内にオプションをコンマ区切りの一覧で記載します。

ipa dnsrecord-add の使用方法および IdM がサポートする DNS レコードタイプについての詳細情報は、**ipa dnsrecord-add --help** コマンドを実行してください。

表32.2 一般的な ipa dnsrecord-add オプション

全般的なレコードのオプション	
オプション	説明
--ttl=number	レコードの有効期間を設定します。
--structured	raw DNS レコードを解析し、それらを構造化された形式で返します。

"A" レコードのオプション

オプション	説明
--a-rec=ARECORD	A レコードのリストを渡します。
--a-ip-address=string	レコードの IP アドレスを渡します。

"AAAA" レコードのオプション

オプション	説明
--aaaa-rec=AAAAARECORD	AAAA (IPv6) レコードのリストを渡します。
--aaaa-ip-address=string	レコードの IPv6 アドレスを渡します。

"PTR" レコードのオプション

オプション	説明
--ptr-rec=PTRRECORD	PTR レコードのリストを渡します。
--ptr-hostname=string	レコードのホスト名を渡します。

"SRV" レコードのオプション

オプション	説明
--srv-rec=SRVRECORD	SRV レコードのリストを渡します。
--srv-priority=number	レコードの優先順位を設定します。あるサービスタイプに複数の SRV レコードがある場合もあります。優先順位 (0 - 65535) はレコードの階級を設定し、数字が小さいほど優先順位が高くなります。サービスは、優先順位の最も高いレコードを最初に使用する必要があります。
--srv-weight=number	レコードの加重を設定します。これは、SRV レコードの優先順位が同じ場合に順序を判断する際に役立ちます。設定された加重は最大 100 とし、これは特定のレコードが使用される可能性をパーセンテージで示しています。
--srv-port=number	ターゲットホスト上のサービスのポートを渡します。

"SRV" レコードのオプション**--srv-target=string**

ターゲットホストのドメイン名を提供します。該当サービスがドメイン内で利用可能でない場合は、単一のピリオド (.) にすることもできます。

32.4.5. コマンドラインからの DNS リソースレコードの追加および修正**例32.3 IPv4 レコードの追加**

以下の例ではレコード **www.example.com** が IP アドレス **192.0.2.123** で作成されます。

```
$ ipa dnsrecord-add example.com www --a-rec 192.0.2.123
```

例32.4 IPv4 ワイルドカードレコードの追加

以下の例ではワイルドカード A レコードが IP アドレス **192.0.2.123** で作成されます。

```
$ ipa dnsrecord-add example.com "*" --a-rec 192.0.2.123
```

例32.5 IPv4 レコードの修正

レコード作成時に A レコードの値を指定するオプションは **--a-record** です。ただし、A レコード修正時には **--a-record** オプションはそのレコードの現在の値を指定します。新しい値は **--a-ip-address** オプションで設定します。

```
$ ipa dnsrecord-mod example.com www --a-rec 192.0.2.123 --a-ip-address 192.0.2.1
```

例32.6 IPv6 レコードの追加

以下の例ではレコード **www.example.com** が IP アドレス **2001:db8::1231:5675** で作成されます。

```
$ ipa dnsrecord-add example.com www --aaaa-rec 2001:db8::1231:5675
```

例32.7 SRV レコードの追加

以下の例では、**_ldap._tcp** が SRV レコードのサービスタイプと接続プロトコルを定義します。**-srv-rec** オプションで優先順位、加重、ポート、およびターゲットの値を定義します。

例を示します。

```
[root@server ~]# ipa dnsrecord-add server.example.com _ldap._tcp --srv-rec="0 51 389 server1.example.com."
[root@server ~]# ipa dnsrecord-add server.example.com _ldap._tcp --srv-
```

```
rec="1 49 389 server2.example.com."
```

加重の値 (この例では **51** と **49**) では最大 100 が追加され、これは特定のレコードが使用される可能性をパーセンテージで示しています。

例32.8 PTR レコード

逆引き DNS レコードを追加する際は、他の DNS レコードの追加時に使用されるものと比べると、**ipa dnsrecord-add** コマンドで使用するゾーン名も逆になります。

```
$ ipa dnsrecord-add reverseNetworkIpAddress hostIpAddress --ptr-rec FQDN
```

通常、*hostIpAddress* は指定されたネットワークの IP アドレスの最後のオクテットになります。

たとえば、以下では、IPv4 アドレス 192.0.2.4 にある **server4.example.com** の PTR レコードが追加されます。

```
$ ipa dnsrecord-add 2.0.192.in-addr.arpa 4 --ptr-rec  
server4.example.com.
```

さらに以下の例では、逆 DNS エントリーが IP アドレス **2001:DB8::1111** のホスト **server2.example.com** の **0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa**. IPv6 逆引きゾーンに追加されます。

```
$ ipa dnsrecord-add 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.  
1.1.1.0.0.0.0.0.0.0.0.0.0.0.0 --ptr-rec server2.example.com.
```

32.4.6. DNS ゾーンからレコードを削除する

Web UI によるレコード削除

リソースレコードから特定のレコードタイプのみを削除するには、以下の手順に従います。

1. 「[Web UI でのゾーン設定編集](#)」にあるように DNS ゾーンページを開きます。
2. **DNS Resource Records** セクションで、リソースレコードの名前をクリックします。

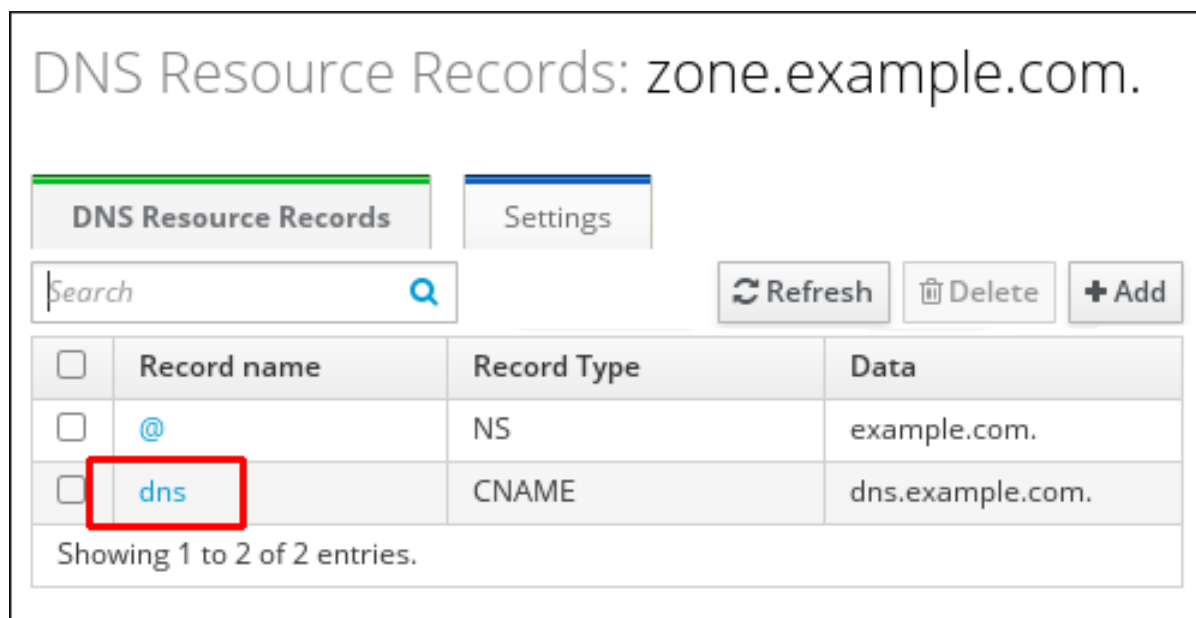


図32.11 DNS リソースレコードの選択

3. 削除するレコードタイプの名前にチェックを入れます。

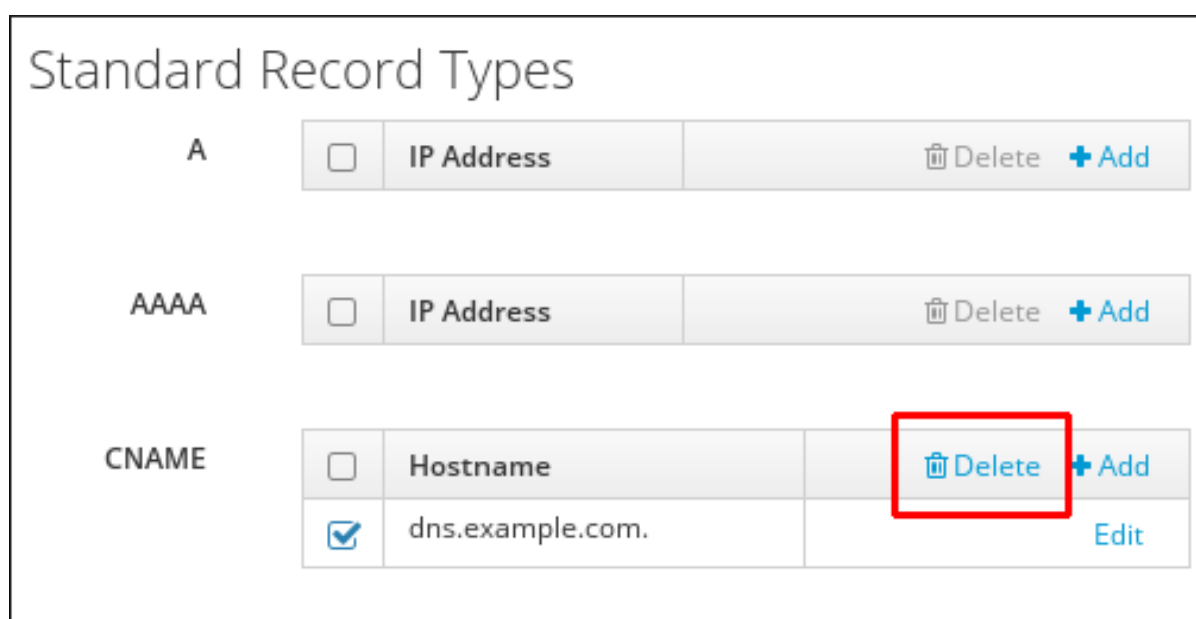


図32.12 DNS リソースレコードの削除

これで選択したレコードタイプのみが削除され、他の設定は有効なままになります。

ゾーン内のリソースの全レコードを削除するには、以下の手順に従います。

1. 「Web UI でのゾーン設定編集」にあるように DNS ゾーンページを開きます。
2. **DNS Resource Records** セクションで、削除するリソースレコードの名前の横にあるチェックボックスを選択し、**Delete** をクリックします。

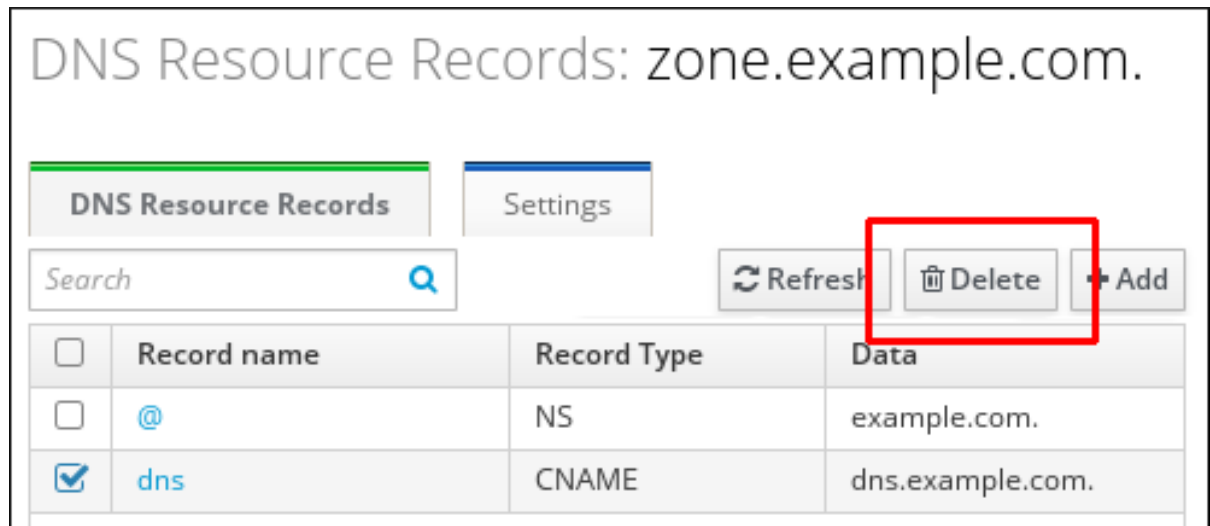


図32.13 リソースレコード全体の削除

これでリソースレコード全体が削除されます。

コマンドラインからのレコード削除

ゾーンからレコードを削除するには、**ipa dnsrecord-del** コマンドに **--recordType-rec** オプションとレコードの値を渡します。

以下の例では、A タイプのレコードが削除されます。

```
$ ipa dnsrecord-del example.com www --a-rec 192.0.2.1
```

オプションなしで **ipa dnsrecord-del** を実行すると、削除するレコードについての情報が要求されます。このコマンドを **--del-all** オプションと使用すると、ゾーンの関連する全レコードが削除されることに注意してください。

ipa dnsrecord-del コマンドの使用方法および使用可能なオプションの完全一覧については、**ipa dnsrecord-del --help** コマンドを実行してください。

32.4.7. ゾーンの有効化および無効化

IdM では、管理者が DNS ゾーンを有効、無効にすることができます。「[Master DNS ゾーンの削除](#)」にある方法で DNS ゾーンを削除するとゾーンエントリーと関連する設定すべてが完全に削除されますが、ゾーンを無効にすると IdM からゾーンを削除することなくアクティビティから除外することができます。無効にしたゾーンは後で有効にすることが可能です。

Web UI でのゾーンの有効化および無効化

Web UI で DNS ゾーンを管理するには、**Network Services** タブから **DNS** サブタブを開き、**DNS Zones** セクションを選択します。

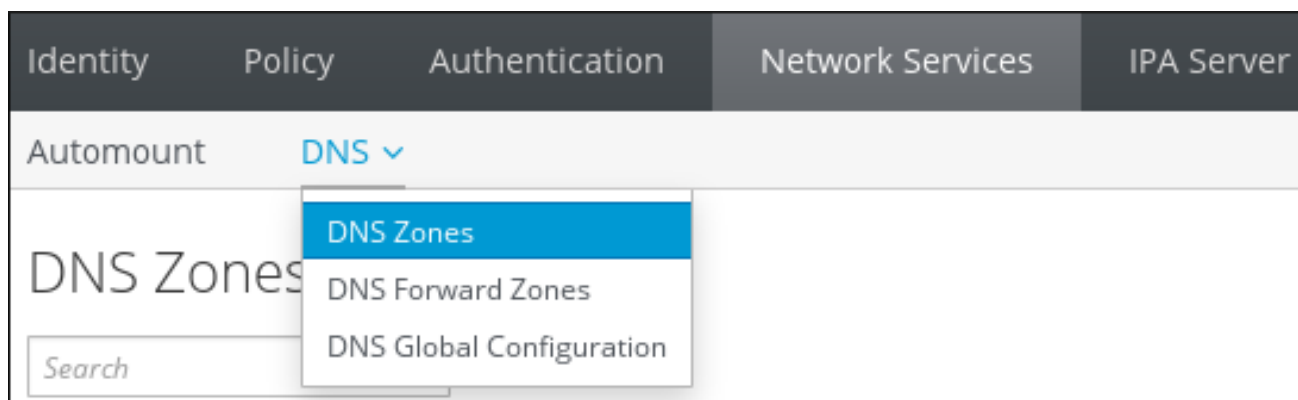


図32.14 DNS ゾーン管理

ゾーンを無効にするには、ゾーン名の横にチェックを入れて **Disable** をクリックします。



図32.15 DNS ゾーンの無効化

同様に、無効にしたゾーンを有効にするには、ゾーン名の横にチェックを入れて **Enable** をクリックします。

コマンドラインからの DNS ゾーンの無効化および有効化

コマンドラインから DNS ゾーンを無効にするには、以下のように **ipa dnszone-disable** コマンドを実行します。

```
[user@server ~]$ ipa dnszone-disable zone.example.com
```

```
-----
Disabled DNS zone "example.com"
-----
```

無効にしたゾーンを再度有効にするには、**ipa dnszone-enable** コマンドを実行します。

32.5. 動的 DNS 更新の管理

32.5.1. 動的 DNS 更新の有効化

IdM の新規 DNS ゾーンにおける動的 DNS 更新は、デフォルトでは有効になっていません。動的更新が許可されていないと、**ipa-client-install** スクリプトは新規クライアントをポイントする DNS レコードを追加することができません。



注記

動的更新を有効にすると、セキュリティーリスクを引き起こす可能性があります。ご使用の環境で動的更新が受け入れ可能な場合は、これによりクライアントのインストールが容易になります。

動的更新を有効にするには、以下が必要になります。

- DNS ゾーンで動的更新を許可する設定にする
- ローカルクライアントが動的更新を送信するように設定する

32.5.1.1. DNS ゾーンで動的更新を許可する設定

Web UI での動的 DNS 更新の有効化

1. **Network Services** タブから **DNS** サブタブを開き、**DNS Zones** セクションを選択します。

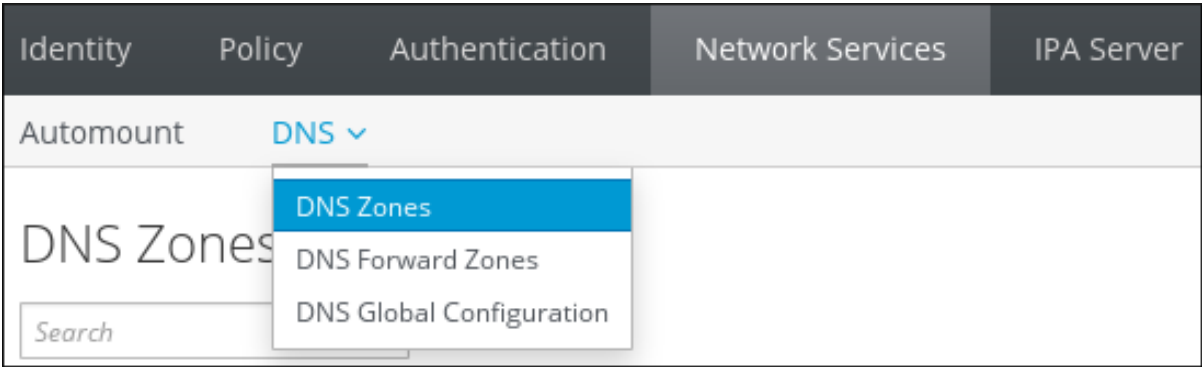


図32.16 DNS ゾーンの管理

2. ゾーンの全一覧からゾーン名をクリックして DNS ゾーンページを開きます。

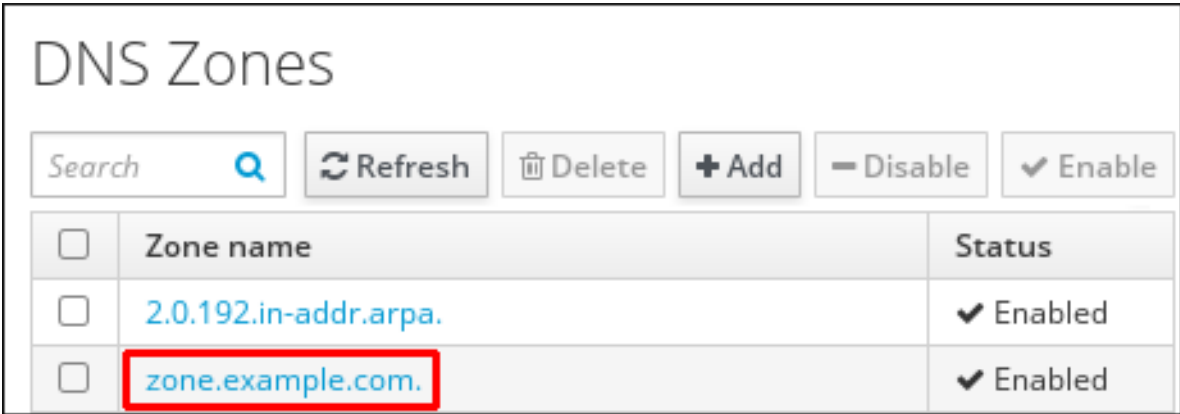
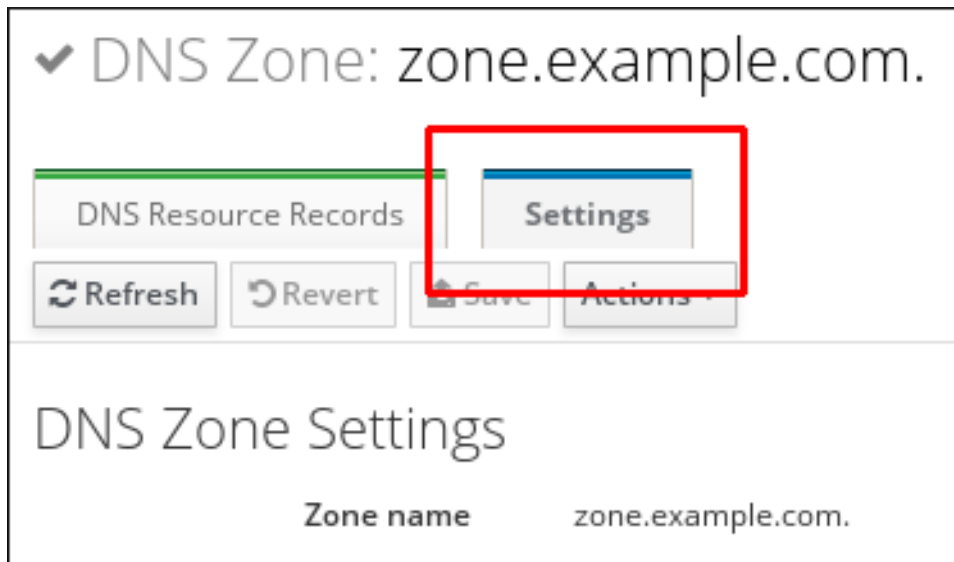


図32.17 マスターゾーンの編集

3. **Settings** をクリックして DNS ゾーン設定タブに切り替えます。

図32.18 マスターゾーン編集ページの **Settings** タブ

4. **Dynamic update** フィールドまでスクロールして、値を **True** に設定します。

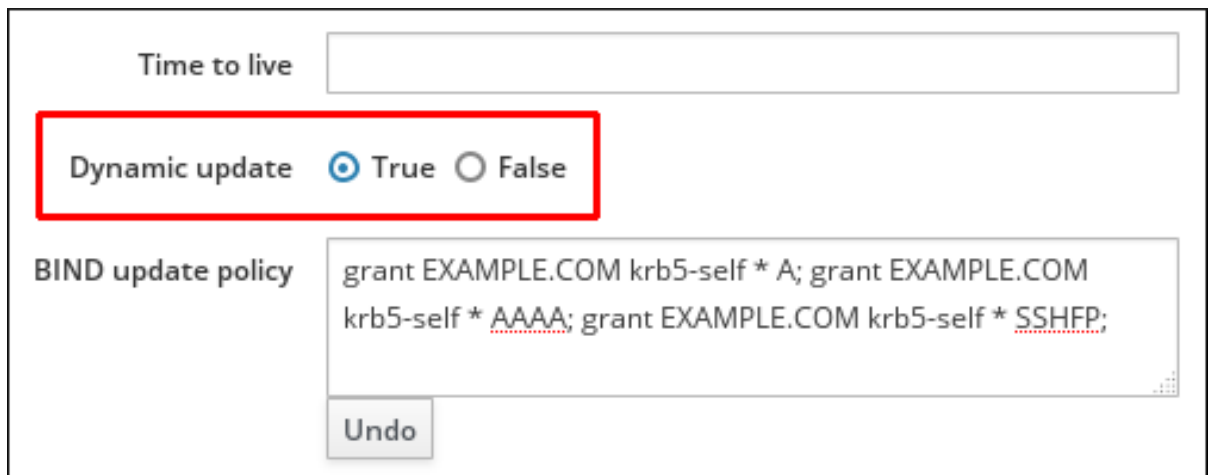


図32.19 動的 DNS 更新の有効化

5. ページ上部にある **Save** をクリックして新規設定を保存します。

コマンドラインからの動的 DNS 更新の有効化

コマンドラインから DNS ゾーンの動的更新を有効にするには、以下のように **ipa dnszone-mod** コマンドを **--dynamic-update=TRUE** オプションと使用します。

```
[user@server ~]$ ipa dnszone-mod server.example.com --dynamic-update=TRUE
```

32.5.1.2. クライアントが動的更新を送信する設定

ipa-client-install スクリプトで **--enable-dns-updates** オプションを使用すると、クライアントがドメインに登録される際に、クライアントが DNS 更新を送信するよう自動的に設定されます。

```
[root@client ~]# ipa-client-install --enable-dns-updates
```

DNS ゾーンでは、SOA 設定内におけるレコードの有効期間 (TTL) の値が設定されています。ただし、動的更新の TTL は、ローカルシステム上で System Security Service Daemon (SSSD) によって管理されます。動的更新の TTL 値を変更するには、SSSD ファイルを編集して値を設定します。デフォルト

ト値は 1200 秒です。

1. SSSD 設定ファイルを開きます。

```
[root@server ~]# vim /etc/sss/sss.conf
```

2. IdM ドメインのドメインセクションを見つけます。

```
[domain/ipa.example.com]
```

3. クライアントの動的更新が有効になっていない場合は、**dyndns_update** の値を true にします。

```
dyndns_updates = true
```

4. **dyndns_ttl** パラメーターの値を秒単位で追加または編集します。

```
dyndns_ttl = 2400
```

32.5.2. A/AAAA と PTR レコードの同期

逆引きゾーンでは、A と AAAA レコードは、PTR レコードとは別に設定されています。これらのレコードは別個なため、A/AAAA レコードに対応する PTR レコードがなかったり、その逆の場合もあり得ます。

PTR 同期が機能するには、以下の DNS 設定が必要になります。

- 正引きおよび逆引きゾーンの両方が IdM サーバーで管理されていること。
 - 両方のゾーンで動的更新が有効になっていること。
- 動的更新の有効化については、「[動的 DNS 更新の有効化](#)」で説明されています。
- PTR 同期が逆引きゾーンではなく、マスターの正引きゾーンで有効になっていること。
 - PTR レコードは、要求しているクライアント名が PTR レコード内の名前と一致する場合にのみ、更新されます。



重要

IdM の Web UI やコマンドラインツールによる変更、または LDAP エントリーを直接編集して変更した場合、PTR レコードは更新されません。DNS サービス自体による変更の場合にのみ、PTR レコードは同期されます。



警告

クライアントシステムは、自身の IP アドレスを更新できます。つまり、危険にさらされたクライアントを使って IP アドレスを変更すると、PTR レコードの上書きが可能になります。

Web UI による PTR レコード同期の設定

PTR レコードの同期は、PTR レコードがある逆引き DNS ゾーンではなく、A または AAAA レコードが保存されているゾーンで設定する必要があることに注意してください。

1. **Network Services** タブから **DNS** サブタブを開き、**DNS Zones** セクションを選択します。

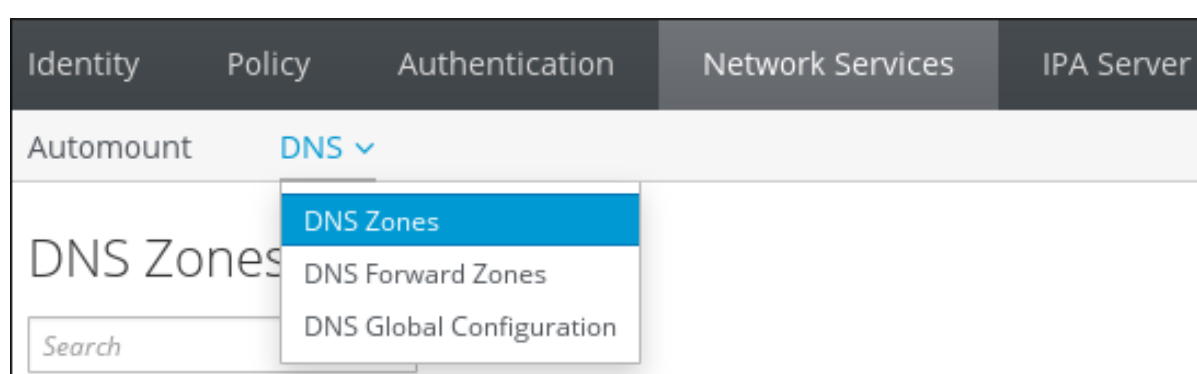


図32.20 DNS ゾーンの管理

2. ゾーンの全一覧からゾーン名をクリックして DNS ゾーンページを開きます。

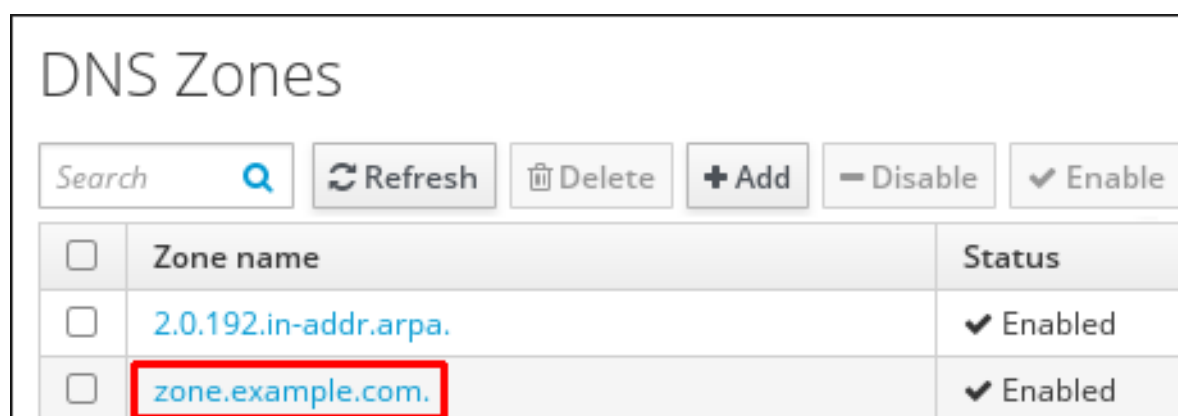
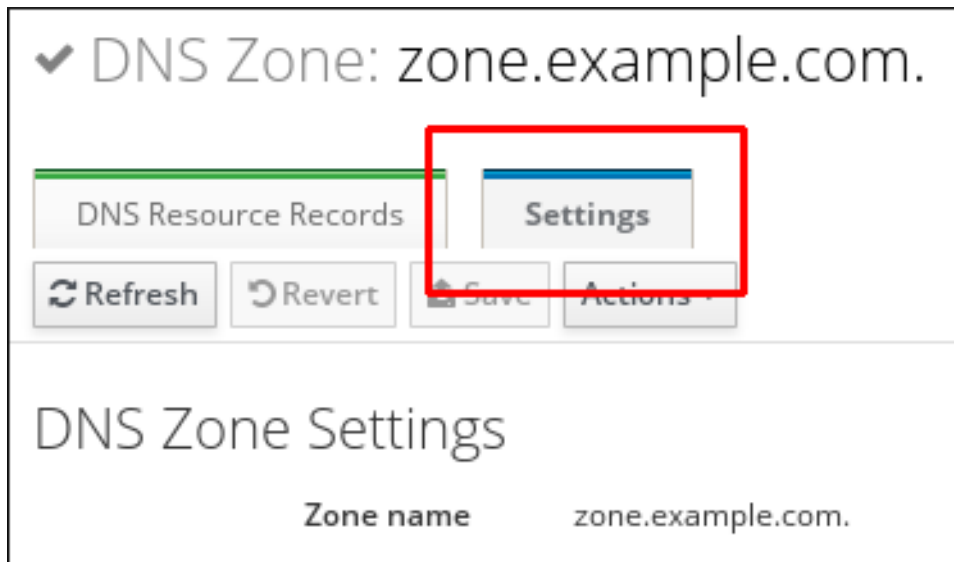


図32.21 DNS ゾーンの編集

3. **Settings** をクリックして DNS ゾーン設定タブに切り替えます。

図32.22 マスターゾーン編集ページの **Settings** タブ

4. **Allow PTR sync** チェックボックスを選択します。

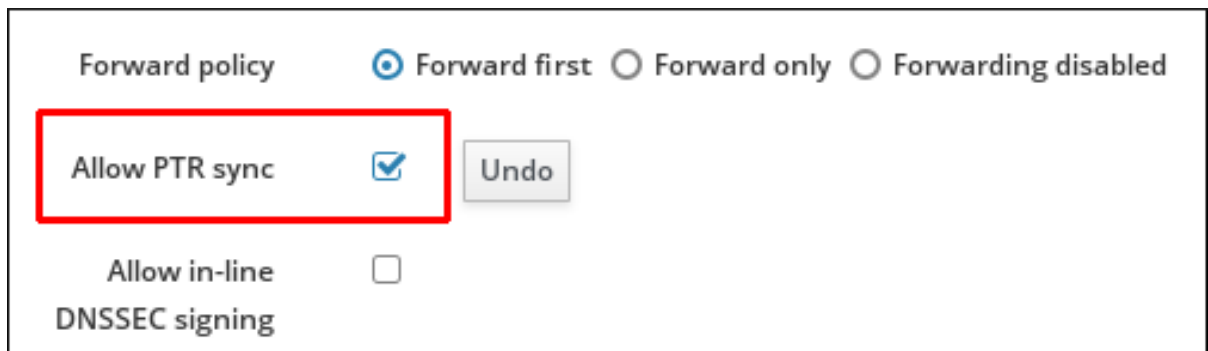


図32.23 PTR 同期の有効化

5. ページ上部にある **Save** をクリックして新規設定を保存します。

コマンドラインからの PTR レコード同期の設定

PTR レコードの同期は、PTR レコードがある逆引き DNS ゾーンではなく、A または AAAA レコードが保存されているゾーンで設定する必要があることに注意してください。

DNS ゾーンで正引きおよび逆引きエントリが自動で同期するように設定するには、ゾーンの作成時または編集時に **--allow-sync-ptr** オプションを **1** に設定します。たとえば、以下のように **ipa dnszone-mod** コマンドを使って既存のゾーンを編集します。

```
[user@server ~]$ ipa dnszone-mod --allow-sync-ptr=1 zone.example.com
```

--allow-sync-ptr のデフォルト値は **0** で、この場合は同期が無効になります。

32.5.3. DNS 動的更新ポリシーの更新

IdM サーバーが管理している DNS ドメインは、RFC 3007^[4] に準拠した DNS の動的更新に対応します。

特定のクライアントがどのレコードを修正可能かを判定するルールは、**/etc/named.conf** ファイル内の **update-policy** ステートメントと同じ構文になります。動的更新ポリシーについての詳細は、[BIND 9 documentation](#) を参照してください。

DNS ゾーンにおける DNS の動的更新が無効になっていると、すべての DNS 更新は動的更新ポリシーステートメントを反映せずに拒否されることに注意してください。動的 DNS 更新についての詳細は、「[動的 DNS 更新の有効化](#)」を参照してください。

Web UI による DNS 更新ポリシーの更新

1. **Network Services** タブから **DNS** サブタブを開き、**DNS Zones** セクションを選択します。

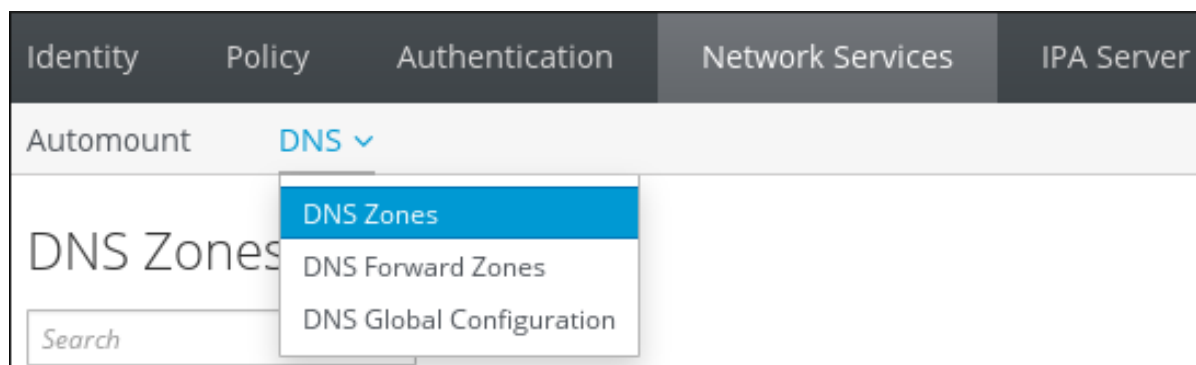


図32.24 DNS ゾーンの管理

2. ゾーンの全一覧からゾーン名をクリックして DNS ゾーンページを開きます。

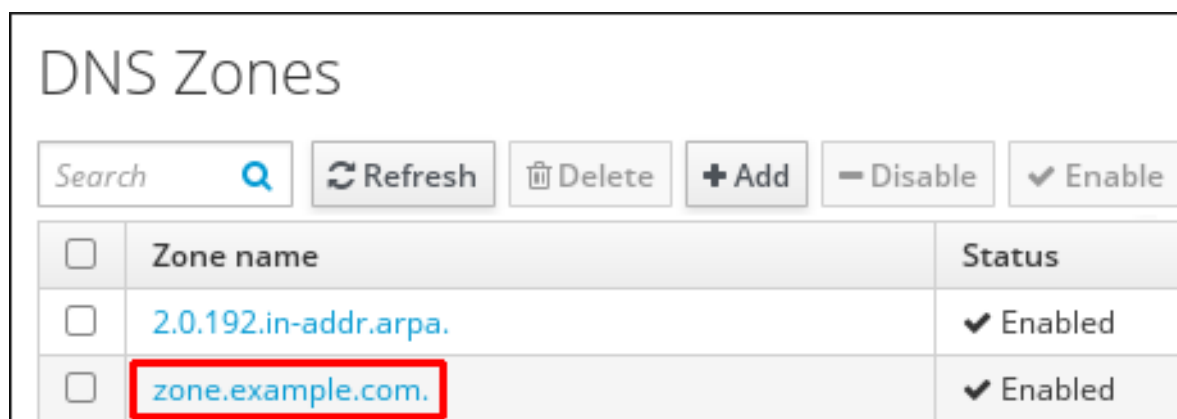
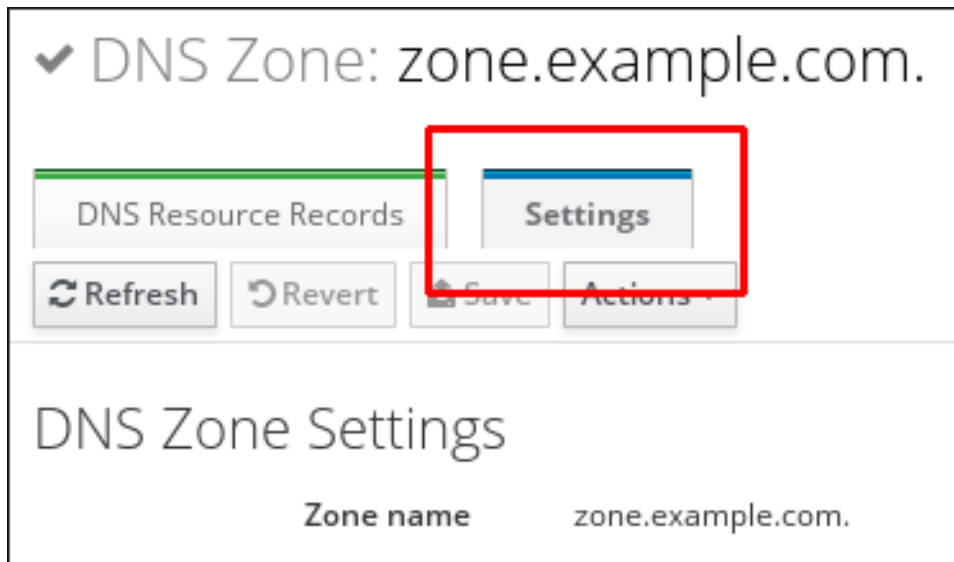


図32.25 DNS ゾーンの編集

3. **Settings** をクリックして DNS ゾーン設定タブに切り替えます。

図32.26 マスターゾーン編集ページの **Settings** タブ

4. **BIND update policy** テキストボックス内のセミコロン区切りのリストで、必要な更新ポリシーを設定します。

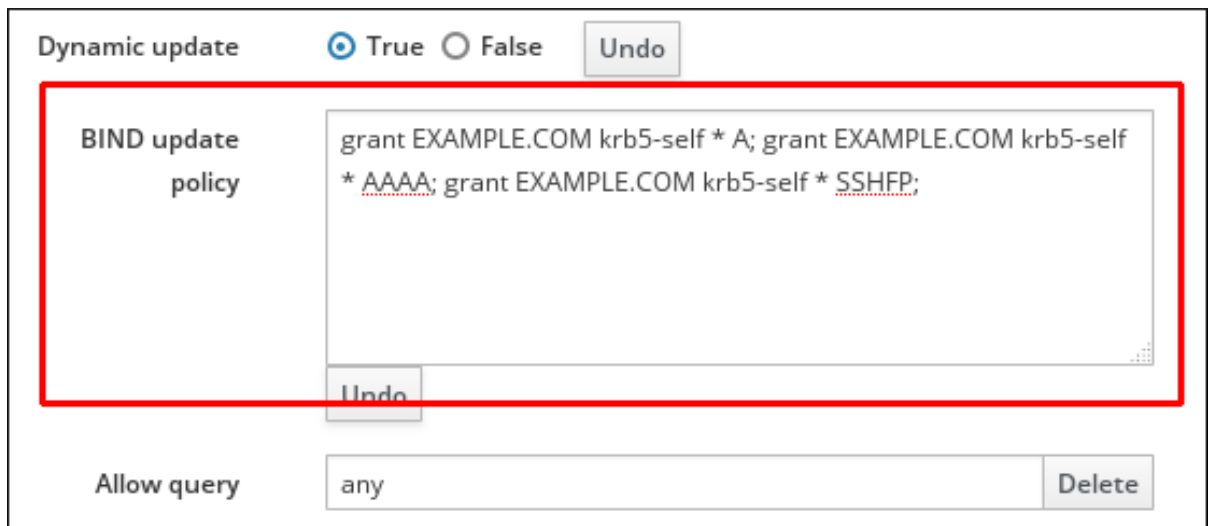


図32.27 DNS 更新ポリシーの設定

5. DNS ゾーンページ上部にある **Save** をクリックして新規設定を保存します。

コマンドラインからの **DNS 更新ポリシーの更新**

コマンドラインから DNS 更新ポリシーを設定するには、**--update-policy** オプションを使用して、その後のステートメントにアクセス制御ルールを追加します。例を示します。

```
$ ipa dnszone-mod zone.example.com --update-policy "grant EXAMPLE.COM
krb5-self * A; grant EXAMPLE.COM krb5-self * AAAA; grant EXAMPLE.COM
krb5-self * SSHFP;"
```

32.6. DNS 転送の管理

DNS 転送は、DNS クエリーの返答方法に影響を与えます。デフォルトでは、IdM と統合されている BIND サービスは、権威 DNS サーバーおよび再帰 DNS サーバーの両方として機能するように設定されます。

IdM サーバーが権威サーバーとなっている DNS ゾーンに属する名前のクエリーを DNS クライアントが行うと、BIND は設定済みゾーンに含まれているデータで応答します。権威データは常に他のデータよりも優先されます。

IdM サーバーが権威を持っていない名前のクエリーを DNS クライアントが行うと、BIND は他の DNS サーバーを使ってこのクエリーを解決しようとします。フォワーダーが定義されていない場合は、BIND はインターネット上の root サーバーに質問し、再帰解決アルゴリズムを使用して DNS クエリーに答えます。

以下のような場合には、BIND が他の DNS サーバーに直接連絡してインターネット上で利用可能なデータを元に再帰を実行することは望ましくありません。

- DNS サーバーが異なるクライアントに異なる応答をする *Split DNS* 設定または *DNS views* 設定となっている場合。Split DNS 設定は、いくつかの DNS 名が企業ネットワーク内では利用可能になっているものの、外部からは利用できないという環境でよく使われています。
- インターネット上の DNS へのアクセスをファイアウォールが制限している設定。
- DNS レベルでのフィルタリングやロギングが集中化されている設定。
- ローカル DNS キャッシュへの転送でネットワークトラフィックの最適化を図っている設定。

これらの設定では、BIND は公開インターネット上で完全な再帰を使用しません。代わりに、フォワーダーと呼ばれる別の DNS サーバーを使用してクエリーを解決します。BIND がフォワーダーを使用するように設定されている場合、クエリーと応答は IdM サーバーとフォワーダー間で行き来し、IdM サーバーが権威のないデータの DNS キャッシュとして機能します。

転送ポリシー

IdM は *first* および *only* の標準 BIND 転送ポリシーのほかに、IdM 固有の *none* 転送ポリシーをサポートしています。

Forward first (デフォルト)

DNS クエリーは設定済みフォワーダーに転送されます。サーバーエラーやタイムアウトでクエリーが失敗すると、BIND はインターネット上のサーバーを使用する再帰解決にフォールバックします。forward first はデフォルトのポリシーで、トラフィックを最適化します。

Forward only

DNS クエリーは設定済みフォワーダーに転送されます。サーバーエラーやタイムアウトでクエリーが失敗すると、BIND はクライアントにエラーを返します。forward only ポリシーは、split DNS 環境で推奨されます。

None: 転送の無効化

DNS クエリーは転送されません。転送の無効化は、グローバルの転送設定でゾーン固有の上書きとしてのみ、役に立ちます。このオプションは、BIND 設定でフォワーダーの空のリストを指定する IdM と同等のものです。

転送は IdM と他の DNS サーバーからのデータを結合しない

転送では IdM 内のデータと他の DNS サーバーからのデータは結合できません。BIND サービスは、IdM サーバーの権限があるゾーンにクエリーされた DNS 名が属している場合は、クエリーを別のサーバーに転送しません。このため、IdM が管理するゾーンに存在しない名前をクライアントがクエリーすると、転送は使用されません。

例32.9 シナリオ例

IdM サーバーは **test.example.** DNS ゾーンに対して権威があります。BIND は、**192.0.2.254** IP アドレスの DNS サーバーにクエリーを転送するように設定されています。

クライアントが **nonexistent.test.example.** DNS 名のクエリーを送信すると、BIND は IdM サーバーが **test.example.** ゾーンに対して権威があることを検出し、**192.0.2.254.** サーバーにクエリーを転送しません。このため、DNS クライアントは **NXDomain** の応答を受信し、クエリーされたドメインが存在しないことをユーザーに知らせます。

32.6.1. グローバルフォワーダーの設定

グローバルフォワーダーは「[DNS 転送の管理](#)」にあるように、IdM サーバーの権威がない全 DNS クエリーの解決に使用される DNS サーバーです。

管理者は、以下の 2 つの方法でグローバル転送の IP アドレスと転送ポリシーを設定することができます。

IdM Web UI での ipa dnsconfig-mod コマンドの使用

これらのネイティブ IdM ツールを使用した設定は、全 IdM DNS サーバーに即座に適用されます。「[DNS 設定の優先順位](#)」の説明にあるように、グローバル DNS の設定は、**/etc/named.conf** ファイルで定義されたローカル設定よりも優先度が高くなります。

/etc/named.conf ファイルの編集

各 IdM DNS 上の **/etc/named.conf** を手動で編集すると、サーバーごとに異なるグローバルフォワーダーとポリシーを使用できるようになります。**/etc/named.conf** の変更後に BIND サービスを再起動する必要があることに注意してください。

Web UI でのフォワーダーの設定

IdM Web UI で DNS グローバル設定を定義するには、以下の手順に従います。

1. **Network Services** タブをクリックして **DNS** サブタブを開き、**DNS Global Configuration** セクションを選択します。
2. 新規のグローバルフォワーダーを追加するには、**Add** をクリックして IP アドレスを入力します。新規の転送ポリシーを定義するには、利用可能なポリシー一覧から選択します。

図32.28 Web UI でのグローバル DNS 設定の編集

3. **Save** をクリックして新規設定を保存します。

コマンドラインからのフォワーダー設定

コマンドラインからフォワーダーのグローバルリストを設定するには、**ipa dnsconfig-mod** コマンドを使用します。これは、LDAP データを編集することで DNS グローバル設定を編集します。**ipa dnsconfig-mod** コマンドおよびそのオプションは、即座に全 IdM DNS サーバーに反映され、ローカル設定を上書きします。

グローバルフォワーダーのリストを編集するには、以下のように **ipa dnsconfig-mod** を使います。

```
[user@server ~]$ ipa dnsconfig-mod --forwarder=192.0.2.254
Global forwarders: 192.0.2.254
```

32.6.2. 正引きゾーンの設定

正引きゾーンには権威 データが含まれず、特定のゾーンに属する名前へのクエリーを指定されたフォワーダーに転送するようにネームサーバーに指示します。

重要

正引きゾーンは、絶対に必要な場合以外は使用しないでください。これは、グローバル転送の設定の上書きに限定してください。ほとんどの場合は、「[グローバルフォワダーの設定](#)」にあるように、**グローバル転送の設定のみで十分**で、正引きゾーンは必要ありません。

正引きゾーンは非標準の解決法で、これを使用すると予想外かつ問題のある動作につながる恐れがあります。新規 DNS ゾーン作成の際には、Red Hat では NS レコードを使用した標準 DNS 委任を常に使用し、正引きゾーンの使用は回避することを推奨しています。

サポートされる転送ポリシーについての情報は、「[転送ポリシー](#)」を参照してください。

BIND サービスについての詳細情報は、『[Red Hat Enterprise Linux ネットワークガイド](#)』、`/usr/share/doc/bind-version_number/` ディレクトリーにある BIND 9 Administrator Reference Manual、または外部ソース ^[5]。

Web UI での正引きゾーンの設定

Web UI で正引きゾーンを管理するには、**Network Services** タブから **DNS** サブタブを開き、**DNS Forward Zones** セクションを選択します。

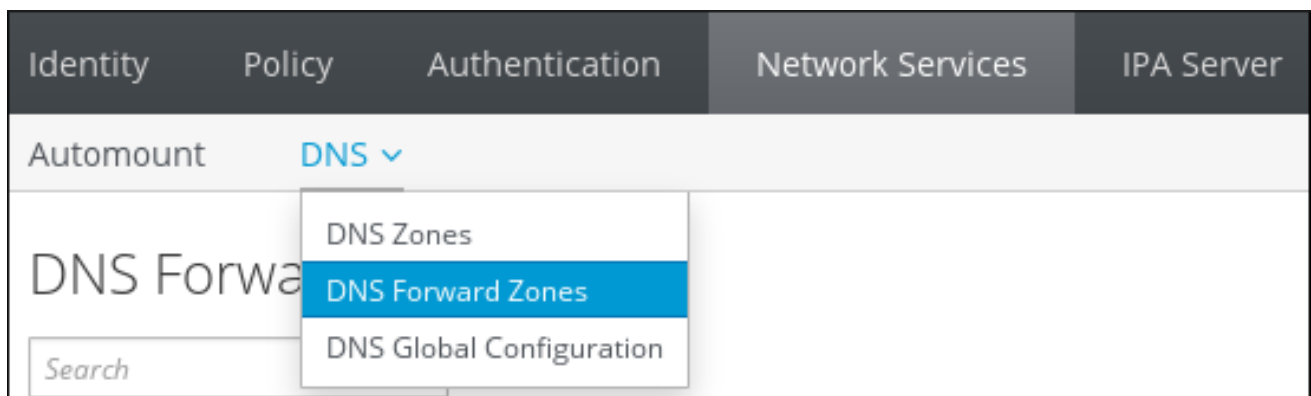


図32.29 DNS 正引きゾーンの管理

DNS 正引きゾーン のセクションでは、管理者は以下の正引きゾーンに関する全操作を処理できます。正引きゾーンの現行一覧の表示、新規正引きゾーンの追加、正引きゾーンの削除、正引きゾーンの表示、正引きゾーンのフォワーダーおよび転送ポリシーの修正、正引きゾーンの有効化および無効化。

コマンドラインからの正引きゾーンの設定

コマンドラインから正引きゾーンを管理するには、以下のように **ipa dnsforwardzone-*** コマンドを使用します。

注記

ipa dnsforwardzone-* コマンドは、マスターゾーンの管理に使用する **ipa dnszone-*** コマンドと同様の動作をします。

ipa dnsforwardzone-* コマンドは、**--forwarder**、**--forward-policy**、および **--name-from-ip** などのオプションを受け付けます。利用可能なオプションについての詳細情報は、[表 32.1 「ゾーン属性」](#) を参照するか、以下のようにコマンドに **--help** オプションを追加して実行してください。

```
ipa dnsforwardzone-add --help
```

正引きゾーンの追加

dnsforwardzone-add コマンドを使って新規の正引きゾーンを追加します。転送ポリシーが **none** に設定されている場合を除いて、少なくとも 1 つのフォワーダーを指定する必要があります。

```
[user@server ~]$ ipa dnsforwardzone-add zone.test. --
forwarder=172.16.0.1 --forwarder=172.16.0.2 --forward-policy=first

Zone name: zone.test.
Zone forwarders: 172.16.0.1, 172.16.0.2
Forward policy: first
```

正引きゾーンの修正

dnsforwardzone-mod コマンドを使って正引きゾーンを修正します。転送ポリシーが **none** に設定されている場合を除いて、少なくとも 1 つのフォワーダーを指定する必要があります。修正は以下のいずれかの方法で実行できます。

```
[user@server ~]$ ipa dnsforwardzone-mod zone.test. --
forwarder=172.16.0.3

Zone name: zone.test.
Zone forwarders: 172.16.0.3
Forward policy: first
```

```
[user@server ~]$ ipa dnsforwardzone-mod zone.test. --forward-policy=only

Zone name: zone.test.
Zone forwarders: 172.16.0.3
Forward policy: only
```

正引きゾーンの表示

dnsforwardzone-show コマンドを使用して指定した正引きゾーンの情報を表示します。

```
[user@server ~]$ ipa dnsforwardzone-show zone.test.

Zone name: zone.test.
Zone forwarders: 172.16.0.5
Forward policy: first
```

正引きゾーンの検索

dnsforwardzone-find コマンドを使用して指定した正引きゾーンを検索します。

```
[user@server ~]$ ipa dnsforwardzone-find zone.test.

Zone name: zone.test.
Zone forwarders: 172.16.0.3
Forward policy: first
-----
Number of entries returned 1
-----
```

正引きゾーンの削除

dnsforwardzone-del コマンドを使用して指定した正引きゾーンを削除します。

```
[user@server ~]$ ipa dnsforwardzone-del zone.test.
-----
Deleted forward DNS zone "zone.test."
-----
```

正引きゾーンの有効化および無効化

dnsforwardzone-enable と **dnsforwardzone-disable** のコマンドを使用して正引きゾーンを有効化および無効化します。正引きゾーンはデフォルトでは有効化されています。

```
[user@server ~]$ ipa dnsforwardzone-enable zone.test.
-----
Enabled forward DNS zone "zone.test."
-----
```

```
[user@server ~]$ ipa dnsforwardzone-disable zone.test.
-----
Disabled forward DNS zone "zone.test."
-----
```

パーミッションの追加および削除

dnsforwardzone-add-permission と **dnsforwardzone-remove-permission** のコマンドを使用してシステムのパーミッションを追加または削除します。

```
[user@server ~]$ ipa dnsforwardzone-add-permission zone.test.
-----
Added system permission "Manage DNS zone zone.test."
-----
Manage DNS zone zone.test.
```

```
[user@server ~]$ ipa dnsforwardzone-remove-permission zone.test.
-----
Removed system permission "Manage DNS zone zone.test."
-----
Manage DNS zone zone.test.
```

32.7. 逆引き DNS ゾーンの管理

逆引き DNS ゾーンは以下のいずれかの方法で確認できます。

- **reverse_ipv4_address.in-addr.arpa** または **reverse_ipv6_address.ip6.arpa** という形式のゾーン名。

逆引き IP アドレスは、IP アドレスの要素を反転させて作成します。たとえば、IPv4 ネットワークが **192.0.2.0/24** の場合、逆引きゾーン名は **2.0.192.in-addr.arpa.** になります (最後のピリオドを含む)。

- **network_ip_address/subnet_mask_bit_count** の形式でのネットワークアドレス。

ゾーンを IP ネットワークで作成するには、ネットワーク情報をサブネットマスクのビットカウントが付いた (正引きスタイルの) IP アドレスに設定します。ビットカウントは、IPv4 アドレスの場合は 8 の倍数、IPv6 アドレスの場合は 4 の倍数にします。

Web UI での逆引き DNS ゾーンの追加

1. **Network Services** タブから **DNS** サブタブを開き、**DNS Zones** セクションを選択します。

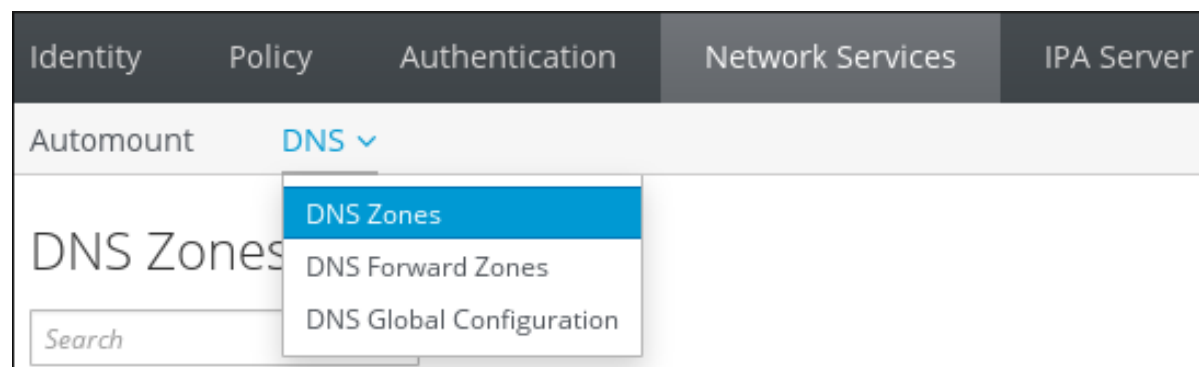


図32.30 DNS ゾーン管理

2. 全ゾーン一覧上部にある **Add** をクリックします。

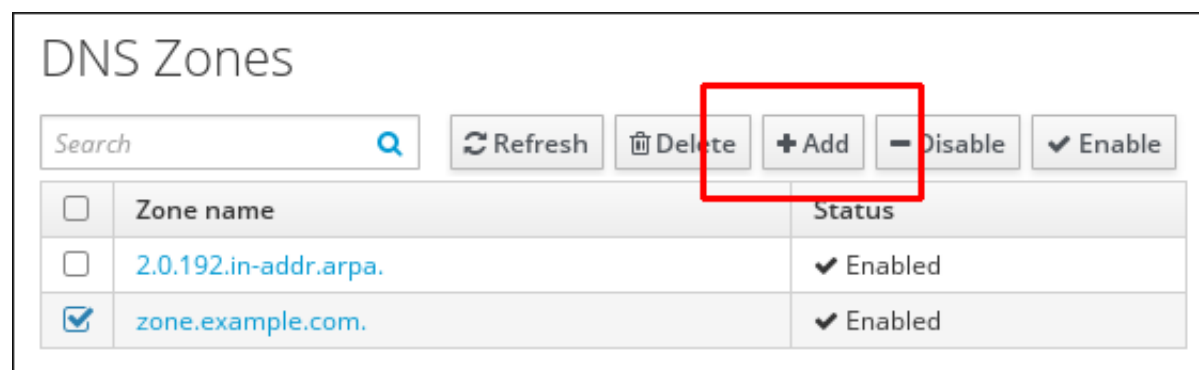


図32.31 逆引き DNS ゾーンの追加

3. ゾーン名または逆引きゾーン IP ネットワークを入力します。
 - a. たとえば、ゾーン名で逆引き DNS ゾーンを追加するには、以下のようにします。

図32.32 名前での逆引きゾーンの作成

- b. または、逆引きゾーン IP ネットワークで逆引き DNS ゾーンを追加するには、以下のようになります。

図32.33 IP ネットワークでの逆引きゾーンの作成

Reverse zone IP network フィールドの記入中には無効なネットワークアドレスについての警告が表示されますが、完全なネットワークアドレスが入力されるとこれは表示されなくなります。

4. **Add** をクリックして新規逆引きゾーンを保存します。

コマンドラインからの逆引き DNS ゾーンを追加

コマンドラインから逆引き DNS ゾーンを作成するには、**ipa dnszone-add** コマンドを実行します。

たとえば、ゾーン名で逆引きゾーンを作成するには、以下のようになります。

```
[user@server]$ ipa dnszone-add 2.0.192.in-addr.arpa.
```

または、IP ネットワークで逆引きゾーンを作成するには、以下のようになります。


```
[user@server ~]$ ipa dnszone-add --name-from-ip=192.0.2.0/24
```

逆引き DNS ゾーンの他の管理操作

「[Master DNS ゾーン管理](#)」では他のゾーン管理操作を説明しており、この中には DNS ゾーンの編集、無効化、または有効化など、逆引き DNS ゾーン管理に適用可能なものもあります。

32.8. DNS クエリーポリシーの定義

DNS ドメイン内でホスト名を解決するために、DNS クライアントは DNS ネームサーバーにクエリーを発行します。特定のセキュリティーコンテキストやパフォーマンスの面から、クライアントがゾーン内の DNS レコードにクエリーすることについては制限することが推奨されます。

DNS クエリーは、ゾーンの作成時または **--allow-query** オプションを使用した **ipa dnszone-mod** コマンドでクエリー発行が許可されるクライアントのリストを設定する際に、設定できます。

例を示します。

```
[user@server ~]$ ipa dnszone-mod --allow-  
query=192.0.2.0/24;2001:DB8::/32;203.0.113.1 example.com
```

--allow-query のデフォルト値は **any** で、この場合、ゾーンにはどのクライアントもクエリーを実行できます。

32.9. DNS の場所

32.9.1. DNS ベースのサービス検出

DNS ベースのサービス検出は、**LDAP** や **Kerberos** といった特定のサービスを提供するネットワーク内でサーバーの場所を特定するためにクライアントが DNS プロトコルを使用するプロセスです。よくあるタイプの操作では、一番近いネットワークインフラストラクチャー内でクライアントが認証サーバーを特定するというものがあります。この場合、より高いスループットが提供される一方でネットワーク遅延は短くなり、総コストも抑えられます。

サービス検出の主な利点は以下のとおりです。

- クライアントを近くのサーバー名で明示的に設定する必要がない。
- DNS サーバーがポリシーの集中プロバイダーとして使用される。同一の DNS サーバーを使用するクライアントは、サービスプロバイダーについての同じポリシーとその優先順位についてアクセスできるようになります。

IdM ドメイン内では、DNS サービスレコード (SRV レコード) は LDAP、Kerberos、および他のサービスに関して存在します。たとえば、以下のコマンドは、IdM DNS ドメイン内で TCP ベースの Kerberos サービスを提供するホストについて DNS サーバーにクエリーを行います。

例32.10 DNS の場所の個別の結果

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com  
0 100 88 idmsvr-01.idm.example.com.  
0 100 88 idmsvr-02.idm.example.com.
```

この出力には、以下の情報が含まれています。

- **0** (優先順位): ターゲットホストの優先順位。値が低いほど、優先順位が高くなります。
- **100** (加重): 優先順位が同じエントリーの相対的加重を指定します。詳細は [RFC 2782, section 3](#) を参照してください。
- **88** (ポート番号): サービスのポート番号。
- サービスを提供しているホストの正規名。

上記の例では、返された 2 つのホスト名で優先順位と加重が同じものになっています。この場合、クライアントはこの結果一覧からランダムでエントリーを使用します。

クライアントがある DNS の場所の中にある DNS サーバーにクエリーを行った場合は、出力が違ってきます。ある場所に割り当てられている IdM サーバーの場合は、定められた値が返されます。以下の例では、クライアントは **germany** という場所にある DNS サーバーにクエリーを行なっています。

例32.11 DNS の場所ベースの結果

```
$ dig -t SRV +short _kerberos._tcp.idm.example.com
_kerberos._tcp.germany._locations.idm.example.com.
0 100 88 idmserver-01.idm.example.com.
50 100 88 idmserver-02.idm.example.com.
```

IdM DNS サーバーは自動的に DNS エイリアス (CNAME) を返します。これは、ローカルサーバーを優先する、DNS の場所に固有の SRV レコードをポイントします。この CNAME レコードは、出力の一行目に表示されます。上記の例では、ホスト **idmserver-01.idm.example.com** の優先順位の値が最も低いので、これが優先されます。**idmserver-02.idm.example.com** の優先順位の値は高いので、優先ホストが利用できない場合にのみ、バックアップとして使用されます。

32.9.2. デプロイメントでの DNS の場所についての考慮事項

プライマリー IdM DNS ドメインに対して権威がある IdM DNS サーバーの場合、IdM は場所に固有の SRV レコードを生成することができます。各 IdM DNS サーバーが場所に固有の SRV レコードを生成するため、各 DNS の場所で少なくとも 1 つの IdM DNS サーバーをインストールする必要があります。

クライアントの DNS の場所へのアフィニティーは、クライアントが受け取る DNS レコードでのみ定義されます。このため、DNS サービス検出を行なっているクライアントが IdM DNS サーバーからの場所固有のレコードを解決する場合、IdM DNS サーバーと非 IdM DNS スレーブサーバーおよび **recursor** を結合することができます。

IdM と非 IdM DNS サーバーが混ざっているほとんどのデプロイメントでは、DNS **recursor** は往復時間のメトリクスを使って、自動的に一番近い IdM DNS サーバーを選択します。通常、これによって非 IdM DNS サーバーを使用しているクライアントが一番近い DNS の場所のレコードを取得し、最適な IdM サーバーを使用することになります。

32.9.2.1. DNS 有効期間 (TTL)

クライアントは、ゾーン構成で設定された期間、DNS リソースレコードをキャッシュすることができます。このキャッシュのために、有効期間 (TTL) が過ぎるまで、クライアントが変更を受け取れない可能性があります。IdM の TTL のデフォルト値は **1 day** です。

クライアントのコンピューターがサイト間でローミングする場合は、ご自分の IdM DNS ゾーン向けに

TTL の値を調節してください。クライアントがサイト間でのローミングに必要な時間よりも低い値に設定します。こうすることで、クライアントが別のサイトに再接続して場所固有の SRV レコードをリフレッシュするように DNS サーバーにクエリーする前に、クライアント上でキャッシュされた DNS エントリーの有効期間が切れるようになります。

DNS ゾーンの TTL のデフォルト値を修正する方法については、「[マスター DNS ゾーンの追加設定](#)」を参照してください。

32.9.3. DNS の場所の作成

Web UI での DNS の場所の作成

1. **IPA Server** タブを開いてから、**Topology** サブタブを選択します。
2. ナビゲーションバーで **IPA Locations** をクリックします。
3. 場所一覧上部にある **Add** をクリックします。
4. 場所の名前を入力します。
5. **Add** をクリックして場所を保存します。

各場所ごとに上記のステップを繰り返します。

コマンドラインからの DNS の場所の作成

たとえば、新しい場所である **germany** を作成するには、以下を実行します。

```
[root@server ~]# ipa location-add germany
-----
Added IPA location "germany"
-----
Location name: germany
```

各場所ごとに上記のコマンドを繰り返します。

32.9.4. DNS の場所への IdM サーバーの割り当て

Web UI から IdM サーバーを DNS の場所に割り当てる手順

1. **IPA Server** タブを開いてから、**Topology** サブタブを選択します。
2. ナビゲーションバーで **IPA Servers** をクリックします。
3. IdM サーバー名をクリックします。
4. DNS の場所を選択し、オプションでサービスの加重を設定します。

IPA Server: idmserver-01.idm.example.com

Refresh Revert Save

Server name	idmserver-01.idm.example.com.
Min domain level	0
Max domain level	1
Managed suffixes	domain ca
Location	germany
Service weight	100

図32.34 DNS の場所へのサーバーの割り当て

5. **Save** をクリックします。
6. 上記のステップで DNS の場所を割り当てたホスト上で **named-pkcs11** サービスを再起動します。

```
[root@idmserver-01 ~]# systemctl restart named-pkcs11
```

DNS の場所を割り当てる IdM サーバーで上記のステップを繰り返します。

コマンドラインから IdM サーバーを DNS の場所に割り当てる手順

1. オプション: 設定済み DNS の全場所を一覧表示します。

```
[root@server ~]# ipa location-find
-----
2 IPA locations matched
-----
Location name: australia
Location name: germany
-----
Number of entries returned: 2
-----
```

2. サーバーを DNS の場所に割り当てます。たとえば、**germany** にサーバー *idmserver-01.idm.example.com* を割り当てるには、以下を実行します。

```
[root@server ~]# ipa server-mod idmserver-01.idm.example.com --
location=germany
ipa: WARNING: Service named-pkcs11.service requires restart on IPA
server
idmserver-01.idm.example.com to apply configuration changes.
-----
Modified IPA server "idmserver-01.idm.example.com"
-----
```

```
Servername: idmserver-01.idm.example.com
Min domain level: 0
Max domain level: 1
Location: germany
Enabled server roles: DNS server, NTP server
```

3. 上記のステップで DNS の場所を割り当てたホスト上で **named-pkcs11** サービスを再起動します。

```
[root@idmserver-01 ~]# systemctl restart named-pkcs11
```

DNS の場所を割り当てる IdM サーバーで上記のステップを繰り返します。

32.10. 外部 DNS の使用時に DNS レコードを組織的に更新する手順

外部の DNS を使用する場合は、Identity Management はトポロジで変更がなされても、自動的に DNS レコードが更新されることはありません。以下では、外部 DNS で管理する DNS レコードを組織的に更新する方法について説明します。こうすることで、手動による DNS 更新の必要性が低減されます。

基本的な概要については、「[Identity Management での外部 DNS の更新](#)」を参照してください。

手順と例については、以下を参照してください。

- GUI で外部 DNS のレコードを管理する場合は、「[GUI: 外部 DNS レコードの更新](#)」
- **nsupdate** ユーティリティーを使用して外部 DNS のレコードを管理する場合は、「[コマンドライン: nsupdate を使用した外部 DNS レコード更新](#)」

32.10.1. Identity Management での外部 DNS の更新

DNS レコードを更新すると、古い DNS レコードまたは無効な DNS レコードが削除され、新しいレコードが追加されます。

トポロジで変更が合った場合は、DNS レコードを更新する必要があります。たとえば、以下のような場合です。

- レプリカをインストールまたはアンインストールした後
- Identity Management サーバーで CA、DNS、KRA、または Active Directory の信頼をインストールした後

32.10.2. GUI: 外部 DNS レコードの更新

1. 更新する必要があるレコードを表示します。**ipa dns-update-system-records --dry-run** コマンドを使用します。

```
$ ipa dns-update-system-records --dry-run
IPA DNS records:
_kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88
ipa.example.com.
_kerberos-master._udp.example.com. 86400 IN SRV 0 100 88
ipa.example.com.
[... output truncated ...]
```

2. 外部の DNS GUI を使用してレコードを更新します。

32.10.3. コマンドライン: **nsupdate** を使用した外部 **DNS** レコード更新

nsupdate 向けに **DNS** レコードのあるファイルを生成

1. **ipa dns-update-system-records --dry-run** コマンドに **--out** オプションを付けて実行します。このオプションは、生成するファイルのパスを指定します。

```
$ ipa dns-update-system-records --dry-run --out
dns_records_file.nsupdate
IPA DNS records:
  _kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88
ipa.example.com.
  _kerberos-master._udp.example.com. 86400 IN SRV 0 100 88
ipa.example.com.
[... output truncated ...]
```

生成されたファイルには、**nsupdate** ユーティリティーで使用可能な形式の必須の DNS レコードが含まれます。

2. 生成されるレコードは以下に依存します。
 - 。レコードが更新されるゾーンの自動検出
 - 。そのゾーンの権威サーバーの自動検出

通常とは異なる DNS 設定を使用してる場合、またはゾーン委任がない場合は、**nsupdate** は適切なゾーンやサーバーを検出できないことがあります。その場合は、生成されるファイルの最初に以下のオプションを追加してください。

- 。 **server** で、**nsupdate** がレコードを送信する権威 DNS サーバーのサーバー名もしくはポートを指定します。
- 。 **zone** で、**nsupdate** がレコードを見つけるゾーンのゾーン名を指定します。

例:

```
$ cat dns_records_file.nsupdate
zone example.com.
server 192.0.2.1
; IPA DNS records
update delete _kerberos-master._tcp.example.com. SRV
update add _kerberos-master._tcp.example.com. 86400 IN SRV 0 100 88
ipa.example.com.
[... output truncated ...]
```

ネームサーバーへの動的 **DNS** 更新リクエストの送信

nsupdate を使用してリクエストを送信する際には、以下のメカニズムを使用してリクエストのセキュリティを確保してください。

Transaction Signature (TSIG) プロトコル

TSIG を使用すると、**nsupdate** で共有キーを利用することができます。詳細は [手順32.1「TSIG を使用した **nsupdate** リクエストの送信](#)」を参照してください。

GSS algorithm for TSIG (GSS-TSIG)

GSS-TSIG では、GSS-API インターフェイスを使用して秘密 TSIG キーを取得します。GSS-TSIG は TSIG プロトコルの拡張機能です。詳細は [手順32.2「GSS-TSIG を使用した nsupdate リクエストの送信」](#) を参照してください。

手順32.1 TSIG を使用した nsupdate リクエストの送信

1. 以下の前提条件を満たしていることを確認します。
 - 。お使いの DNS サーバーが TSIG 向けに設定されている必要があります。サーバー設定の例については、[BIND](#)、[PowerDNS](#)、[Knot DNS 1 + Knot DNS 2 + Knot DNS 3](#) を参照してください。
 - 。DNS サーバーとそのクライアントの両方に共有キーがある必要があります。
2. **nsupdate** を実行し、以下のいずれかのオプションで共有秘密を提供します。
 - 。 **-k** では、TSIG 認証キーを提供します。

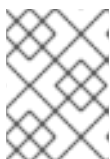
```
$ nsupdate -k tsig_key.file dns_records_file.nsupdate
```

- 。 **-y** では、キーの名前と Base64 でエンコードされた共有秘密から署名を生成します。

```
$ nsupdate -y algorithm:keyname:secret dns_records_file.nsupdate
```

手順32.2 GSS-TSIG を使用した nsupdate リクエストの送信

1. 以下の前提条件を満たしていることを確認します。
 - 。お使いの DNS サーバーが GSS-TSIG 向けに設定されている必要があります。サーバー設定の例については、[BIND](#)、[PowerDNS](#)、[Windows DNS](#) を参照してください。



注記

この手順では、Kerberos V5 プロトコルが GSS-API のテクノロジーとして使用されることを前提としています。

2. DNS 更新リクエストを送信するには、レコードの更新が可能なプリンシパルと認証を行い、**nsupdate** に **-g** オプションを追加して実行し、GSS-TSIG モードを有効にします。

```
$ kinit principal_allowed_to_update_records@REALM
$ nsupdate -g dns_records_file.nsupdate
```

その他のリソース

- 。 **nsupdate(8)** man ページ
- 。 TSIG プロトコルについては、[RFC 2845](#)
- 。 GSS-TSIG アルゴリズムについては、[RFC 3645](#)

32.11. 既存のサーバーへの DNS サービスのインストール

当初は DNS サービスをインストールしなかった IdM サーバーにも、後から DNS サービスをインストールすることができます。これを実行するには、ipa-server-dns パッケージがインストールされていることを確認してから、**ipa-dns-install** ユーティリティーを使用します。

ipa-dns-install を使用して DNS サービスを設定する方法は、**ipa-server-install** を使用して DNS をインストールする方法とほぼ同じです。「[統合 DNS のあるサーバーのインストール](#)」を参照してください。

ipa-dns-install についての詳細は、ipa-dns-install(1) man ページを参照してください。

32.11.1. ネームサーバーの追加設定

IdM は、新たに設定された IdM DNS サーバーを、**/etc/resolv.conf** ファイルの DNS サーバー一覧に追加します。IdM サーバーが利用できなくなった時のために、バックアップサーバーとして他の DNS サーバーを手動で追加することが推奨されます。例を示します。

```
search example.com

; the IdM server
nameserver 192.0.2.1

; backup DNS servers
nameserver 198.51.100.1
nameserver 198.51.100.2
```

/etc/resolv.conf の設定に関する詳細は、resolv.conf(5) man ページを参照してください。

[3] GSS-TSIG についての詳細は、[RFC 3545](#) を参照。

[4] RFC 3007 の全内容については <http://tools.ietf.org/html/rfc3007> を参照してください。

[5] 詳細は [BIND 9 Configuration Reference](#) を参照してください。

第33章 AUTOMOUNT の使用

Automount は、複数のシステムにまたがってディレクトリーを管理、組織、アクセスするものです。ディレクトリーへのアクセスがリクエストされると、Automount はディレクトリーを自動的にマウントします。これは、ドメイン内のクライアント上におけるディレクトリー共有を容易にするので、IdM ドメイン内で非常にうまく機能します。これはユーザーホームディレクトリーで特に重要となります。「[ユーザーホームディレクトリーの設定](#)」を参照してください。

IdM では、automount は内部 LDAP ディレクトリーおよび (設定されている場合は) DNS サービスと機能します。

33.1. AUTOMOUNT と IDM

Automount は、ディレクトリーを分かりやすく組織化する方法を提供します。各ディレクトリーはマウントポイントはキーと呼ばれます。複数のキーをグループ化したものがマップで、マップはそれらの物理的位置または概念上の場所にしたがって関連付けられます。

automount のベース設定ファイルは、`/etc` ディレクトリー内の **auto.master** ファイルです。必要な場合は、複数の **auto.master** 設定ファイルを別々の場所にあるサーバーに配置することができます。

autofs をサーバー上で設定し、そのサーバーが IdM ドメイン内のクライアントである場合、automount の全設定情報は IdM ディレクトリーに保存されます。マップや場所、キーといった **autofs** の設定は、別々のテキストファイルではなく、LDAP エントリーとして保存されます。たとえば、デフォルトのマップファイルである **auto.master** は、以下のように保存されます。

```
dn: automountmapname=auto.master,cn=default,cn=automount,dc=example,dc=com
objectClass: automountMap
objectClass: top
automountMapName: auto.master
```



重要

Identity Management は既存の **autofs** デプロイメントとは機能しますが、**autofs** 自体を設定するわけではありません。

新たな場所は、**cn=automount,dc=example,dc=com** の下にコンテナエントリーとして追加され、マップとキーはそれぞれその場所の下に保存されます。

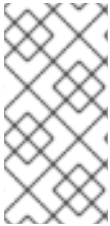
他の IdM ドメインサービスと同様に、automount は IdM とネイティブに機能します。automount の設定は、IdM ツールで管理できます。

- **Locations** には、**ipa automountlocation*** コマンドを使用します。
- 直接または間接の **maps** には、**ipa automountmap*** コマンドを使用します。
- **Keys** には、**ipa automountkey*** コマンドを使用します。

automount が IdM ドメイン内で機能するには、NFS サーバーが IdM クライアントとして設定されている必要があります。NFS 自体の設定は、[Red Hat Enterprise Linux ストレージ管理ガイド](#) で説明しています。

33.2. AUTOMOUNT の設定

Identity Management では、場所やマップといった automount エントリーの設定には、既存の autofs または NFS サーバーが必要になります。automount エントリーを作成しても、基礎となる **autofs** 設定は作成されません。**Autofs** は、LDAP または SSSD をデータストアとして使用して手動で設定するか、自動で設定することが可能です。



注記

automount の設定を変更する前に、少なくとも 1 人のユーザーが **/home** ディレクトリーをコマンドラインから正常にマウントできるかテストしてください。NFS が正常に機能していることを確認すると、後で IdM automount 設定エラーが発生しても解決が容易になります。

33.2.1. NFS の自動設定

システムを IdM クライアントとして設定したら (ドメインクライアントとして設定された IdM サーバーとレプリカを含む)、**autofs** は IdM ドメインを NFS ドメインとして使用するよう設定し、**autofs** サービスを有効にすることができます。

デフォルトでは、**ipa-client-automount** ユーティリティーは、**/etc/sysconfig/nfs** および **/etc/idmapd.conf** という NFS 設定ファイルを自動設定します。また、SSSD が NFS の認証情報を管理するようにも設定します。**ipa-client-automount** コマンドをオプションなしで実行すると、DNS 検索スキャンが実行されて利用可能な IdM サーバーを特定し、**default** という名前のデフォルトの場所を作成します。

```
[root@ipa-server ~]# ipa-client-automount
Searching for IPA server...
IPA server: DNS discovery
Location: default
Continue to configure the system with these values? [no]: yes
Configured /etc/nsswitch.conf
Configured /etc/sysconfig/nfs
Configured /etc/idmapd.conf
Started rpcidmapd
Started rpcgssd
Restarting sssd, waiting for it to become available.
Started autofs
```

IdM サーバーがデフォルト以外の automount の場所を使用、作成することも可能です。

```
[root@server ~]# ipa-client-automount --server=ipaserver.example.com --
location=boston
```

NFS の設定に加え、**ipa-client-automount** ユーティリティーは外部の IdM ストアがアクセス不能になった場合に備えて、SSSD が automount マップをキャッシュするよう設定します。SSSD の設定では、以下の 2 つが実行されます。

- サービスの設定情報が SSSD 設定に追加されます。IdM ドメインエントリーには、autofs プロバイダーとマウントの場所の設定があります。

```
autofs_provider = ipa
ipa_automount_location = default
```

NFS がサポート対象サービスのリスト (**services = nss,pam,autofs...**) に追加され、空白の設定エントリー (**[autofs]**) が提供されます。

- Name Service Switch (NSS) サービス情報が更新され、automount 情報についてまず SSSD がチェックされ、次にローカルファイルがチェックされます。

```
automount: sss files
```

クライアントによる automount マップのキャッシュが適切でないといった、非常に安全性の高い環境のインスタンスでは、**ipa-client-automount** コマンドを **--no-sssd** オプションと実行することが可能です。その場合、必須の NFS 設定ファイルはすべて変更されますが、SSSD 設定は変更されません。

```
[root@server ~]# ipa-client-automount --no-sssd
```

--no-sssd を使用しないと、**ipa-client-automount** で更新される設定ファイルは違ったものになります。

- **/etc/sysconfig/nfs** ではなく **/etc/sysconfig/autofs** が更新されます。
- **/etc/autofs_ldap_auth.conf** を IdM LDAP 設定で設定します。
- **/etc/nsswitch.conf** が automount マップに LDAP サービスを使用するように設定されます。



注記

ipa-client-automount コマンドは 1 回しか実行できません。設定にエラーがある場合は、設定ファイルを手動で編集する必要があります。

33.2.2. autofs が SSSD と Identity Management を使用するように手動で設定する手順

1. **/etc/sysconfig/autofs** ファイルを編集し、autofs が検索するスキーマ属性を指定します。

```
#
# Other common LDAP naming
#
MAP_OBJECT_CLASS="automountMap"
ENTRY_OBJECT_CLASS="automount"
MAP_ATTRIBUTE="automountMapName"
ENTRY_ATTRIBUTE="automountKey"
VALUE_ATTRIBUTE="automountInformation"
```

2. LDAP 設定を指定します。これには 2 通りの方法があります。最も簡単な方法は、automount サービスが LDAP サーバーのその場所を自分で発見するようにすることです。

```
LDAP_URI="ldap:///dc=example,dc=com"
```

別の方法では、使用する LDAP サーバーと LDAP 検索のベース DN を明示的に設定します。

```
LDAP_URI="ldap://ipa.example.com"
SEARCH_BASE="cn=location,cn=automount,dc=example,dc=com"
```



注記

location のデフォルト値は **default** です。新たな場所が追加されると (「[場所の設定](#)」)、クライアントがその場所を使用するように向けることができます。

3. `/etc/autofs_ldap_auth.conf` ファイルを編集して、autofs が IdM LDAP サーバーを使ったクライアント認識を許可するようにします。

- **authrequired** を **yes** に変更します。
- プリンシパルを NFS クライアントサーバー用 Kerberos ホストプリンシパル **host/fqdn@REALM** に設定します。プリンシパル名は、GSS クライアント認証の一部として IdM ディレクトリーへの接続に使用されます。

```
<autofs_ldap_sasl_conf
    usetls="no"
    tlsrequired="no"
    authrequired="yes"
    authtype="GSSAPI"
    clientprinc="host/server.example.com@EXAMPLE.COM"
/>
```

必要な場合は、**klist -k** を実行して正確なホストプリンシパル情報を取得します。

4. autofs を、SSSD が管理するサービスとして設定します。

1. SSSD 設定ファイルを開きます。

```
[root@server ~]# vim /etc/sss/sss.conf
```

2. autofs サービスを、SSSD が処理するサービス一覧に追加します。

```
[sss]
services = nss,pam,autofs
```

3. 新規の **[autofs]** セクションを作成します。これは空白のままにしても構いません。autofs サービスのデフォルト設定は、ほとんどのインフラストラクチャーで機能します。

```
[nss]

[pam]

[sudo]

[autofs]

[ssh]

[pac]
```

4. オプションとして、autofs エントリーの検索ベースを設定します。デフォルトではこれは LDAP 検索ベースですが、**ldap_autofs_search_base** パラメーターでサブツリーを指定することもできます。

```
[domain/EXAMPLE]
...
ldap_search_base = "dc=example,dc=com"
ldap_autofs_search_base = "ou=automount,dc=example,dc=com"
```

5. SSSD を再起動します。

```
[root@server ~]# systemctl restart sssd.service
```

6. **/etc/nsswitch.conf** ファイルで、SSSD が automount 設定のソースに記載されていることを確認します。

```
automount: sss files
```

7. autofs を再起動します。

```
[root@server ~]# systemctl restart autofs.service
```

8. ユーザーの **/home** ディレクトリを一覧表示して、設定をテストします。

```
[root@server ~]# ls /home/userName
```

これでリモートのファイルシステムがマウントされない場合は、**/var/log/messages** ファイルでエラーをチェックします。必要な場合は、**/etc/sysconfig/autofs** ファイルで **LOGGING** パラメーターを **debug** に設定してデバッグレベルを高めます。

注記

automount で問題がある場合は、IdM インスタンスの 389 Directory Server アクセスログで automount 試行を相互参照します。ここでは、試行されたアクセス、ユーザー、および検索ベースが表示されます。

またシンプルな方法では、automount をフォアグラウンドで実行し、デバッグのログを記録します。

```
automount -f -d
```

これで LDAP のアクセスログと automount のログを相互参照することなく、デバッグのログ情報が直接出力されます。

33.2.3. Solaris での Automount の設定

注記

Solaris では、Identity Management で使用されるスキーマとは別のスキーマを autofs 設定に使用します。Identity Management では 2307bis-style automount スキーマが使用され、これは 389 Directory Server で定義されています (また、IdM の内部 Directory Server インスタンスでも使用されています)。

1. NFS サーバーが Red Hat Enterprise Linux 上で稼働している場合、Solaris マシン上で NFSv3 が最大のサポートバージョンであることを指定します。**/etc/default/nfs** ファイルで以下のパラメーターを設定します。

```
NFS_CLIENT_VERSMAX=3
```

2. **ldapclient** コマンドを使って、ホストが LDAP を使用するよう設定します。

```
ldapclient -v manual -a authenticationMethod=none
-a defaultSearchBase=dc=example,dc=com
-a defaultServerList=ipa.example.com
-a
serviceSearchDescriptor=passwd:cn=users,cn=accounts,dc=example,dc=com
-a
serviceSearchDescriptor=group:cn=groups,cn=compat,dc=example,dc=com
-a
serviceSearchDescriptor=auto_master:automountMapName=auto.master,cn=location,cn=automount,dc=example,dc=com?one
-a
serviceSearchDescriptor=auto_home:automountMapName=auto_home,cn=location,cn=automount,dc=example,dc=com?one
-a objectClassMap=shadow:shadowAccount=posixAccount
-a searchTimelimit=15
-a bindTimeLimit=5
```

3. **automount** を有効にします。

```
# svcadm enable svc:/system/filesystem/autofs
```

4. 設定をテストします。

1. LDAP 設定を確認します。

```
# ldapclient -l auto_master

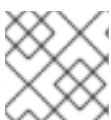
dn:
automountkey=/home,automountmapname=auto.master,cn=location,cn=automount,dc=example,dc=com
objectClass: automount
objectClass: top
automountKey: /home
automountInformation: auto.home
```

2. ユーザーの **/home** ディレクトリーを一覧表示します。

```
# ls /home/userName
```

33.3. KERBEROS 対応の NFS サーバーの設定

Identity Management を使って Kerberos 対応の NFS サーバーを設定することができます。



注記

NFS サーバーは Red Hat Enterprise Linux 上で稼働する必要はありません。

33.3.1. Kerberos 対応の NFS サーバーの設定

1. IdM ツールを実行する前に Kerberos チケットを取得します。

```
[jsmith@server ~]$ kinit admin
```

2. NFS ホストマシンが IdM ドメインにクライアントとして追加されていない場合は、ホストエントリーを作成します。[「ホストエントリーの追加」](#) を参照してください。
3. IdM ドメインで NFS サービスエントリーを作成します。例を示します。

```
[jsmith@server ~]$ ipa service-add nfs/nfs-server.example.com
```

詳細は、[「サービスエントリーおよび Keytab の追加と編集」](#) を参照してください。

4. **ipa-getkeytab** コマンドで NFS サーバー用の NFS サービス keytab を作成し、キーをホスト keytab に直接保存します。例を示します。

```
[jsmith@server ~]$ ipa-getkeytab -s ipaserver.example.com -p  
nfs/nfs-server.example.com -k /etc/krb5.keytab
```



注記

サービスエントリーをチェックして、NFS サービスが IdM で適切に設定されていることを keytab で確認します。

```
[jsmith@server ~]$ ipa service-show nfs/nfs-  
server.example.com  
Principal: NFS/nfs-server.example.com@EXAMPLE.COM  
Keytab: True
```



注記

この手順では、**ipa-getkeytab** が実行可能な Red Hat Enterprise Linux または UNIX システム上で、NFS サーバーが稼働していることを想定しています。

NFS サーバーが **ipa-getkeytab** を実行できないシステム上で稼働している場合は、システムツールを使って keytab を作成します。これには、以下の 2 つを実行する必要があります。

- 。キーは **/root** (またはそれに相当する) ディレクトリー内で作成する必要があります。
- 。 **ktutil** コマンドは、キーをシステムの **/etc/krb5.keytab** ファイルにマージすることができます。このツールの使用方法は、[ktutil man page](#) で説明されています。

5. NFS パッケージをインストールします。

```
[root@nfs-server ~]# yum install nfs-utils
```

6. weak crypto のサポートを設定します。これは、ドメイン内の **いずれかの** クライアント (Red Hat Enterprise Linux 5 クライアントのような) が DES といった古い暗号化オプションを使用する場合に、すべての NFS クライアントで必要になります。

1. **krb5.conf** ファイルを編集して、weak crypto を許可します。

```
[root@nfs-server ~]# vim /etc/krb5.conf

allow_weak_crypto = true
```

2. IdM サーバーの Kerberos 設定を更新し、DES 暗号化タイプに対応させます。

```
[jsmith@ipaserver ~]$ ldapmodify -x -D "cn=directory manager" -w
password -h ipaserver.example.com -p 389

dn: cn=EXAMPLEREALM,cn=kerberos,dc=example,dc=com
changetype: modify
add: krbSupportedEncSaltTypes
krbSupportedEncSaltTypes: des-cbc-crc:normal
-
add: krbSupportedEncSaltTypes
krbSupportedEncSaltTypes: des-cbc-crc:special
-
add: krbDefaultEncSaltTypes
krbDefaultEncSaltTypes: des-cbc-crc:special
```

7. **ipa-client-automount** コマンドを実行して NFS 設定を設定します。

デフォルトでは、**/etc/sysconfig/nfs** ファイル内でセキュアな NFS が有効になり、IdM DNS ドメインを **/etc/idmapd.conf** ファイル内の **Domain** パラメーターに設定します。

8. **/etc/exports** ファイルを編集して、Kerberos 情報を追加します。

```
/export *(rw,sec=krb5:krb5i:krb5p)
```

9. NFS サーバーと関連サービスを再起動します。

```
[root@nfs-server ~]# systemctl restart nfs.service
[root@nfs-server ~]# systemctl restart nfs-server.service
[root@nfs-server ~]# systemctl restart nfs-secure.service
[root@nfs-server ~]# systemctl restart nfs-secure-server.service
```

10. 「[Kerberos 対応の NFS クライアントの設定](#)」の説明にしたがって、NFS サーバーを NFS クライアントとして設定します。

33.3.2. Kerberos 対応の NFS クライアントの設定

1. IdM ツールを実行する前に Kerberos チケットを取得します。

```
[jsmith@server ~]$ kinit admin
```

2. NFS クライアントが IdM ドメインにクライアントとして登録されていない場合は、[「ホストエントリーの追加」](#)にあるように必要なホストエントリーを作成します。

3. **ipa-client-automount** コマンドを実行して NFS 設定を設定します。

デフォルトでは、**/etc/sysconfig/nfs** ファイル内でセキュアな NFS が有効になり、IdM DNS ドメインを **/etc/idmapd.conf** ファイル内の **Domain** パラメーターに設定します。

4. GSS デーモンを起動します。

```
[root@nfs-client-server ~]# systemctl start rpc-gssd.service
[root@nfs-client-server ~]# systemctl start rpcbind.service
[root@nfs-client-server ~]# systemctl start nfs-idmapd.service
```

5. ディレクトリーをマウントします。

```
[root@nfs-client-server ~]# echo "$NFSSERVER:/this /mnt/this nfs4
sec=krb5i,rw,proto=tcp,port=2049" >>/etc/fstab
[root@nfs-client-server ~]# mount -av
```

6. クライアントシステム上の SSSD がホームディレクトリーを管理し、Kerberos チケットを更新するように設定します。

1. **--enablemkhomedir** オプションで SSSD を有効にします。

```
[root@nfs-client-server ~]# authconfig --update --enablesssd --
enablesssdauth --enablemkhomedir
```

2. OpenSSH クライアントを再起動します。

```
[root@nfs-client-server ~]# systemctl restart sssh.service
```

3. SSSD 設定ファイルの IdM ドメインセクションを編集し、keytab 更新オプションを設定します。

```
[root@nfs-client-server ~]# vim /etc/sss/sss.conf

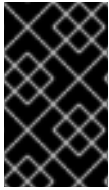
[domain/EXAMPLE.COM]
cache_credentials = True
krb5_store_password_if_offline = True
ipa_domain = example.com
id_provider = ipa
auth_provider = ipa
...
krb5_renewable_lifetime = 50d
krb5_renew_interval = 3600
```

4. SSSD を再起動します。

```
[root@nfs-client-server ~]# systemctl restart sssd.service
```

33.4. 場所の設定

場所はマップのセットで、すべて **auto.master** に保存されます。また、場所には複数のマップを保存できます。場所のエントリは、マップエントリーのコンテナとしてのみ機能します。それ自体は、automount 設定ではありません。

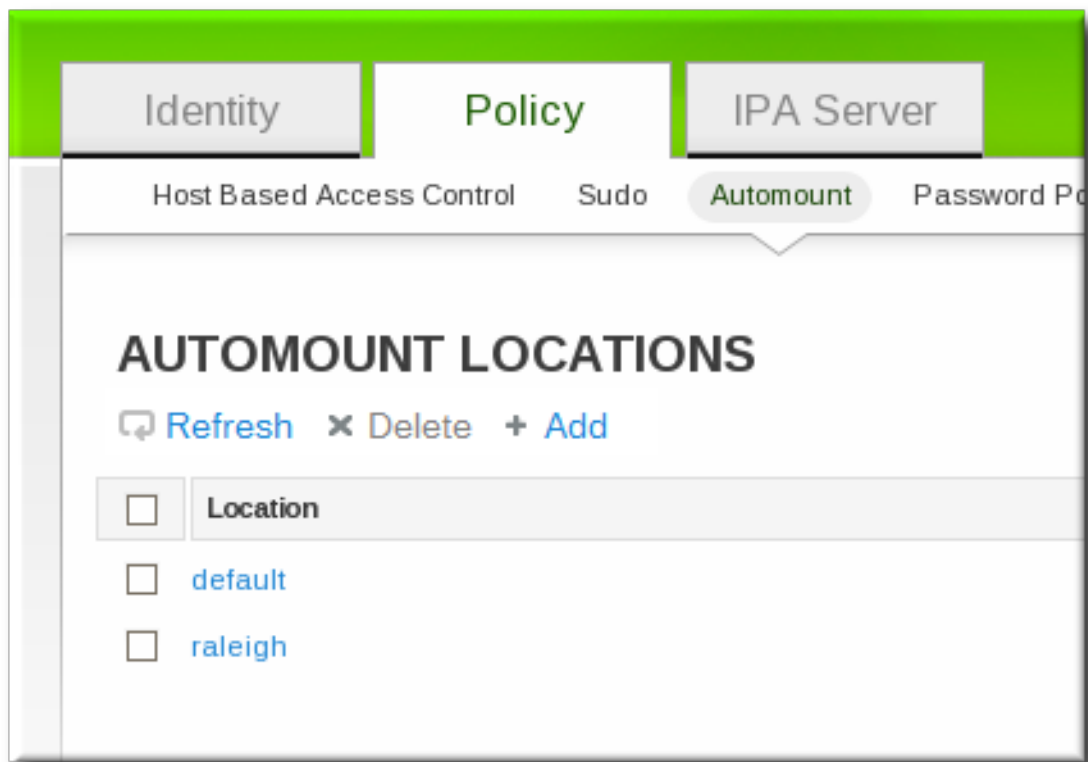


重要

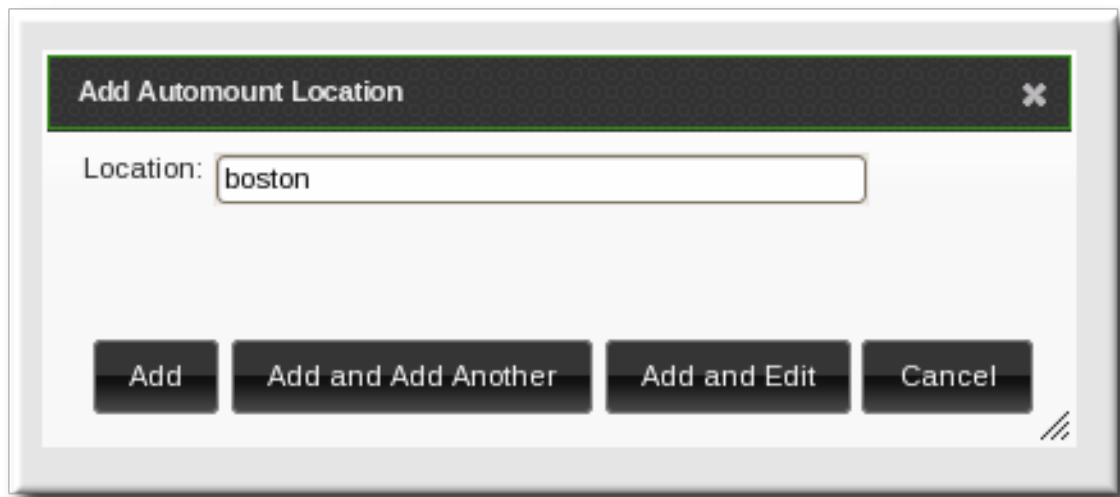
Identity Management は autofs の設置や設定は行いません。これは別個に行う必要があります。Identity Management は既存の autofs デプロイメントと機能するものです。

33.4.1. Web UI での場所の設定

1. **Policy** タブをクリックします。
2. **Automount** サブタブをクリックします。
3. automount の場所一覧の上部にある **Add** をクリックします。



4. 新しい場所の名前を入力します。



5. **Add and Edit** をクリックして、新規の場所のマップ設定に移動します。[「Web UI でのダイレクトマップの設定」](#) および [「Web UI での間接マップの設定」](#) にあるように、マップを作成します。

33.4.2. コマンドラインでの場所の設定

マップを作成するには、**automountlocation-add** を使用して場所の名前を提供します。

```
$ ipa automountlocation-add location
```

例を示します。

```
$ ipa automountlocation-add raleigh
-----
Added automount location "raleigh"
-----
Location: raleigh
```

新規の場所が作成されると、**auto.master** および **auto.direct** という 2 つのマップが自動的に作成されます。**auto.master** は、当該場所の automount マップすべての root マップです。**auto.direct** は直接マウント用のデフォルトマップで、**/-** にマウントされます。

ある場所用に設定されたマップすべてがまるでファイルシステム上に導入されているかのように表示するには、**automountlocation-tofiles** コマンドを使用します。

```
$ ipa automountlocation-tofiles raleigh
/etc/auto.master:
/-      /etc/auto.direct
-----
/etc/auto.direct:
```

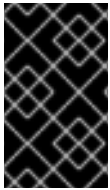
33.5. マップの設定

マップを設定するとマップが作成されるだけでなく、キーによってマウントポイントに関連付けられ、ディレクトリーにアクセスした際に使用するマウントポイントが割り当てられます。IdM は、ダイレクトおよび間接マップの両方をサポートします。



注記

異なるクライアントは別のマップセットを使用できます。マップセットはツリー構造を使用しているので、マップを場所の間で共有することはできません。



重要

Identity Management は autofs の設置や設定は行いません。これは別個に行う必要があります。Identity Management は既存の autofs デプロイメントと機能するものです。

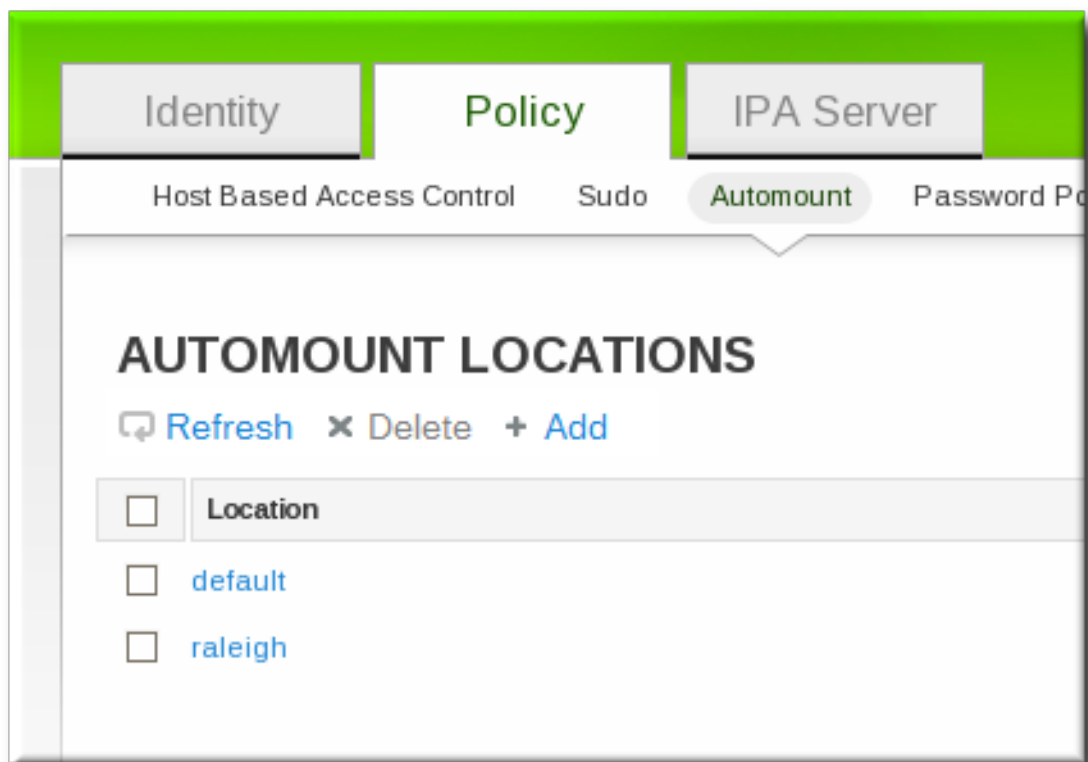
33.5.1. ダイレクトマップの設定

ダイレクトマップは、ファイルマウントポイントへの正確な場所、つまり完全パスを定義します。ローカルエントリーでは、ダイレクトマップは前に付けるフォワードスラッシュで特定されます。

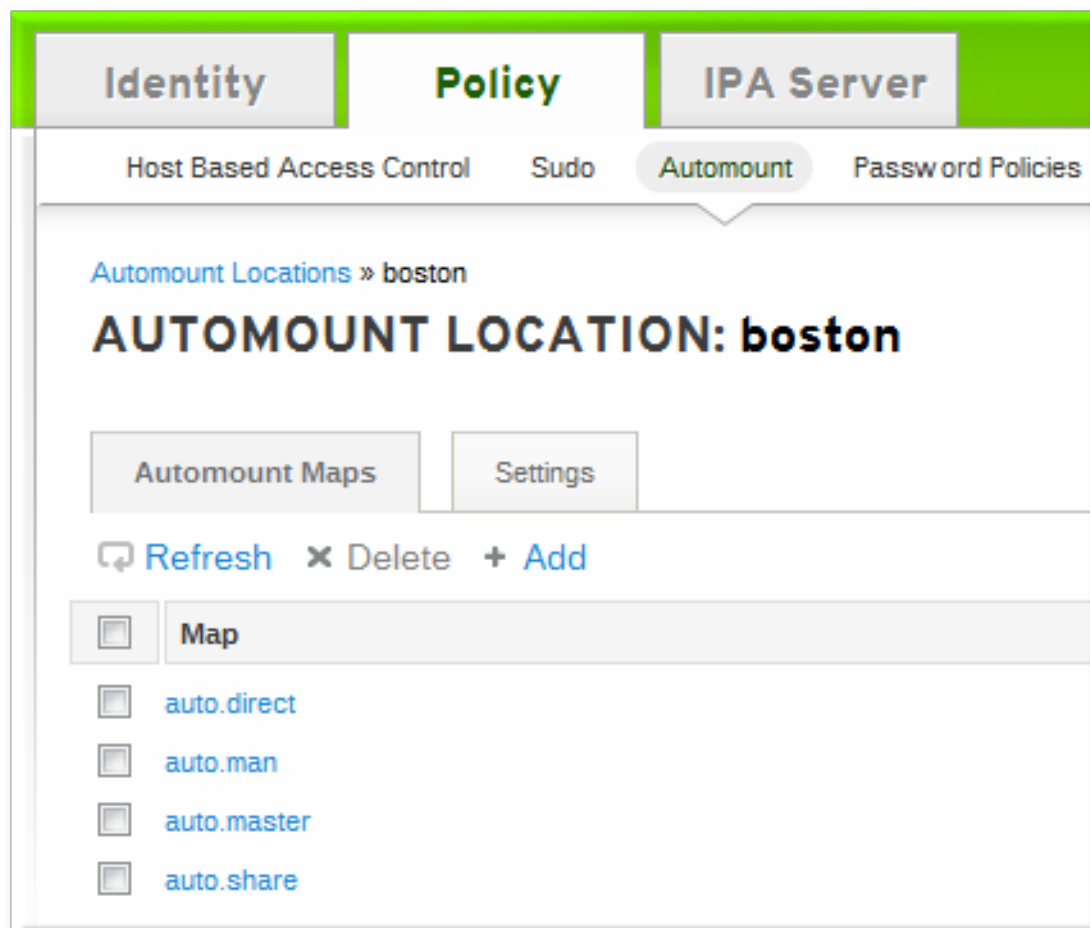
```
-----
/etc/auto.direct:
/shared/man server.example.com:/shared/man
```

33.5.1.1. Web UI でのダイレクトマップの設定

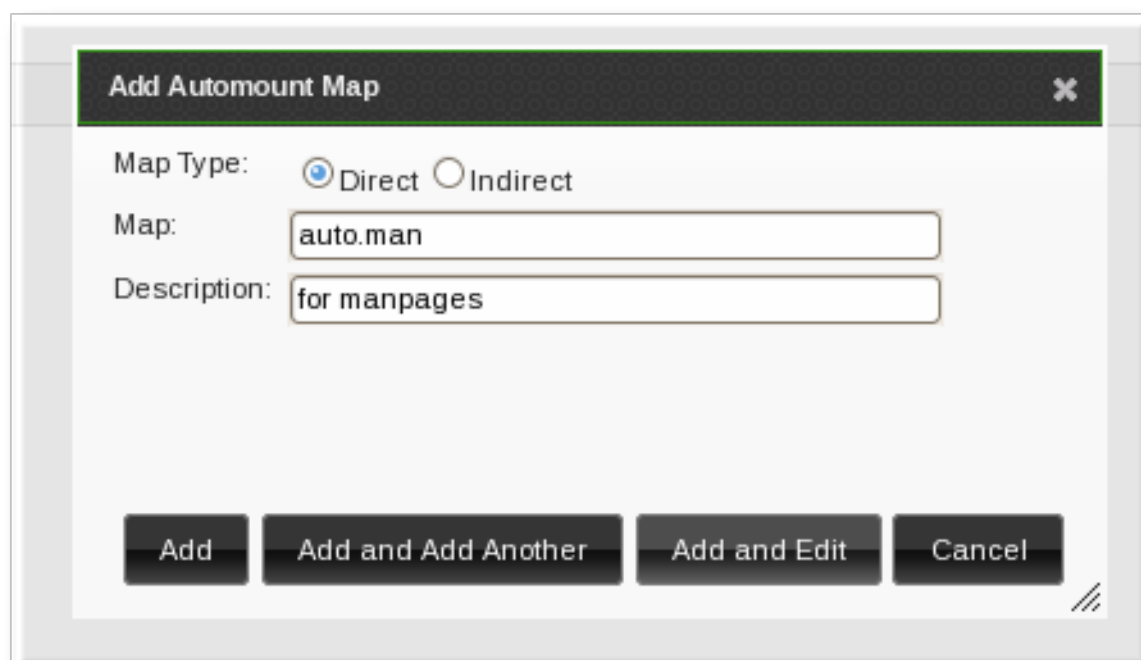
1. **Policy** タブをクリックします。
2. **Automount** サブタブをクリックします。
3. マップの追加先となる automount の場所の名前をクリックします。



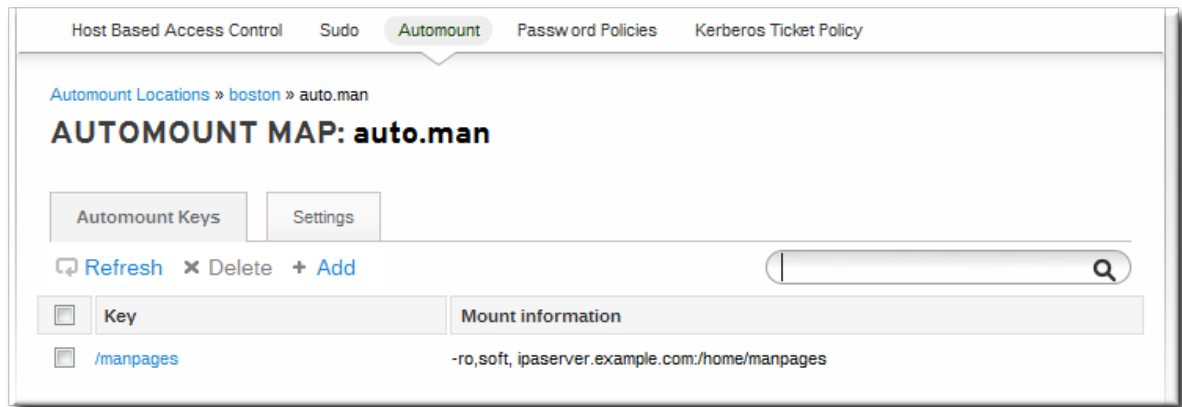
4. **Automount Maps** タブで **Add** をクリックして新規マップを作成します。



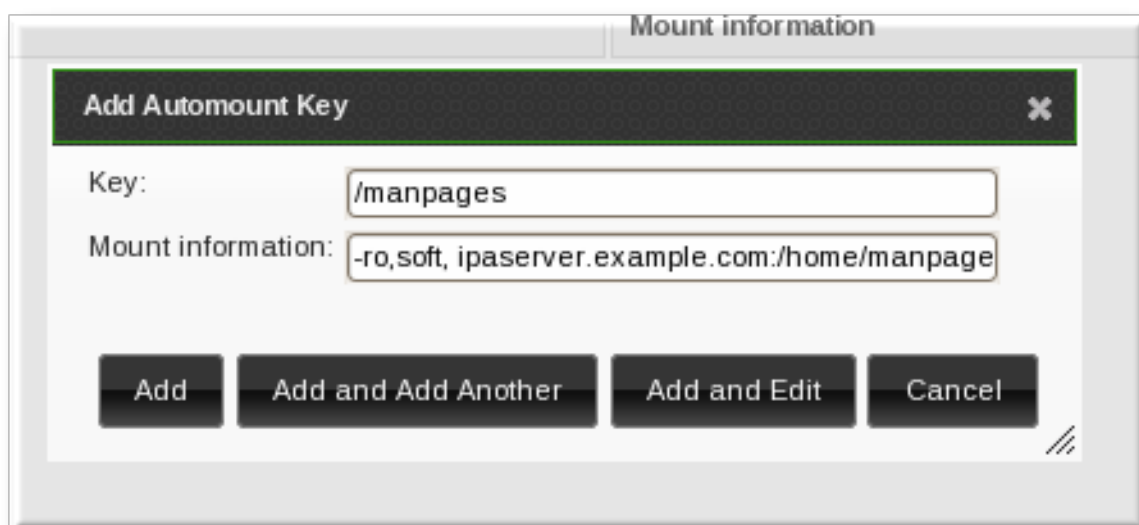
5. ポップアップウィンドウで **Direct** ラジオボタンを選択し、新規マップの名前を入力します。



6. **Automount Keys** タブで **Add** をクリックしてマップの新規キーを作成します。



7. マウントポイントを入力します。key では、実際のマウントポイントを key の名前で定義します。**Info** フィールドでは、ディレクトリーのネットワークの場所と、使用する **mount** オプションを設定します。



8. **Add** をクリックして新規キーを保存します。

33.5.1.2. コマンドラインでのダイレクトマップの設定

key では、実際のマウントポイントとオプションを key の名前で定義します。キーの形式に基づいて、マップはダイレクトまたは間接マップになります。

各場所は **auto.direct** アイテムと共に作成されます。最もシンプルな設定では、automount キーを既存のダイレクトマップエントリーに追加することでダイレクトマップを定義します。異なるダイレクトマップエントリーを作成することも可能です。

ダイレクトマップのキーを場所の **auto.direct** ファイルに追加します。**mount** の他のオプションに加えて、**--key** オプションはマウントポイントを特定し、**--info** はディレクトリーのネットワークの位置を提供します。

```
$ ipa automountkey-add raleigh auto.direct --key=/share --
info="ro,soft,ipaserver.example.com:/home/share"
Key: /share
Mount information: ro,soft,ipaserver.example.com:/home/share
```

Mount のオプションは、man ページ <http://linux.die.net/man/8/mount> で説明されています。

Solaris では、**ldapclient** コマンドで LDAP エントリーを直接追加することで、ダイレクトマップとキーを追加します。

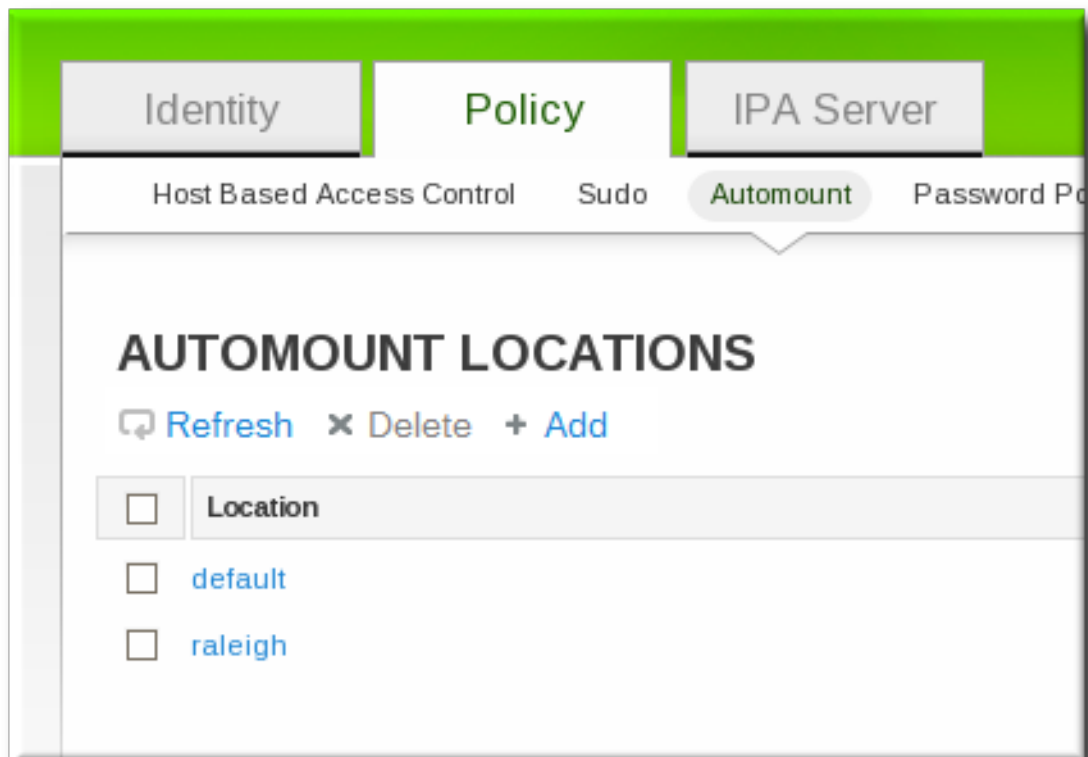
```
ldapclient -a  
serviceSearchDescriptor=auto_direct:automountMapName=auto.direct,cn=location,cn=automount,dc=example,dc=com?one
```

33.5.2. 間接マップの設定

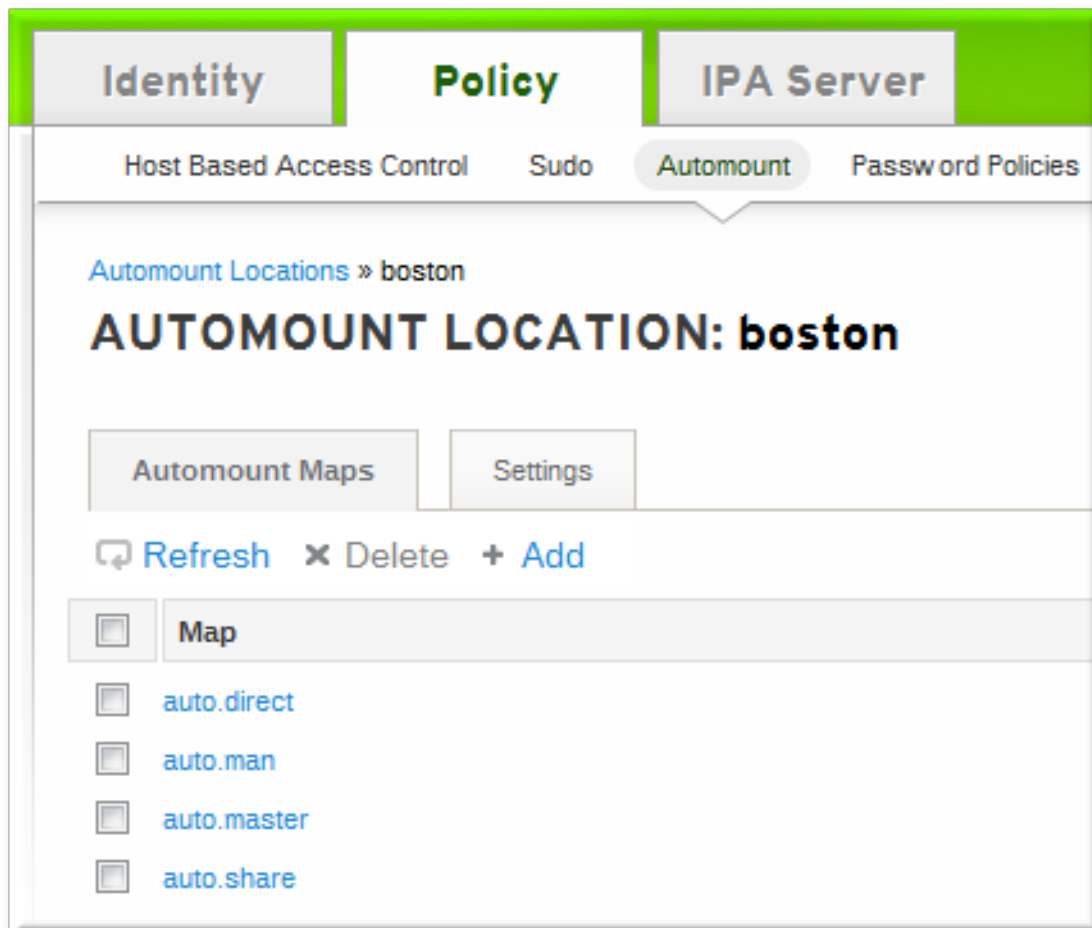
間接マップは、基本的にマップの相対パスを指定するものです。親エントリーがすべての間接マップのベースディレクトリーを設定します。間接マップキーはサブディレクトリーを設定します。間接マップの場所が読み込まれる度に、キーはそのベースディレクトリーに追加されます。たとえば、ベースディレクトリーが **/docs** でキーが **man** の場合、マップは **/docs/man** となります。

33.5.2.1. Web UI での間接マップの設定

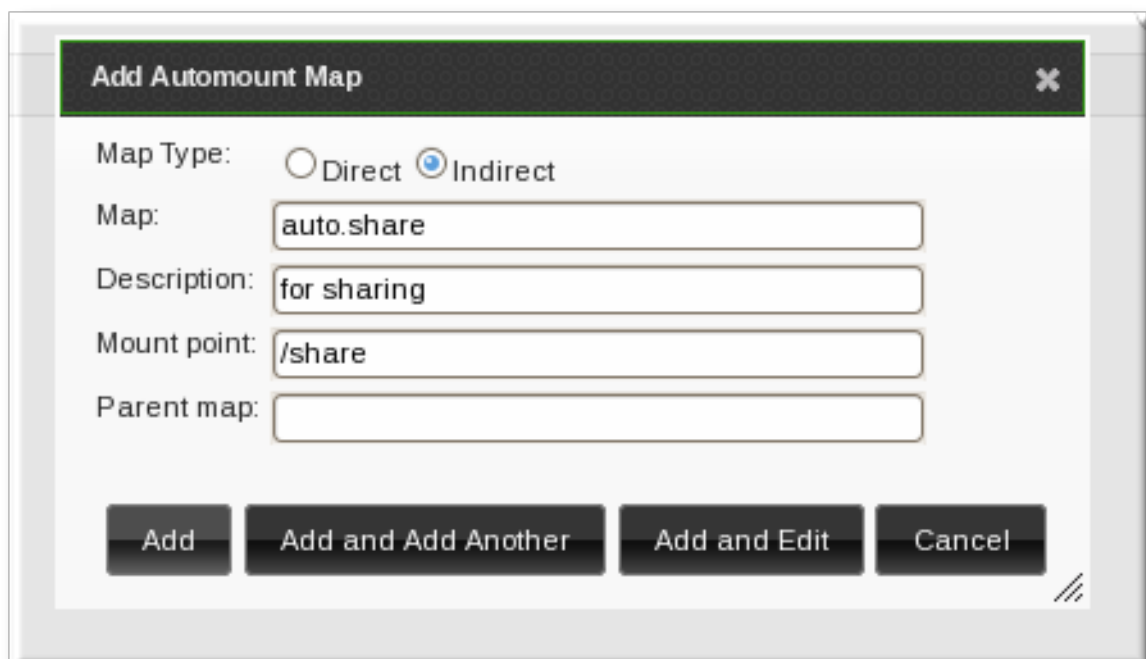
1. **Policy** タブをクリックします。
2. **Automount** サブタブをクリックします。
3. マップの追加先となる automount の場所の名前をクリックします。



4. **Automount Maps** タブで **Add** をクリックして新規マップを作成します。



5. ポップアップウィンドウで **Indirect** ラジオボタンを選択し、以下の必要な間接マップの情報を入力します。



- 。新規マップの名前
- 。マウントポイント。**Mount** フィールドでは、すべての間接マップキーに使用するベースディレクトリーを設定します。

- 。オプションで親マップ。デフォルトの親は **auto.master** ですが、使用する別のマップがある場合は、**Parent Map** フィールドでそれを指定できます。

6. **Add** をクリックして新規キーを保存します。

33.5.2.2. コマンドラインでの間接マップの設定

ダイレクトマップと間接マップの主な違いは、間接キーの前にはフォワードスラッシュがないことです。

```
-----
/etc/auto.share:
man      ipa.example.com:/docs/man
-----
```

1. **automountmap-add-indirect** コマンドを使って間接マップを作成し、ベースエントリを設定します。 **--mount** オプションでは、すべての間接マップキーに使用するベースディレクトリを設定します。デフォルトの親エントリは **auto.master** ですが、使用する別のマップがある場合は、 **--parentmap** オプションを使って指定することができます。

```
$ ipa automountmap-add-indirect location mapName --mount=directory
[--parentmap=mapName]
```

例を示します。

```
$ ipa automountmap-add-indirect raleigh auto.share --mount=/share
-----
Added automount map "auto.share"
-----
```

2. マウントする場所の間接キーを追加します。

```
$ ipa automountkey-add raleigh auto.share --key=docs --
info="ipa.example.com:/export/docs"
-----
Added automount key "docs"
-----
Key: docs
Mount information: ipa.example.com:/export/docs
```

3. 設定を確認するために、 **automountlocation-tofiles** を使って場所のファイル一覧をチェックします。

```
$ ipa automountlocation-tofiles raleigh
/etc/auto.master:
/-      /etc/auto.direct
/share  /etc/auto.share
-----
/etc/auto.direct:
-----
/etc/auto.share:
man      ipa.example.com:/export/docs
```

Solaris では、**ldapclient** コマンドで LDAP エントリーを直接追加することで、間接マップを追加します。

```
ldapclient -a  
serviceSearchDescriptor=auto_share:automountMapName=auto.share,cn=location  
,cn=automount,dc=example,dc=com?one
```

33.5.3. Automount マップのインポート

既存の automount マップがある場合は、それを IdM automount 設定にインポートすることができます。

```
ipa automountlocation-import location map_file [--continuous]
```

必要となる情報は、IdM automount の場所とマップファイルの完全パスおよびファイル名のみです。--**continuous** オプションを使うと、**automountlocation-import** コマンドはエラーに遭遇してもマップファイルの最後までインポートを継続します。

例を示します。

```
$ ipa automountlocation-import raleigh /etc/custom.map
```

パート **VIII.** セキュリティーの強化

第34章 IDENTITY MANAGEMENT 向け TLS の設定

ここでは、Red Hat Enterprise Linux 7.3 以降で Identity Management が TLS プロトコルバージョン 1.2 を必要とするように設定する方法について説明します。

TLS 1.2 は TLS の以前のバージョンよりも安全であるとみなされています。お使いの IdM サーバーが高セキュリティ要件の環境にデプロイされている場合、同サーバーが TLS 1.2 よりも安全ではないプロトコルを使用して通信できないように設定することが可能です。



重要

TLS 1.2 を使用する各 IdM サーバーで以下のステップを繰り返してください。

34.1. HTTPD デーモンの設定

1. `/etc/httpd/conf.d/nss.conf` ファイルを開いて、`NSSProtocol` と `NSSCipherSuite` のエントリーで以下の値を設定します。

```
NSSProtocol TLSv1.2
NSSCipherSuite
+ecdhe_ecdsa_aes_128_sha,+ecdhe_ecdsa_aes_256_sha,+ecdhe_rsa_aes_128
_sha,+ecdhe_rsa_aes_256_sha,+rsa_aes_128_sha,+rsa_aes_256_sha
```

もしくは、以下のコマンドでこれらの値を設定します。

```
# sed -i 's/^NSSProtocol .*/NSSProtocol TLSv1.2/'
/etc/httpd/conf.d/nss.conf
# sed -i 's/^NSSCipherSuite .*/NSSCipherSuite
+ecdhe_ecdsa_aes_128_sha,+ecdhe_ecdsa_aes_256_sha,+ecdhe_rsa_aes_128
_sha,+ecdhe_rsa_aes_256_sha,+rsa_aes_128_sha,+rsa_aes_256_sha/'
/etc/httpd/conf.d/nss.conf
```

2. `httpd` デーモンを再起動します。

```
# systemctl restart httpd
```

34.2. DIRECTORY SERVER コンポーネントの設定

Directory Server (DS) を手動で設定するには、以下を実行します。

1. DS を停止します。

```
# systemctl stop dirsrv@EXAMPLE-COM.service
```

2. `/etc/dirsrv/slapd-EXAMPLE-COM/dse.ldif` ファイルを開いて、`cn=encryption,cn=config` エントリーを以下に設定します。

```
sslVersionMin: TLS1.2
```

3. DS を起動します。

```
# systemctl start dirsrv@EXAMPLE-COM.service
```

もしくは **ldapmodify** ユーティリティーを使用して DS を自動設定します。

1. **ldapmodify** を使って設定を変更します。

```
ldapmodify -h localhost -p 389 -D 'cn=directory manager' -W << EOF
dn: cn=encryption,cn=config
changeType: modify
replace: sslVersionMin
sslVersionMin: TLS1.2
EOF
```

2. DS を再起動して新しい設定を読み込みます。

```
# systemctl restart dirsrv@EXAMPLE-COM.service
```

34.3. 証明書サーバーコンポーネントの設定

1. 証明書サーバー (CS) を手動で設定するには、`/etc/pki/pki-tomcat/server.xml` ファイルを開き、**sslVersionRangeStream** と **sslVersionRangeDatagram** のパラメーターをすべて以下の値に設定します。

```
sslVersionRangeStream="tls1_2:tls1_2"
sslVersionRangeDatagram="tls1_2:tls1_2"
```

もしくは、以下のコマンドでこれらの値を設定します。

```
# sed -i 's/tls1_[01]:tls1_2/tls1_2:tls1_2/g' /etc/pki/pki-
tomcat/server.xml
```

2. CS を再起動します。

```
# systemctl restart pki-tomcatd@pki-tomcat.service
```

34.4. 結果

Identity Management サーバーは TLS 1.2 が必要な設定になっています。このため、TLS の以前のバージョンにしか対応しない Identity Management クライアントは、Identity Management との通信ができなくなっています。

第35章 ANONYMOUS バインドの無効化

ドメインのリソースにアクセスしてクライアントのツールを実行する場合は、常に Kerberos 認証が必要になります。ただし、IdM サーバーで使用されるバックエンドの LDAP ディレクトリーにより、anonymous バインドはデフォルトで許可されます。これによりユーザーやマシン、グループ、サービス、ネットグループ、DNS 設定などのドメインの全設定が非認証ユーザーに公開されてしまう可能性があります。

LDAP ツールを使って **nsslapd-allow-anonymous-access** 属性をリセットすると 389 Directory Server インスタンスで anonymous バインドを無効にすることができます。

1. **nsslapd-allow-anonymous-access** 属性を **rootdse** に変更します。

```
$ ldapmodify -x -D "cn=Directory Manager" -W -h server.example.com -
p 389 -ZZ
Enter LDAP Password:
dn: cn=config
changetype: modify
replace: nsslapd-allow-anonymous-access
nsslapd-allow-anonymous-access: rootdse

modifying entry "cn=config"
```

重要

Anonymous アクセスは完全に許可したり (on) ブロックしたり (off) することができます。ただし、anonymous アクセスを完全にブロックすると外部クライアントがサーバー設定をチェックすることもできなくなります。LDAP および web クライアントはドメインクライアントに限られるわけではないため、こうしたクライアントは匿名で接続を行ってルートの DSE ファイルを読み取り接続情報を取得します。

rootdse はディレクトリーデータにはアクセスさせずにルート DSE とサーバー設定へのアクセスを許可します。

2. 389 Directory Server インスタンスを再起動して新しい設定をロードします。

```
# systemctl restart dirsrv.target
```

パート **IX.** パフォーマンスチューニング

第36章 エントリーの一括プロビジョニングのパフォーマンスチューニング

ユーザーの追加 (11章 [ユーザーアカウントの管理](#)) など、通常のワークフローを使用して大量のエントリーを追加すると、非常に時間がかかる可能性があります。本章では、プロビジョニングをできるだけ早く完了できるようにプロセスを調節する方法を説明します。

手順には以下が含まれます。

- Identity Management (IdM) は LDIF ファイルからプロビジョニングされたエントリーを読み込み、対象の IdM LDAP インスタンスにインポートします。
- 管理者は、キャッシュサイズなど特定の属性のカスタム値を設定し、MemberOf および Schema Compatibility プラグインを無効化します。この手順では、MemberOf が無効になったことに対応するため、プロビジョニングされたエントリーで **fixup-memberof.pl** プラグインを実行します。

この手順は、以下のエントリータイプ (ユーザー、ユーザーグループ、ホスト、ホストグループ、sudo ルール、ホストベースのアクセス制御 (HBAC)) をプロビジョニングするために、設計、テストされました。

一括プロビジョニングの推奨事項と前提条件

推奨事項:

- 大量のエントリー (10,000 件以上) をプロビジョニングする場合には、LDAP クライアントが、エントリーのプロビジョニング先のサーバーにアクセスしたり、サーバーの情報に依存したりできないようにします。たとえば、これはサーバーのポート 389 および 636 を無効にするか、Unix ソケットで機能する LDAPi を使用することで実現できます。

理由: MemberOf プラグインはサーバー上で無効になっているのでサーバーのメンバーシップの情報は無効です。

- プロビジョニング中に実行しておく必要がないアプリケーションは停止します。

理由: これにより、できるだけマシン上のメモリーを開放されます。ファイルシステムのキャッシュが空いたメモリーを使用することで、プロビジョニングのパフォーマンスを向上します。

以下の手順にはすでに IdM サービスを停止して、Directory Server (DS) インスタンスのみを再起動する手順が含まれている点に注意してください。**tomcat** などの IdM サービスは大量のメモリーを消費しますが、プロビジョニング中には使用されません。

- 新規インストールされた IdM デプロイメントで、サーバーが 1 台のみ含まれている場合に、この手順を実行してください。プロビジョニングの完了後のみ、レプリカを作成します。

理由: プロビジョニングのスループットは、レプリケーションよりもはるかに早いので、デプロイメントに 1 台以上含まれていると、レプリカの情報的大幅に古くなってしまいます。

前提条件:

- プロビジョニングするエントリーが含まれる LDIF ファイルを生成します。たとえば、既存の IdM デプロイメントを移行する場合には、**ldapsearch** ユーティリティーを使用して全エントリーをエクスポートし、LDIF ファイルを作成します。

LDIF 形式の詳細は、『Red Hat Directory Server Administration Guide』の「[About the LDIF File Format](#)」を参照してください。

現在の DS チューニングパラメーター値のバックアップ

1. DS チューニングパラメーターの現在の値を取得します。

- 。 データベースのキャッシュサイズおよびデータベースのロック

```
# ldapsearch -D "cn=directory manager" -w secret -b
"cn=config,cn=ldb database,cn=plugins,cn=config" nsslapd-
dbcachesize nsslapd-db-locks

...
nsslapd-dbcachesize: 10000000
nsslapd-db-locks: 50000
...
```

- 。 エントリーのキャッシュサイズと DN キャッシュサイズ

```
# ldapsearch -D "cn=directory manager" -w secret -b
"cn=userRoot,cn=ldb database,cn=plugins,cn=config" nsslapd-
cachememsize nsslapd-dncachememsize

...
nsslapd-cachememsize: 10485760
nsslapd-dncachememsize: 10485760
...
```

2. 取得した値をメモします。プロビジョニングが完了したら、パラメーターをこれらの値にリセットします。

データベース、ドメインエントリー、DN キャッシュサイズの調節

データベースのキャッシュサイズの場合:

1. 必要な値を決定します。

推奨の値は通常 200 MB から 500 MB の間となっています。各ユースケースに適した値は、システムで利用できるメモリーにより異なります。

- 。 メモリー 8 GB 以上 → 500 MB
- 。 メモリー 8 GB - 4 GB → 200 MB
- 。 メモリー 4 GB 未満 → 100 MB

2. 以下のテンプレートを使用して、決定した値を設定します。

```
dn: cn=config,cn=ldb database,cn=plugins,cn=config
changetype: modify
replace: nsslapd-dbcachesize
nsslapd-dbcachesize: db_cache_size_in_bytes
```

ldapmodify ユーティリティーを使用して LDAP 属性を変更する例は、[例36.1 「ldapmodify を使用した LDAP 属性の変更」](#)を参照してください。

例36.1 ldapmodify を使用した LDAP 属性の変更

1. **ldapmodify** コマンドを実行して、属性値を変更するステートメントを追加します。以下に例を示します。

```
# ldapmodify -D "cn=directory manager" -w secret -x
dn: cn=config,cn=ldb database,cn=plugins,cn=config
changetype: modify
replace: nsslapd-dbcachesize
nsslapd-dbcachesize: 200000000
```

2. **Ctrl+D** を押して、サーバーへの変更を確定、送信します。操作が正常に完了したら、以下のメッセージが表示されます。

```
modifying entry "cn=config,cn=ldb database,cn=plugins,cn=config"
```

ドメインエントリーのキャッシュサイズの場合:

1. 必要な値を決定します。

推奨の値は 100 MB から 400 MB の間となっています。適切な値は、お使いのシステムで利用可能なメモリにより異なります。

- 。メモリ 4 GB 以上 → 400 MB
- 。メモリ 2 GB - 4 GB → 200 MB
- 。メモリ 2 GB 未満 → 100 MB

大規模な静的グループをプロビジョニングする場合には、エントリーキャッシュに、グループおよびメンバーなど全エントリーが格納できるサイズにすることを推奨します。

2. 以下のテンプレートを使用して、決定した値を設定します。

```
dn: cn=userRoot,cn=ldb database,cn=plugins,cn=config
changetype: modify
replace: nsslapd-cachememsize
nsslapd-cachememsize: entry_cache_size_in_bytes
```

ドメイン名 (DN) のキャッシュサイズの場合:

1. 最適なパフォーマンスを得るには、DN キャッシュに、プロビジョニングしたエントリーの DN すべてを格納できるようにすることを推奨します。ユースケースに適した値を見積もるには以下を行ってください。

- a. ファイル内の全 DN エントリー数を確認します。DN エントリーは、**dn:** で始まる行に含まれます。たとえば、**# grep**、**sed**、**wc** を使用して確認します。

```
# grep '^dn: ' ldif_file | sed 's/^dn: //' | wc -l
92200
```

- b. LDIF ファイルに含まれる全 DN エントリーの文字列のサイズを確認します。

```
# grep '^dn: ' ldif_file | sed 's/^dn: //' | wc -c
9802460
```

- c. 平均の DN サイズの取得: ファイル内の全 DN エントリー数で、全 DN エントリーの文字列のサイズを除算します。

例: $9,802,460 / 92,200 \approx 106$

- d. 平均のメモリーサイズの取得: 平均の DN サイズに 2 を乗算した数値に、32 を加算します。

例: $(106 * 2) + 32 = 244$

- e. 適切な DN のキャッシュサイズを取得: 平均メモリーサイズで LDIF ファイルの DN エントリーの総数を乗算します。

例: $244 * 92,200 = 22,496,800$

2. 以下のテンプレートを使用して、決定した値を設定します。

```
dn: cn=userRoot,cn=ldb database,cn=plugins,cn=config
changetype: modify
Replace: nsslapd-dncachememsize
Nsslapd-dncachememsize: dn_cache_size
```

不要なサービスの無効化およびデータベースロックの調節

1. MemberOf および Schema Compatibility プラグインを無効にします。

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: off
```

```
dn: cn=Schema Compatibility,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: off
```

MemberOf を無効にすると、プロビジョニングが大幅に加速されます。また、Schema Compatibility を無効にすると、操作の時間を短縮することができます。

ldapmodify ユーティリティーを使用して LDAP 属性を変更する例は、[例36.1 「ldapmodify を使用した LDAP 属性の変更」](#)を参照してください。

2. トポロジーにレプリカがインストールされていない場合には (「[一括プロビジョニングの推奨事項と前提条件](#)」で推奨)、Content Synchronization および Retro Changelog プラグインを無効にします。

```
dn: cn=Content Synchronization,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: off
```

```
dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: off
```

■

これらのプラグインを無効にすると、プロビジョニングのパフォーマンス向上に役立ちます。

3. IdM サーバーを停止します。これにより、DS インスタンスも停止されます。

```
# ipactl stop
```

次の手順でデータベースのロックの数を設定するには、DS を停止する必要があります。後で DS を再起動します。

4. データベースロックの数を調節します。プロビジョニングのエントリー数の半分が適切な値です。

- 最小値は 10,000 です。
- 最大値は 200,000 です。

DS が停止されたので、`/etc/dirsrv/slapd-EXAMPLE-COM/dse.ldif` ファイルを編集して値を設定する必要があります。

```
dn: cn=config,cn=ldbm database,cn=plugins,cn=config
...
nsslapd-db-locks: db_lock_number
```

IdM は、メンバーシップのコンピューティングの際には、大容量のデータベースページにアクセスします。アクセスするページが多いと、プロビジョニングに必要なロックも増えます。

5. DS を起動します。

```
# systemctl start dirsrv.target
```

エントリーのインポート

LDIF ファイルから IdM LDAP インスタンスに新規エントリーをインポートするには、**ldapadd** ユーティリティなどを使用します。

```
# ldapadd -D "binddn" -y password_file -f ldif_file
```

ldapadd の使用方法については、`ldapadd(1)` の man ページを参照してください。

無効にしたサービスの再有効化および元の属性値の復元

1. MemberOf を有効にします。

```
dn: cn=MemberOf Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

ldapmodify ユーティリティを使用して LDAP 属性を変更する例は、[例36.1 「ldapmodify を使用した LDAP 属性の変更」](#)を参照してください。

2. DS を再起動します。

```
# systemctl restart dirsrv.target
```

MemberOf を以前の手順で有効化したので、この時点で DS を再起動する必要があります。

3. **fixup-memberof.pl** のスクリプトに (**objectClass=***) フィルターを使用して実行し、全プロビジョニングエントリーの **memberOf** 属性を再生成、更新します。以下に例を示します。

```
# fixup-memberof.pl -D "cn=directory manager" -j password_file -Z
server_id -b "suffix" -f "(objectClass=*)" -P LDAP
```

エントリーのインポートの際に、MemberOf プラグインを無効にしたので、**fixup-memberof.pl** を実行する必要があります。スクリプトが正しく完了しないと、プロビジョニングを続行できません。

fixup-memberof.pl の詳細は、fixup-memberof.pl(8) の man ページを参照してください。

4. Schema Compatibility プラグインを有効にします。

```
dn: cn=Schema Compatibility,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

5. Content Synchronization および Retro Changelog プラグインを「[不必要なサービスの無効化およびデータベースロックの調節](#)」で無効にした場合には、有効化しなおしてください。

```
dn: cn=Content Synchronization,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

```
dn: cn=Retro Changelog Plugin,cn=plugins,cn=config
changetype: modify
replace: nsslapd-pluginEnabled
nsslapd-pluginEnabled: on
```

6. 「[現在の DS チューニングパラメーター値のバックアップ](#)」でバックアップしたデータベースキャッシュ、エントリーキャッシュ、DN キャッシュサイズの元の値を復元します。

```
dn: cn=config,cn=ldbm database,cn=plugins,cn=config
changetype: modify
replace: nsslapd-dbcachesize
nsslapd-dbcachesize: backup_db_cache_size
```

```
dn: cn=userRoot,cn=ldbm database,cn=plugins,cn=config
changetype: modify
Replace: nsslapd-dncachememsize
Nsslapd-dncachememsize: backup_dn_cache_size
-
replace: nsslapd-cachememsize
nsslapd-cachememsize: backup_entry_cache_size
```

7. DS を停止します。

```
# systemctl stop dirsrv.target
```

-
8. 「現在の DS チューニングパラメーター値のバックアップ」でバックアップしたデータベースロックの元の値を復元します。DS が停止されたので、**/etc/dirsrv/slapd-EXAMPLE-COM/dse.ldif** ファイルを編集して値を設定する必要があります。

```
dn: cn=config,cn=ldbm database,cn=plugins,cn=config
...
nsslapd-db-locks: backup_db_lock_number
```

9. IdM サーバーを起動します。

```
# ipactl start
```

これにより、DS を含むすべての IdM サービスが起動します。

パート **X.** 移行

第37章 LDAP ディレクトリーから IDM への移行

管理者として認証およびアイデンティティルックアップ用の LDAP サーバーをデプロイしたので、次に Identity Management にバックエンドを移行します。IdM 移行ツールを使用して、パスワードやグループなどユーザーアカウントをデータを損失することなく転送します。また、クライアント上での大規模なコストのかかる設定更新を回避することができます。

ここに記載の移行プロセスは、LDAP に名前空間が 1 つ、IdM に 1 つという単純なデプロイメントシナリオが想定されています。複数の名前空間やカスタムのスキーマなど、より複雑な環境については、Red Hat サポートサービスにお問い合わせください。

37.1. LDAP から IDM への移行に関する概要

LDAP サーバーから Identity Management に移動する実際の移行部分はかなり単純です (1 つのサーバーから別のサーバーにデータを移動させるプロセス)。データ、パスワード、クライアントの順で移動する単純なプロセスです。

最もコストのかかる移行プロセスは、Identity Management を使用するようにクライアントをどのように設定するかを決定する部分です。インフラストラクチャーのクライアントごとに、どのサービス (Kerberos、SSSD など) を使用して最終的な IdM デプロイメントで使用可能なサービスがどれかを決定する必要があります。

2 番目に重要な考慮事項はパスワードの移行方法の計画です。Identity Management ではパスワードに加え各ユーザーアカウントすべてに Kerberos ハッシュが必要になります。パスワードの移行パスおよび考慮すべき点については、いくつか「[パスワード移行のプランニング](#)」で説明しています。

37.1.1. クライアント設定のプランニング

Identity Management はさまざまなレベルの機能性、柔軟性、安全性で多数の異なるクライアント設定に対応することができます。クライアントのオペレーティングシステム、機能領域 (開発用マシン、実稼動サーバー、ユーザーのラップトップ)、IT メンテナンスの優先性などに応じて **クライアントごと個別に** 最適となる設定を選択してください。



重要

異なるクライアント設定は **相互に排他的とはなりません**。ほとんどの環境でクライアントが IdM ドメインへの接続に使用する方法はクライアントによって異なります。管理者は各クライアント別に最適となるシナリオを決定しなければなりません。

37.1.1.1. クライアント初期設定 (移行前)

Identity Management でのクライアント設定を決定する前にまず移行前の状態を確認します。

移行予定の LDAP デプロイメントの初期の状態の場合、ほとんど全てに ID および認証サービスを提供している LDAP サービスがあります。

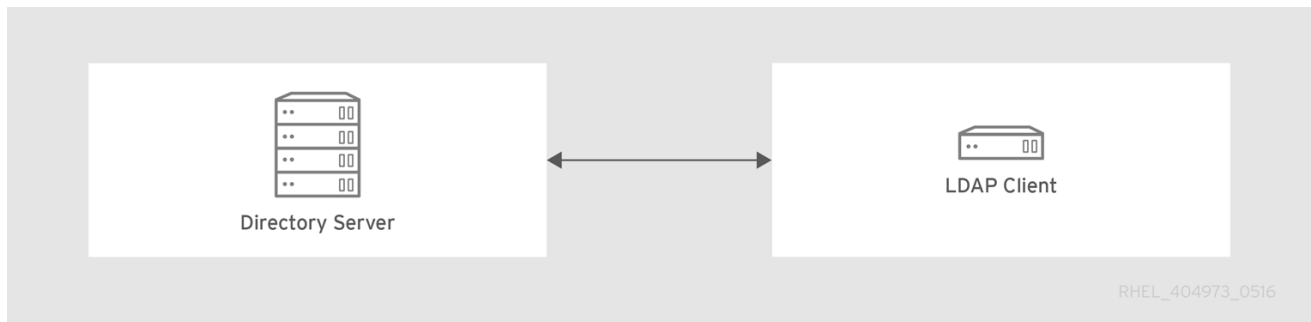


図37.1 基本的な LDAP ディレクトリーとクライアント設定

Linux および Unix のクライアントは PAM_LDAP と NSS_LDAP ライブラリーを使って LDAP サービスに直接接続を行います。このライブラリーを使ってクライアントはユーザー情報を LDAP ディレクトリーから取得します。まるでデータが `/etc/passwd` や `/etc/shadow` に格納されていたかのように見えます。(現実的には ID 検索に LDAP、認証に Kerberos や別の設定を使用している場合などインフラストラクチャーはもう少し複雑になる場合があります。)

LDAP ディレクトリーと IdM サーバーの間には特にスキーマサポートとディレクトリーツリーに構造的な違いがあります (構造的な違いの背景については「[Identity Management と標準 LDAP ディレクトリーの比較](#)」を参照してください)。こうした違いはデータ (特にエントリー名に影響するディレクトリーツリー) には影響する可能性があります。クライアントの設定 にはほとんど影響しないため、Identity Management にクライアントを移行させる上では実際にはほとんど影響がありません。

37.1.1.2. Red Hat Enterprise Linux クライアント向けの推奨設定

Red Hat Enterprise Linux には **System Security Services Daemon (SSSD)** と呼ばれるサービスがあります。SSSD は特殊な PAM と NSS ライブラリー (`pam_sss` と `nss_sss`) を使用します。このライブラリーによって SSSD と Identity Management の緊密な統合が行われ、Identity Management の認証機能および ID 機能をフル活用できるようになります。中央サーバーとの接続が失われた場合でもユーザーがログインできるよう ID 情報をキャッシングできる機能など、SSSD には便利な機能が多数搭載されています。こうした便利な機能については『システムレベルの認証ガイド』で詳しく説明しています。

汎用の LDAP ディレクトリーサービス (`pam_ldap` と `nss_ldap` を使用する) とは異なり、SSSD はドメイン 定義によって ID 情報と認証情報間の関係を確立します。SSSD のドメインは認証、ID 検索、アクセス、パスワード変更の 4 つのバックエンド機能を定義します。この SSSD ドメインを 4 つの機能のうちの 1 つの機能 (またはすべて) の情報を提供する **プロバイダー** を使用するよう設定します。ID プロバイダーはドメイン設定に必ず必要になります。他の 3 つのプロバイダーはオプションです。認証、アクセス、またはパスワードプロバイダーが定義されていない場合は ID プロバイダーがその機能に使用されます。

SSSD ではそのすべてのバックエンド機能に Identity Management を使用できるため理想的な設定になります。LDAP ID の汎用プロバイダーや Kerberos 認証とは異なり、多岐に渡る Identity Management の機能性をすべて利用することができます。たとえば、SSSD では 日常的な運用時に、Identity Management でセキュリティー機能やホストベースのアクセス制御ルールを有効化させることができます。



注記

LDAP ディレクトリーから Identity Management への移行プロセスではユーザーによる介入を必要とすることなくユーザーのパスワード移行が SSSD によりシームレスに行われます。

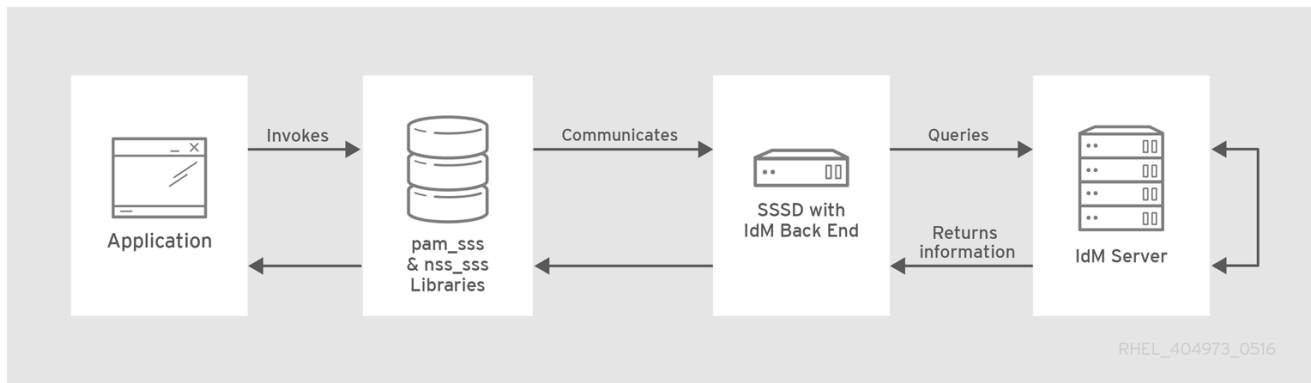


図37.2 IdM バックエンドによるクライアントおよび SSSD

ipa-client-install スクリプトにより自動的に SSSD がバックエンドの全 4 サービスに IdM を使用するよう設定されるため、Red Hat Enterprise Linux クライアントはデフォルトで推奨設定にセットアップされます。



注記

このクライアント設定に対応するのは最新の SSSD と **ipa-client** のバージョンに対応する Red Hat Enterprise Linux 6.1 以降および Red Hat Enterprise Linux 5.7 以降のみになります。これより旧式の Red Hat Enterprise Linux については「[推奨設定以外で対応している設定](#)」の説明に従って設定を行ってください。

37.1.1.3. 推奨設定以外で対応している設定

Mac、Solaris、HP-UX、AIX、Scientific Linux などの Unix および Linux システムでは IdM で管理されるすべてのサービスに対応していますが SSSD は使用しません。同様に旧式の Red Hat Enterprise Linux バージョン (6.1 および 5.6) は SSSD には対応しますが、ID プロバイダーの IdM には対応していない旧バージョンが搭載されています。

最近の SSSD バージョンを使用できない場合は、IdM サーバーへの接続は ID 検索性 LDAP ディレクトリーサービスへの接続のようにクライアントを設定します (**nss_ldap** を使用)。また IdM への接続は通常の Kerberos KDC への接続のように設定を行います (**pam_krb5** を使用)。

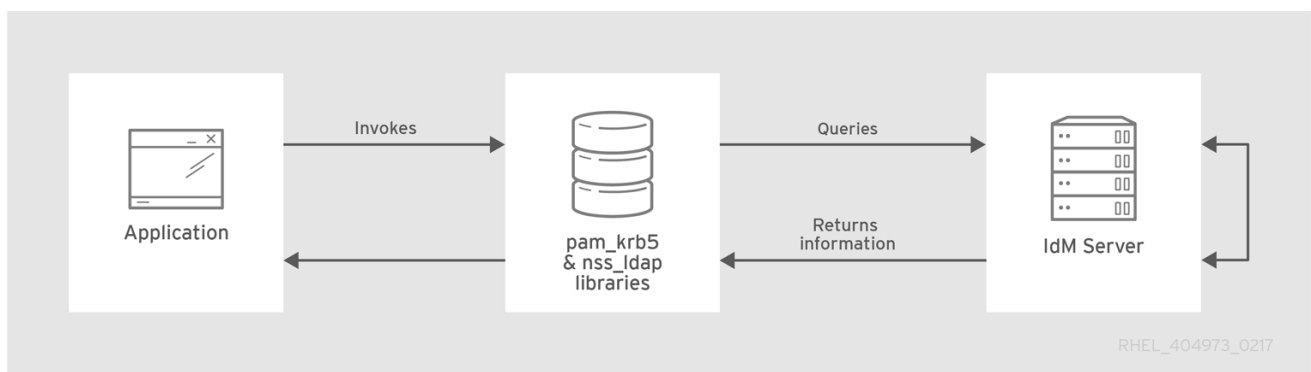


図37.3 LDAP と Kerberos を使用したクライアントと IdM

Red Hat Enterprise Linux クライアントで旧式の SSSD バージョンを使用している場合、IdM サーバーを ID プロバイダーおよびその Kerberos 認証ドメインとして使用するよう SSSD を設定することができます。詳細については『システムレベルの認証ガイド』の SSSD 設定セクションで説明しています。

IdM ドメインクライアントはいずれも、**nss_ldap** と **pam_krb5** を使用して IdM サーバーに接続するよう設定することができます。共通する構成要素が最低限となるようなメンテナンス環境や IT インフ

ラストラクチャーなどの場合には LDAP を ID と認証の両方に使用する必要があるかもしれません (`nss_ldap` と `pam_ldap`)。ただし、クライアントにはできる限り安全な設定を使用するのが一般的にはベストプラクティスです (つまり SSSD と Kerberos または LDAP と Kerberos)。

37.1.2. パスワード移行のプランニング

LDAP から Identity Management への移行に影響する可能性がある問題としてもっともよく知られているのが恐らくユーザーのパスワード移行でしょう。

Identity Management では認証に Kerberos を使用し (デフォルト)、各ユーザーには標準のユーザーパスワード以外にも Identity Management Directory Server に格納する Kerberos ハッシュが必要になります。このハッシュを生成するため、IdM サーバー側でユーザーのパスワードがクリアテキストで使用できなければなりません。ユーザーの作成時は、パスワードがハッシュ化されて、Identity Management に保存される前に、パスワードをクリアテキストの状態で行き渡すことができます。ただし、ユーザーを LDAP ディレクトリーから移行する場合には関連するユーザーパスワードがすでにハッシュ化されているため該当する Kerberos キーは生成できません。



重要

ユーザーに Kerberos ハッシュを持たせるまでユーザーによる IdM ドメインへの認証や IdM リソースへのアクセスは行えません。

ユーザーに Kerberos ハッシュがない場合^[6]、そのユーザーはユーザーアカウントがあっても IdM ドメインにログインすることができません。パスワード移行にはパスワード変更の実施、web ページの使用、SSSD の使用の 3 通りの方法があります。

既存システムからユーザーを移行すると遷移プロセスはスムーズですが、移行と遷移期間を通じて LDAP ディレクトリーおよび IdM を平行管理する必要があります。パスワードを維持しない場合は、移行はより迅速に行うことができますが管理者およびユーザーによる手作業が多く必要になります。

37.1.2.1. 方法 1: 一時的なパスワードの使用とパスワード変更の強制

Identity Management でパスワードを変更すると、適切な Kerberos ハッシュも作成されます。このため方法の 1 つとしてユーザーアカウントの移行時にすべてのユーザーパスワードをリセットしてユーザーにパスワードの変更を強制する方法があります。新規ユーザーには一時的なパスワードが割り当てられ、初回のログインで変更することになります。パスワードの移行はありません。

詳細情報は「[ユーザーパスワードの変更およびリセット](#)」を参照してください。

37.1.2.2. 方法 2: 移行用 Web ページの使用

移行モードで実行している場合は Identity Management の web UI 内に特殊な web ページが用意されています。このページを使用するとクリアテキストのパスワードのキャプチャと適切な Kerberos ハッシュの作成が行われます。

`https://ipaserver.example.com/ipa/migration`

管理者はユーザーに対して上記の web ページで一度だけ認証を行うよう通知します。これによりユーザーのアカウントがユーザーのパスワードと Kerberos ハッシュで正しく更新されます。パスワードの変更は必要ありません。

37.1.2.3. 方法 3: SSSD の使用 (推奨)

SSSD は IdM と連携し必要なユーザーキーを生成することで移行の際にユーザーに与える影響を軽減することができます。大量のユーザーを導入する場合やユーザーにパスワード変更の面倒をかけさせない場合に最適なシナリオです。

1. ユーザーが SSSD でマシンにログインします。
2. SSSD は Kerberos 認証を IdM サーバーに対して試行します。
3. ユーザーがシステムに存在しても Kerberos ハッシュがないため **key type is not supported** エラーで認証に失敗します。
4. SSSD は次に安全な接続でプレーンテキストの LDAP バインドを行います。
5. IdM によってこのバインド要求が遮断されます。ユーザーが Kerberos プリンシパルは持っているのに Kerberos ハッシュを持っていない場合、IdM ID プロバイダーはハッシュを生成してユーザーのエントリに格納します。
6. 認証に成功すると SSSD は IdM との接続を切断し Kerberos 認証を再試行します。この場合、エントリにハッシュが存在しているため要求は成功します。

プロセス全体がユーザーに対しては透過的に行われるので、ユーザーは単純にクライアントサービスにログインし、通常通りに動作したということしかわかりません。

37.1.2.4. クリアテキスト LDAP パスワードの移行

ほとんどのデプロイメントでは暗号化された LDAP パスワードが格納されますが、ユーザーまたは環境によってユーザーエントリにクリアテキストのパスワードが使用される場合があります。

ユーザーを LDAP サーバーから IdM サーバーに移行する場合にはクリアテキストのパスワードは移行されません。Identity Management はクリアテキストのパスワードを許可していません。代わりにユーザーには Kerberos プリンシパルが作成され、keytab が true に設定されるためパスワードは有効期限切れとして設定されます。つまり、次のログインでユーザーによるパスワードのリセットが必要になります。



注記

パスワードがハッシュ化されると「[方法 2: 移行用 Web ページの使用](#)」と「[方法 3: SSSD の使用 \(推奨\)](#)」と同様に SSSD および移行用 web ページからの移行に成功します。

37.1.2.5. 要件を満たしていないパスワードの自動リセット

オリジナルのディレクトリーにあるユーザーパスワードが Identity Management で定義されているパスワードポリシーに合わない場合は移行後にパスワードのリセットが必要になります。

パスワードのリセットはユーザーが **kinit** をはじめて IdM ドメインで発行したときに自動的行われます。

```
[jsmith@server ~]$ kinit
Password for jsmith@EXAMPLE.COM:
Password expired. You must change it now.
Enter new password:
Enter it again:
```

37.1.3. 移行における考慮事項と要件

LDAP から Identity Management に移行を計画している場合には、使用する LDAP 環境が Identity Management の移行スクリプトで正しく動作できることを確認してください。

37.1.3.1. 移行に対応している LDAP サーバー

LDAP から Identity Management への移行プロセスで移行を行う際に特殊なスクリプト **ipa migrate-ds** が使用されます。このスクリプトは正しく動作するため LDAP ディレクトリーおよび LDAP エントリーに一定の構造を期待します。移行に対応しているのは複数の共通ディレクトリーを含む LDAPv3 準拠のディレクトリーサービスのみになります。

- Sun ONE Directory Server
- Apache Directory Server
- OpenLDAP

LDAP サーバーから Identity Management への移行は Red Hat Directory Server および OpenLDAP でテストが行われています。



注記

Microsoft Active Directory の場合、移行用スクリプトを使った移行には**対応していません**。Microsoft Active Directory は LDAPv3 準拠のディレクトリーではありません。Active Directory からの移行については Red Hat プロフェッショナルサービスにご連絡ください。

37.1.3.2. 移行環境に関する要件

Red Hat Directory Server と Identity Management はいずれもさまざまに異なる設定状況が考えられ、それぞれに移行プロセスに影響を与える可能性があります。本章で説明している移行例の場合、以下に示すような環境を想定しています。

- 1 つの LDAP ディレクトリードメインを 1 つの IdM レルムに移行します。統合はありません。
- ユーザーのパスワードは LDAP ディレクトリー内にハッシュで格納されています。サポートされているハッシュの一覧は、[『Red Hat Directory Server 10 Administration Guide』の「Password Policy Attributes」の表](#)に含まれる **passwordStorageScheme** 属性を参照してください。
- LDAP ディレクトリーインスタンスは ID 格納および認証方法の両方になります。クライアントのマシンは LDAP サーバーへの接続に **pam_ldap** または **nss_ldap** を使用するよう設定します。
- エントリーに使用するのは標準の LDAP スキーマのみです。カスタムオブジェクトクラスまたは属性に含まれるエントリーは Identity Management には移行されません。

37.1.3.3. 移行 — IdM システムの要件

中規模サイズのディレクトリーの場合 (ユーザー数 10,000、グループ数 10 程度)、移行を進めるには移行先のシステム (IdM システム) に十分な性能を必要とします。移行に最小限必要となる要件を以下に示します。

- 4 コア
- 4GB の RAM

- 30GB のディスク領域
- 2MB の SASL バッファ (IdM サーバーのデフォルト)

移行で問題が発生した場合には、バッファサイズを増やしてください。

```
[root@ipaserver ~]# ldapmodify -x -D 'cn=directory manager' -w
password -h ipaserver.example.com -p 389
```

```
dn: cn=config
changetype: modify
replace: nsslapd-sasl-max-buffer-size
nsslapd-sasl-max-buffer-size: 4194304
```

```
modifying entry "cn=config"
```

nsslapd-sasl-max-buffer-size の値をバイト単位で設定します。

37.1.3.4. 移行ツール

LDAP ディレクトリーのデータが正しくフォーマット化され、IdM サーバーに適切にインポートされるように Identity Management は **ipa migrate-ds** コマンドを使って移行プロセスを進めます。**ipa migrate-ds** を使用する場合、**--bind-dn** オプションで指定されたリモートのシステムユーザーには **userPassword** 属性に対する読み取りのアクセスを割り当てる必要があります。アクセス権がないとパスワードが移行されません。

Identity Management サーバーを移行モードで実行するよう設定する必要があります。移行スクリプトはこの設定を行うと使用できるようになります。詳細は、「[LDAP サーバーの Identity Management への移行](#)」を参照してください。

37.1.3.5. 移行のパフォーマンス改善

LDAP 移行とは基本的に IdM サーバー内の 389 Directory Server インスタンスに対する特殊なインポート動作になります。インポート動作のパフォーマンスがよくなるよう 389 Directory Server インスタンスをチューニングすると移行の全体的なパフォーマンスが向上します。

インポートのパフォーマンスに直接影響するパラメーターが 2 種類あります。

- **nsslapd-cachememsize** 属性、エントリーキャッシュに許可するサイズを定義します。キャッシュの合計メモリーサイズの 80% に自動的に設定されるバッファです。大規模なインポート動作の場合にはこのパラメーター (およびメモリーキャッシュ自体) を増やして多数のエントリーまたは大きい属性を持つエントリーをより効率的に処理できるようにします。

ldapmodify を使用して属性を変更する方法は、『[Red Hat Directory Server Performance Tuning Guide](#)』の[該当のセクション](#)を参照してください。

- システムの **ulimit** 設定オプションでは、システムユーザーに対して許可するプロセス数の最大値を設定します。大規模なデータベースを処理すると上限を超える可能性があります。このような場合には、値を増やしてください。

```
[root@server ~]# ulimit -u 4096
```

詳しい情報は、Red Hat Directory Server 『Performance Tuning Guide』のhttps://access.redhat.com/documentation/en-US/Red_Hat_Directory_Server/10/html-single/Performance_Tuning_Guide/index.htmlを参照してください。

37.1.3.6. 移行順序

Identity Management への移行は大きく分けて 4 ステップになります。ただし、サーバーを先に移行するのかクライアントを先に移行するのかによってこの順序は若干異なります。

クライアントを先に移行する場合は SSSD を使ってクライアント設定を変更し、IdM サーバーを設定します。

1. SSSD を導入します。
2. クライアントが現在の LDAP サーバーに接続し IdM にフェールオーバーするよう再設定を行います。
3. IdM サーバーをインストールします。
4. IdM の **ipa migrate-ds** スクリプトを使ってユーザーデータの移行を行います。これによりデータが LDAP ディレクトリーからエクスポートされ、IdM スキーマ用にフォーマット化されて IdM にインポートされます。
5. LDAP サーバーをオフラインにしてクライアントを透過的に Identity Management にフェールオーバーさせます。

サーバーを先に移行する場合は、LDAP から Identity Management への移行を最初に行います。

1. IdM サーバーをインストールします。
2. IdM の **ipa migrate-ds** スクリプトを使ってユーザーデータの移行を行います。これによりデータが LDAP ディレクトリーからエクスポートされ、IdM スキーマ用にフォーマット化されて IdM にインポートされます。
3. オプションです。SSSD を導入します。
4. クライアントが IdM に接続するよう再設定を行います。LDAP サーバーと単純に差し替えることはできません。IdM ディレクトリーツリー — およびユーザーエントリーの DN — は以前のディレクトリーツリーとは異なります。

クライアントの再設定は必要ですが、直ちに再設定を行う必要はありません。更新したクライアントは IdM サーバーをポイントし、他のクライアントは旧 LDAP ディレクトリーをポイントするためデータ移植後に適度なテストと移行段階を持たせることができます。



注記

LDAP ディレクトリーと IdM サーバーを長期に渡っては並行稼働させないでください。2 つのサービス間でユーザーデータの整合性が失われる危険を招くことになります。

どちらも一般的な移行手順になりますが、すべての環境では動作しない場合があります。実際の LDAP 環境を移行する前に、テスト用の LDAP 環境を設定して移行プロセスの検証を行ってください。

37.2. IPA MIGRATE-DS の使用例

データの移行は **ipa migrate-ds** コマンドで行います。一番単純な例では移行するディレクトリーの LDAP URL を取得し、共通デフォルト設定をもとにデータをエクスポートします。

```
ipa migrate-ds ldap://ldap.example.com:389
```

移行済みのエントリー

migrate-ds コマンドは、**posixAccount** オブジェクトクラスに必要な **gidNumber** 属性と、**person** オブジェクトクラスに必要な **sn** 属性を含むアカウントのみを移行します。

プロセスのカスタマイズ

ipa migrate-ds コマンドは、データを特定してエクスポートする方法をカスタマイズすることができます。元のディレクトリーツリーがユニークな構造である場合や、エントリー内のエントリーや属性を除外すべき場合に便利です。詳しい情報は、コマンドに **--help** を指定してください。

バインド DN

デフォルトでは、DN "**cn=Directory Manager**" を使用して、リモートの LDAP ディレクトリーにバインドします。**--bind-dn** オプションをコマンドに渡して、カスタムのバインド DN を指定します。詳細情報は、「[移行ツール](#)」を参照してください。

ネーミングコンテキストの変更

Directory Server のネーミングコンテキストが Identity Management で使用するものとは異なる場合には、オブジェクトのベース DNs は変換されます。たとえ

ば、**uid=user,ou=people,dc=ldap,dc=example,dc=com** は

uid=user,ou=people,dc=idm,dc=example,dc=com に移行します。**--base-dn** を **ipa**

migrate-ds コマンドに渡して、移行用にリモートの LDAP サーバーで使用するベース DN を設定します。

37.2.1. 特定のサブツリーの移行

デフォルトのディレクトリー構造の場合、人のエントリーは **ou=People** サブツリーに配置されグループのエントリーは **ou=Groups** サブツリーに配置されます。こうしたサブツリーは異なるタイプのディレクトリーデータ用のコンテナ・エントリーになります。**migrate-ds** に何もオプションを渡さないとユーティリティーは指定 LDAP ディレクトリーでは **ou=People** と **ou=Groups** の構造が使用されると仮定します。

多くのデプロイメントは完全に異なるディレクトリー構造をしている場合があります (またディレクトリーツリーの特定部分のみをエクスポートする場合もあります)。管理者は、2 種の方法で、ソースの LDAP サーバー上のグループサブツリーや異なるユーザーの RDN を指定できます。

- **--user-container**
- **--group-container**



注記

いずれの場合もサブツリーを RDN のみにしてベース DN に相対的にする必要があります。たとえば、**>ou=Employees,dc=example,dc=com** ディレクトリーツリーは **--user-container=ou=Employees** を使用して移行できます。

以下に例を示します。

```
[root@ipaserver ~]# ipa migrate-ds --user-container=ou=employees \
--group-container="ou=employee groups" \
ldap://ldap.example.com:389
```

--scope オプションを **ipa migrate-ds** コマンドに渡して、スコープを設定します。

- **onelevel**: デフォルト。指定のコンテナのエントリーのみが移行されます。
- **subtree**: 指定のコンテナおよびすべてのサブコンテナに含まれるエントリーが移行されます。
- **base**: 指定したオブジェクト自体が移行されます。

37.2.2. 特定のエントリーのみを包含および除外

デフォルトでは **ipa migrate-ds** スクリプトは、**person** オブジェクトクラスで全ユーザーエントリーを、**groupOfUniqueNames** または **groupOfNames** オブジェクトクラスで全グループエントリーをインポートします。

一部の移行パスでは特定のユーザータイプやグループタイプのみをエクスポートする必要がある場合、逆にエクスポートから除外する必要がある場合があります。

オプションの 1 つとして、追加するユーザーやグループの **タイプ** を設定する方法があります。これは、ユーザーまたはグループエントリーの検索時に特定するオブジェクトクラスを設定することで、タイプの設定が可能です。

異なるユーザータイプにカスタムのオブジェクトクラスが使用されている環境では非常に便利なオプションです。たとえば、このオプションによりカスタムの **fullTimeEmployee** オブジェクトクラスの付いたユーザーのみを移行します。

```
[root@ipaserver ~]# ipa migrate-ds --user-objectclass=fullTimeEmployee  
ldap://ldap.example.com:389
```

グループのタイプが異なる場合にも、特定のグループタイプのみを移行し、証明書グループなど他のグループタイプは除外することができ非常に便利なオプションになります。以下に例を示します。

```
[root@ipaserver ~]# ipa migrate-ds --group-objectclass=groupOfNames --  
group-objectclass=groupOfUniqueNames ldap://ldap.example.com:389
```

オブジェクトクラスに応じて移行するユーザーとグループを指定することは暗示的にそれ以外のユーザーおよびグループはすべて移行から除外するということになります。

また、ごく少数のエントリー以外、すべてのユーザーとグループのエントリーを移行する場合にも便利です。特定のユーザーまたはグループのアカウントを除外する一方、そのタイプの他のエントリーはすべて移行することができます。以下に趣味のグループと 2 人のユーザーを除外している例を示します。

```
[root@ipaserver ~]# ipa migrate-ds --exclude-groups="Golfers Group" --  
exclude-users=jsmith --exclude-users=bjensen ldap://ldap.example.com:389
```

除外のステートメントは、**uid** のパターンに一致するユーザーと、**cn** 属性に一致するグループに適用されます。

移行オブジェクトクラスの指定と特定エントリーの除外は併用することができます。たとえば、**fullTimeEmployee** オブジェクトクラスを持つユーザーを移行に含め 3 人のマネージャーは除外する例を以下に示します。

```
[root@ipaserver ~]# ipa migrate-ds --user-objectclass=fullTimeEmployee --  
exclude-users=jsmith --exclude-users=bjensen --exclude-users=mreynolds  
ldap://ldap.example.com:389
```

37.2.3. エントリー属性の除外

デフォルトではユーザーやグループエントリーのすべての属性とオブジェクトクラスが移行されますが、帯域幅やネットワーク上に制約があったり、属性データの適正性が失われた場合など現実に即しない場合があります。たとえば、IdM ドメインに参加させるためユーザーには新規のユーザー証明書を割り当てる予定の場合には **userCertificate** 属性を移行する意味はなくなります。

特定のオブジェクトクラスや属性を **migrate-ds** にいくつかのオプションを使って無視させることができます。

- **--user-ignore-objectclass**
- **--user-ignore-attribute**
- **--group-ignore-objectclass**
- **--group-ignore-attribute**

ユーザーの **userCertificate** 属性と **strongAuthenticationUser** オブジェクトクラスおよびグループの **groupOfCertificates** オブジェクトクラスを除外する例を以下に示します。

```
[root@ipaserver ~]# ipa migrate-ds --user-ignore-attribute=userCertificate
--user-ignore-objectclass=strongAuthenticationUser --group-ignore-
objectclass=groupOfCertificates ldap://ldap.example.com:389
```



注記

必要な属性が無視されていないか必ず確認します。また、オブジェクトクラスを除外する場合、そのオブジェクトクラスでしか対応しない属性はすべて除外するようにしてください。

37.2.4. 使用するスキーマの設定

Identity Management は RFC2307bis スキーマを使用して、ユーザー、ホスト、ホストグループ、ネットワーク ID を定義します。ただし、移行元として使用する LDAP サーバーが RFC2307 スキーマを使用する場合には、**--schema** オプションを **ipa migrate-ds** コマンドに渡します。

```
[root@ipaserver ~]# ipa migrate-ds --schema=RFC2307
ldap://ldap.example.com:389
```

37.3. LDAP サーバーの IDENTITY MANAGEMENT への移行



重要

この例は一般的な移行手順のため、あらゆる環境に対応するわけではありません。

実際に LDAP 環境の移行に入る前に、LDAP のテスト環境を設定して移行プロセスを検証することを強く推奨します。

1. カスタム LDAP ディレクトリースキーマなど、IdM サーバーを既存の LDAP ディレクトリーとは別のマシンにインストールします。



注記

IdM では、カスタムユーザーまたはグループスキーマのサポートに限りがあります。オブジェクトの定義に互換性がないので移行中に問題が発生する可能性があります。

2. compat プラグインを無効にします。

```
[root@server ~]# ipa-compat-manage disable
```

compat ツリーから提供されているデータが移行中に必要な場合には、このステップは省略可能です。

3. IdM Directory Server インスタンスを再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

4. IdM サーバーが移行を許可するように設定

```
[root@server ~]# ipa config-mod --enable-migration=TRUE
```

5. IdM 移行用スクリプト **ipa migrate-ds** を実行します。最も基本的な移行の場合、ここで必要となるのは LDAP ディレクトリーインスタンスの LDAP URL のみです。

```
[root@server ~]# ipa migrate-ds ldap://ldap.example.com:389
```

LDAP URL を渡すだけで共通のデフォルト設定を使用するディレクトリーデータはすべて移行されます。ユーザーやグループのデータは「[ipa migrate-ds の使用例](#)」で説明しているように他のオプションを指定することで選択的に移行することが可能です。

compat プラグインが以前のステップで無効化されている場合には、**--with-compat** オプションを **ipa migrate-ds** に渡します。

情報のエクスポートが完了すると、ネーミングコンテキストが異なる場合には、このスクリプトにより、必要とされる IdM オブジェクトクラスおよび属性がすべて追加され、IdM ディレクトリーツリーと一致するよう DN は属性に変換されます。たとえば、**uid=user,ou=people,dc=ldap,dc=example,dc=com** は **uid=user,ou=people,dc=idm,dc=example,dc=com** に移行されます。

6. 無効化されている場合には、移行前に compat プラグインを再度有効にします。

```
[root@server ~]# ipa-compat-manage enable
```

7. IdM Directory Server インスタンスを再起動します。

```
[root@server ~]# systemctl restart dirsrv.target
```

8. 移行モードを無効にします。

```
[root@server ~]# ipa config-mod --enable-migration=FALSE
```

9. オプションです。SSSD ではないクライアントが LDAP 認証 (**pam_ldap**) ではなく Kerberos

認証 (**pam_krb5**) を使用するよう再設定します。全ユーザーが移行されるまで **PAM_LDAP** モジュールを使用し、次に **PAM_KRB5** をしようできるようになります。詳しい情報は、『[システムレベルの認証ガイド](#)』の適切なセクションを参照してください。

10. ハッシュ化された Kerberos パスワードを生成するには、2 種類の方法があります。「[パスワード移行のプランニング](#)」に記載されているように、他にユーザーの対話なしにユーザーのパスワードを両方移行します。

1. SSSD の使用:

1. SSSD をインストールしているクライアントを LDAP バックエンドから IdM バックエンドに移動し、IdM でクライアントとして登録します。これにより必要なキーと証明書がダウンロードされます。

Red Hat Enterprise Linux クライアントで、**ipa-client-install** コマンドを使用して実行できます。以下に例を示します。

```
[root@server ~]# ipa-client-install --enable-dns-update
```

2. IdM の移行 Web ページの使用

1. 以下の移行 Web ページを使用して IdM にログインするように指示を出します。

```
https://ipaserver.example.com/ipa/migration
```

11. ユーザーの移行プロセスを監視するには、パスワードは持っているが Kerberos プリンシパルキーはまだないユーザーアカウントを表示するよう既存の LDAP ディレクトリーに問い合わせます。

```
[user@server ~]$ ldapsearch -LL -x -D 'cn=Directory Manager' -w  
secret -b 'cn=users,cn=accounts,dc=example,dc=com' '(&(!  
(krbprincipalkey=*)))(userpassword=*))' uid
```



注記

フィルターの前後に単一引用符を付けてシェルで解釈されないようにします。

12. クライアントとユーザーすべての移行が完了したら LDAP ディレクトリーを廃止します。

37.4. SSL での移行

移行中の LDAP および IdM のデータ変換を暗号化するには以下を行います。

1. リモートの LDAP サーバーの証明書を発行する CA の証明書を IdM サーバーのファイルに保存します (例: **/etc/ipa/remote.crt**)。
2. 「[LDAP サーバーの Identity Management への移行](#)」に記載の手順に従います。ただし、移行中の暗号化された LDAP 接続の場合には、以下のように URL で **ldaps** のプロトコルを使用して、コマンドに **--ca-cert-file** オプションを渡します。

```
[root@ipaserver ~]# ipa migrate-ds --ca-cert-  
file=/etc/ipa/remote.crt ldaps://ldap.example.com:636
```

[6] Kerberos 認証の代わりに Identity Management では LDAP を使用することができます。この場合ユーザーに Kerberos ハッシュは必要ありません。ただし、これを行うと Identity Management の機能が制限されるため推奨していません。

付録A トラブルシューティングのガイドライン

このセクションでは、ログやサービスステータスにクエリーするなどの、問題の根本原因を判定するための一般的なステップについて説明します。



注記

特定の問題およびその解決方法については、[付録B トラブルシューティング: 特定問題の解決](#)を参照してください。

以下の状況でのトラブルシューティングを解説します。

- [ipa](#) ユーティリティーを使用したコマンド実行時
- [kinit](#) を使用した認証時
- IdM web UI への認証時
- スマートカードでの認証時
- サービスの起動時

IdM の以下のエリアで問題が発生している場合は、各リンク先を参照してください。

- [DNS](#)
- [レプリケーション](#)

本ガイドで問題が解決できず、サポートケースを開かれる場合は、トラブルシューティングの手順で判明したエラー出力をケースレポートに含めてください。[Red Hat テクニカルサポートへのお問い合わせ](#)も参照してください。

A.1. IPA ユーティリティー実行時のエラー

基本的なトラブルシューティング

1. コマンドに **--verbose (-v)** オプションを追加して、デバッグ情報を表示します。
2. コマンドに **-vv** オプションを追加して、JSON 応答およびリクエストを表示します。

高度なトラブルシューティング

[図A.1 「ipa cert-show コマンドを実行するアーキテクチャー」](#) では、ユーザーが IdM コマンドラインユーティリティーを使用する際にどのコンポーネントと対話するかを示しています。これらのコンポーネントにクエリーを実行すると、問題の発生場所と原因を判定する助けとなります。

1. 以下のユーティリティーを使用します。
 - **host** は、IdM サーバーまたはクライアントの DNS 解決をチェックします。
 - **ping** は、IdM サーバーが利用可能かどうかをチェックします。
 - **iptables** は、IdM サーバーの現行のファイアウォール設定をチェックします。
 - **date** は現在の時間をチェックします。
 - **nc** は、「[ポート要件](#)」にある必須ポートへの接続を試行します。

これらのユーティリティーについての詳細は、各コマンドの man ページを参照してください。

2. **KRB5_TRACE** 環境変数を **/dev/stdout** ファイルに設定し、トレースログ出力を **/dev/stdout** に送信します。

```
$ KRB5_TRACE=/dev/stdout ipa cert-find
```

Kerberos キー配布センター (KDC) のログを **/var/log/krb5kdc.log** でチェックします。

3. Apache エラーログをチェックします。
 - a. サーバーのデバッグレベルを有効にします。**/etc/ipa/server.conf** ファイルを開いて、**debug=True** オプションを **[global]** セクションに追加します。
 - b. **httpd** サービスを再起動します。

```
# systemctl restart httpd.service
```

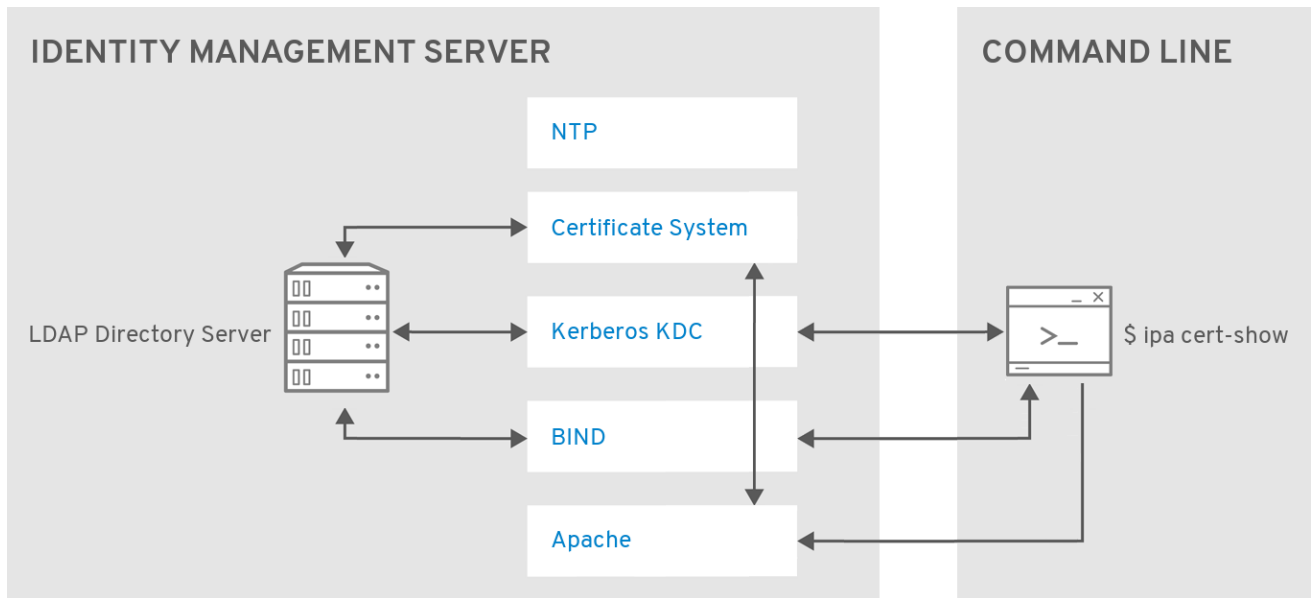
- c. 失敗したコマンドを再度実行します。
 - d. **httpd** エラーログをサーバーの **/var/log/httpd/error_log** でチェックします。
- vvv** オプションを追加してコマンドを実行し、HTTP リクエストと応答を表示します。

4. Apache アクセスログを **/var/log/httpd/access_log** でチェックします。

認証システムコンポーネントのログをチェックします。

- **/var/log/pki/pki-ca-spawn.time_of_installation.log**
- **/var/log/pki/pki-tomcat/ca/debug**
- **/var/log/pki/pki-tomcat/ca/system**
- **/var/log/pki/pki-tomcat/ca/selftests.log**
- **# journalctl -u pki-tomcatd@pki-tomcat.service** コマンドを使用して **journal** ログをレビューします。

5. Directory Server アクセスログをチェックします: **/var/log/dirsrv/slapd-IPA-EXAMPLE-COM/access**



図A.1 ipa cert-show コマンドを実行するアーキテクチャー

関連情報

- さまざまな Identity Management のログファイルに関する説明は、「[Identity Management ログファイルおよびディレクトリー](#)」を参照してください。

A.2. KINIT 認証エラー

全般的トラブルシューティング

1. IdM クライアント上で、**kinit** プロセス空のデバッグメッセージを表示します。

```
$ KRB5_TRACE=/dev/stdout kinit admin
```

2. 以下の点を確認します。

- 。クライアント転送レコードが、サーバーと影響されるクライアントの両方で正常であること。

```
# host client_fully_qualified_domain_name
```

- 。サーバー転送レコードが、サーバーと影響されるクライアントの両方で正常であること。

```
# host server_fully_qualified_domain_name
```

```
# host server_IP_address
```

host server_IP_address は、以下のように完全修飾ホスト名の最後にピリオドが付いたものを返す必要があります。

```
server.example.com.
```

3. クライアント上の **/etc/hosts** ファイルをチェックして、以下を確認します。

- 。ファイル内の全サーバーエントリーが正しいこと。

- 。全サーバーエントリーで、名前が完全修飾ドメイン名であること。

「[/etc/hosts ファイル](#)」も参照してください。

4. 「[ホスト名および DNS の設定](#)」にある他の条件も満たしていることを確認します。
5. IdM サーバー上で、**krb5kdc** と **dirsrv** のサービスが稼働中であることを確認します。

```
# systemctl status krb5kdc
# systemctl status dirsrv.target
```

6. Kerberos キー配布センター (KDC) のログを **/var/log/krb5kdc.log** でチェックします。
7. KDC が **/etc/krb5.conf** ファイル (このファイルは KDC ディレクティブを明示的に設定し、**dns_lookup_kdc = false** 設定を使用します) がハードコーディングされている場合は、各マスターサーバーで **ipactl status** コマンドを使用します。このコマンドで KDC として一覧表示される各サーバーの IdM サービスのステータスをチェックします。

```
# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
ntpd Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

Cannot find KDC for realm エラーのトラブルシューティング

kinit 認証で **Cannot find KDC for realm "EXAMPLE.COM" while getting initial credentials** というエラーが出て失敗する場合は、KDC がサーバーで稼働していないか、クライアントによる DNS の設定が正しくないことを示します。この場合は、以下のステップを実行します。

1. DNS 検出が **/etc/krb5.conf** ファイルで有効になっている場合 (**dns_lookup_kdc = true** の設定)、**dig** ユーティリティを使用して以下のレコードが解決可能かどうかをチェックします。

```
$ dig -t TXT _kerberos.ipa.example.com
$ dig -t SRV _kerberos._udp.ipa.example.com
$ dig -t SRV _kerberos._tcp.ipa.example.com
```

以下は、上記の **dig** コマンドの 1 つが失敗した場合に返す出力の例です。

```
; <<>> DiG 9.11.0-P2-RedHat-9.11.0-6.P2.fc25 <<>> -t SRV
_kerberos._tcp.ipa.server.example
;; global options: +cmd
;; connection timed out; no servers could be reached
```

この出力は、**named** サービスがマスターサーバー上で稼働していないことを示します。

2. DNS ルックアップが失敗する場合は、「[DNS のトラブルシューティング](#)」にあるステップを試してください。

関連情報

- さまざまな Identity Management のログファイルに関する説明は、[「Identity Management ログファイルおよびディレクトリー」](#)を参照してください。

A.3. IDM WEB UI での認証エラー

1. ユーザーが **kinit** ユーティリティーを使ってコマンドラインから認証できることを確認します。認証に失敗する場合は、[「kinit 認証エラー」](#)を参照してください。
2. **httpd** および **dirsrv** サービスが該当サーバー上で稼働していることを確認します。

```
# systemctl status httpd.service
# systemctl status dirsrv@IPA-EXAMPLE-COM.service
```

3. 関連した SELinux Access Vector Cache (AVC) メッセージが **/var/log/audit/audit.log** および **/var/log/messages** ファイルにないことを確認します。

AVC メッセージの解決については、Red Hat Knowledgebase の [Basic SELinux Troubleshooting in CLI](#) を参照してください。

4. 認証を行なっているブラウザのクッキーが有効になっていることを確認します。
5. IdM サーバーと認証を行なっているシステムの時間差が 5 分以内であることを確認します。
6. Apache エラーログをチェックします: **/var/log/httpd/error_log**
7. 問題の診断のために、認証プロセスの詳細なロギングを有効にします。Firefox で詳細なロギングを有効にする方法については、『システムレベルの認証ガイド』の [Firefox の Kerberos 設定のトラブルシューティング](#) を参照してください。

証明書を使用したログインで問題がある場合は、以下を実行します。

1. **/etc/httpd/conf.d/nss.conf** ファイルで、**LogLevel** の属性を **info** に変更します。
2. Apache サーバーを再起動します。

```
# systemctl restart httpd
```

3. 証明書を使って再度ログインします。
4. Apache エラーログをチェックします: **/var/log/httpd/error_log**

ログには **mod_lookup_identity** モジュールが記録したメッセージが表示されます。これには、ログイン試行時にモジュールがユーザーとの一致に成功したかどうかという情報が含まれています。

関連情報

- さまざまな Identity Management のログファイルに関する説明は、[「Identity Management ログファイルおよびディレクトリー」](#)を参照してください。

A.4. スマートカード認証の失敗

1. `/etc/sss/sss.conf` ファイルを開いて `debug_level` オプションを 2 に設定します。
2. `sss_pam.log` と `sss_EXAMPLE.COM.log` ファイルをチェックします。これらのファイルにタイムアウトのエラーメッセージがある場合は、「[スマートカード認証でタイムアウトエラーメッセージが出て失敗する問題](#)」を参照してください。

A.5. サービスが起動に失敗する理由の確認

1. 起動に失敗しているサービスのログをチェックします。「[Identity Management ログファイルおよびディレクトリー](#)」を参照してください。

たとえば、Directory Server のログは `/var/log/dirsrv/slapd-IPA-EXAMPLE-COM/errors` にあります。

2. サービスを稼働するサーバーに完全修飾ドメイン名 (FQDN) があることを確認します。「[サーバーのホスト名の検証](#)」を参照してください。
3. `/etc/hosts` ファイルにサービスを稼働するサーバーのエントリーが含まれている場合は、完全修飾ドメイン名が最初に記載されているかどうかを確認します。「[/etc/hosts ファイル](#)」を参照してください。
4. 「[ホスト名および DNS の設定](#)」にある他の条件も満たしていることを確認します。
5. サービスの認証に使用されるキータブにどのキーが含まれているか判定します。たとえば、`dirsrv` サービスのチケットは以下のようになります。

```
# klist -kt /etc/dirsrv/ds.keytab
Keytab name: FILE:/etc/dirsrv/ds.keytab
KVNO Timestamp Principal
-----
2 01/10/2017 14:54:39 ldap/server.example.com@EXAMPLE.COM
2 01/10/2017 14:54:39 ldap/server.example.com@EXAMPLE.COM
[... output truncated ...]
```

- a. 表示されているプリンシパルがシステムの FQDN に一致していることを確認します。
- b. 上記の表示されているサービスキータブ内のキーのバージョン (KVNO) がサーバーのキータブにある KVNO と一致していることを確認します。サーバーのキータブを表示するには、以下を実行します。

```
$ kinit admin
$ kvno ldap/server.example.com@EXAMPLE.COM
```

- c. クライアント上の正引き (A, AAAA, または両方) および逆引きレコードが表示されているシステム名およびサービスプリンシパルと一致することを確認します。
6. クライアント上の正引き (A, AAAA, または両方) および逆引きレコードが正しいことを確認します。
 7. クライアントとサーバーとのシステムの時間差が 5 分以内であることを確認します。
 8. IdM 管理サーバーの証明書の有効期限が切れると、サービスが起動に失敗することがあります。これが当てはまるかどうかを判定するには、以下を実行します。

- a. `getcert list` コマンドを使用して、`certmonger` ユーティリティーが追跡している証明

書をすべて一覧表示します。

- b. その出力で、IdM 管理証明書である **ldap** と **httpd** のサーバー証明書を見つけます。
- c. **status** と **expires** のフィールドをチェックします。

```
# getcert list
Number of certificates and requests being tracked: 8.
[... output truncated ...]
Request ID '20170421124617':
  status: MONITORING
  stuck: no
  key pair storage: type=NSSDB,location='/etc/dirsrv/slapd-IPA-EXAMPLE-COM',nickname='Server-Cert',token='NSS Certificate DB',pinfile='/etc/dirsrv/slapd-IPA-EXAMPLE-COM/pwdfilere.txt'
  certificate: type=NSSDB,location='/etc/dirsrv/slapd-IPA-EXAMPLE-COM',nickname='Server-Cert',token='NSS Certificate DB'
  CA: IPA
  issuer: CN=Certificate Authority,0=IPA.EXAMPLE.COM
  subject: CN=ipa.example.com,0=IPA.EXAMPLE.COM
  expires: 2019-04-22 12:46:17 UTC
[... output truncated ...]
Request ID '20170421130535':
  status: MONITORING
  stuck: no
  key pair storage:
  type=NSSDB,location='/etc/httpd/alias',nickname='Server-Cert',token='NSS Certificate DB',pinfile='/etc/httpd/alias/pwdfilere.txt'
  certificate:
  type=NSSDB,location='/etc/httpd/alias',nickname='Server-Cert',token='NSS Certificate DB'
  CA: IPA
  issuer: CN=Certificate Authority,0=IPA.EXAMPLE.COM
  subject: CN=ipa.example.com,0=IPA.EXAMPLE.COM
  expires: 2019-04-22 13:05:35 UTC
[... output truncated ...]
```

証明書の有効期限が切れていてもサービスを起動する必要がある場合は、[「IdM を有効期限の切れた証明書で起動できるようにする方法」](#) を参照してください。

A.6. DNS のトラブルシューティング

1. DNS 問題の多くは間違った設定のために発生するので、[「ホスト名および DNS の設定」](#) にある条件を満たしていることを確認してください。
2. **dig** ユーティリティを使って DNS サーバーからの応答をチェックします。

```
# dig _ldap._tcp.ipa.example.com. SRV

; <<>> DiG 9.9.4-RedHat-9.9.4-48.el7 <<>>
_ldap._tcp.ipa.example.com. SRV
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17851
```

```
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 1,
ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;_ldap._tcp.ipa.example.com. IN SRV

;; ANSWER SECTION:
_ldap._tcp.ipa.example.com. 86400 IN SRV      0 100 389
ipaserver.ipa.example.com.

;; AUTHORITY SECTION:
ipa.example.com.      86400 IN NS
ipaserver.ipa.example.com.

;; ADDITIONAL SECTION:
ipaserver.ipa.example.com. 86400 IN A 192.0.21
ipaserver.ipa.example.com 86400 IN AAAA 2001:db8::1
```

3. **host** 有効期限を使って DNS 名のルックアップを実行します。

```
$ host server.ipa.example.com
server.ipa.example.com. 86400 IN A 192.0.21
server.ipa.example.com 86400 IN AAAA 2001:db8::1
```

4. **ipa dnszone-show** コマンドを使って LDAP 内の DNS レコードを確認します。

```
$ ipa dnszone-show zone_name
$ ipa dnsrecord-show zone_name record_name_in_the_zone
```

IdM ツールを使った DNS 管理の詳細については、[32章DNS の管理](#) を参照してください。

5. BIND を再起動して LDAP との再同期を実行します。

```
$ systemctl restart named-pkcs11
```

6. 必要な DNS レコード一覧を取得します。

```
$ ipa dns-update-system-records --dry-run
```

dig ユーティリティを使って表示されているレコードが DNS に存在するかどうかをチェックします。Identity Management DNS を使用している場合は、**ipa dns-update-system-records** コマンドを使って見つからないレコードを更新します。

A.7. レプリケーションのトラブルシューティング

少なくとも 2 台のサーバーで複製をテストします (「[新規レプリカのテスト](#)」を参照)。一方の IdM サーバーでの変更がもう一方のサーバーに複製されない場合、以下の手順を実行します。

1. 「[ホスト名および DNS の設定](#)」にある条件を満たしていることを確認します。
2. 両方のサーバーで互いに正引きと逆引き DNS レコードを解決できることを確認します。

-

```
[root@server1 ~]# dig +short server2.example.com A
[root@server1 ~]# dig +short server2.example.com AAAA
[root@server1 ~]# dig +short -x server2_IPv4_or_IPv6_address
```

```
[root@server2 ~]# dig +short server1.example.com A
[root@server2 ~]# dig +short server1.example.com AAAA
[root@server2 ~]# dig +short -x server1_IPv4_or_IPv6_address
```

3. 両サーバー間の時間差が 5 分以内であることを確認します。
4. 両サーバーの Directory Server エラーログをチェックします:
/var/log/dirsrv/slapd-SERVER-EXAMPLE-COM/errors
5. Kerberos 関連のエラーがある場合は、Directory Server keytab が正しいものであることと、それを使ってもう一方のサーバー (この例では **server2**) にクエリーできることを確認します。

```
[root@server1 ~]# kinit -kt /etc/dirsrv/ds.keytab
ldap/server1.example.com
[root@server1 ~]# klist
[root@server1 ~]# ldapsearch -Y GSSAPI -h server1.example.com -b ""
-s base
[root@server1 ~]# ldapsearch -Y GSSAPI -h server2_FQDN. -b "" -s
base
```

関連情報

- さまざまな Identity Management のログファイルに関する説明は、[「Identity Management ログファイルおよびディレクトリー」](#)を参照してください。

付録B トラブルシューティング: 特定問題の解決

本付録では、以下の問題解決方法を説明しています。

- サーバーについては、[「Identity Management サーバー」](#)
- レプリカについては、[「Identity Management レプリカ」](#)
- クライアントについては、[「Identity Management クライアント」](#)
- 認証については、[「ログインと認証の問題」](#)

B.1. IDENTITY MANAGEMENT サーバー

B.1.1. 外部 CA インストールの失敗

`ipa-server-install --external-ca` コマンドを実行すると、以下のエラーが出て失敗します。

```
ipa          : CRITICAL failed to configure ca instance Command
'/usr/sbin/pkispawn -s CA -f /tmp/configuration_file' returned non-zero
exit status 1
Configuration of CA failed
```

`env|grep proxy` コマンドで、以下のような変数が表示されます。

```
env|grep proxy
http_proxy=http://example.com:8080
ftp_proxy=http://example.com:8080
https_proxy=http://example.com:8080
```

エラー内容:

*`_proxy` 環境変数がサーバーのインストールを妨げています。

解決方法:

1. 以下のシェルスクリプトを使用して `*_proxy` 環境変数の設定を解除します。

```
# for i in ftp http https; do unset ${i}_proxy; done
```

2. `pkidestroy` ユーティリティーを実行して、インストールに失敗した CA サブシステムを削除します。

```
# pkidestroy -s CA -i pki-tomcat; rm -rf /var/log/pki/pki-tomcat
/etc/sysconfig/pki-tomcat /etc/sysconfig/pki/tomcat/pki-tomcat
/var/lib/pki/pki-tomcat /etc/pki/pki-tomcat /root/ipa.csr
```

3. インストールに失敗した IdM サーバーを削除します。

```
# ipa-server-install --uninstall
```

4. `ipa-server-install --external-ca` を再実行します。

B.1.2. named デーモンの起動失敗

統合 DNS の IdM サーバーをインストールした後、**named-pkcs11** が起動に失敗します。`/var/log/messages` ファイルには、**named-pkcs11** サービスと **ldap.so** ライブラリーに関する以下のエラーメッセージが含まれます。

```
ipaserver named[6886]: failed to dynamically load driver 'ldap.so':
libldap-2.4.so.2: cannot open shared object file: No such file or
directory
```

エラー内容:

`bind-chroot` パッケージがインストールされているため **named-pkcs11** サービスの起動を妨げています。

解決方法:

1. `bind-chroot` パッケージをアンインストールします。

```
# yum remove bind-chroot
```

2. IdM サーバーを再起動します。

```
# ipactl restart
```

B.1.3. IPv6 を無効にしたシステムにおけるサーバーのインストールの失敗

IPv6 を無効にしたシステムで IdM サーバーをインストールしようとすると、インストールプロセス中に以下のエラーが発生します。

```
CRITICAL Failed to restart the directory server
Command '/bin/systemctl restart dirsrv@EXAMPLE.service' returned non-zero
exit status 1
```

エラー内容:

サーバーのインストールおよび稼働には、ネットワークで IPv6 が有効になっている必要があります。「[システム要件](#)」を参照してください。

解決方法:

IPv6 を有効にします。詳細情報は、Red Hat ナレッジベースの [Red Hat Enterprise Linux で IPv6 プロトコルを無効または有効にする](#) を参照してください。

Red Hat Enterprise Linux 7 システムではデフォルトで IPv6 が有効になることに留意してください。

B.2. IDENTITY MANAGEMENT レプリカ

ここでは、Red Hat Enterprise Linux の Identity Management によく発生するレプリカの問題について説明します。

その他のリソース:

- Red Hat Directory Server でのレプリカ問題に関する説明については、『Directory Server Administration Guide』の「[Solving Common Replication Conflicts](#)」セクションを参照してください。

- レプリケーションが機能しているかどうかをテストする方法については、「[新規レプリカのテスト](#)」を参照してください。
- Directory Server **repl-monitor** スクリプトはレプリケーションの進捗状況を表示するので、レプリカ問題のトラブルシューティングに役立ちます。このスクリプトについての詳細情報は、『Directory Server Administration Guide』の「[repl-monitor \(Monitors Replication Status\)](#)」セクションを参照してください。

B.2.1. AD ユーザーの新規レプリカに対する認証の失敗

Identity Management-Active Directory 信頼設定で新規レプリカをインストールした後、Active Directory (AD) ユーザーを IdM レプリカに対して認証しようとすると失敗します。

エラー内容:

レプリカは信頼コントローラーや信頼エージェントではないため、AD 信頼空の情報を処理できません。

解決方法:

レプリカを信頼エージェントとして設定します。詳細は、『[Windows Integration Guide](#)』の [Trust Controllers and Trust Agents](#) を参照してください。

B.2.2. レプリカを起動すると **Directory Server** ログに **SASL**、**GSS-API**、および **Kerberos** などのエラーが発生する問題

レプリカが起動すると、認証情報のキャッシュが見つからなかったため GSS-API 接続に失敗したことを示す一連の SASL bind エラーが Directory Server ログに記録されます。

```
slapd_ldap_sasl_interactive_bind - Error: could not perform interactive bind for id [] mech [GSSAPI]: error -2 (Local error) (SASL(-1): generic failure: GSSAPI Error: Unspecified GSS failure. Minor code may provide more information (Credentials cache file '/tmp/krb5cc_496' not found)) ...
```

また、サーバー側でホストプリンシパルの Kerberos 認証情報を取得できなかったことを示すメッセージが記録される場合もあります。

```
set_krb5_creds - Could not get initial credentials for principal [ldap/replica1.example.com] in keytab [WRFIL:/etc/dirsrv/ds.keytab]: -1765328324 (Generic error)
```

エラー内容:

IdM は Kerberos 接続に GSS-API を使用します。DS インスタンスはその Kerberos 認証情報のキャッシュをメモリーに格納します。(IdM レプリカが停止されるなど) DS プロセスが終了すると、認証情報キャッシュが破棄されます。

レプリカが再起動すると、DS は KDC サーバーが起動する前に起動します。この起動順序のために、DS が起動しても Kerberos 認証情報は認証情報キャッシュに保存されておらず、これが理由でエラーが発生します。

最初の失敗の後、DS は KDC の起動後に GSS-API 接続を再試行します。この再試行は成功し、レプリカは想定通りに機能します。

GSS-API 接続が正常に確立され、レプリカが想定通りに機能していれば、**上記の起動時のエラーは無視して構いません**。以下のメッセージは接続が成功したことを示します。

```
Replication bind with GSSAPI auth resumed
```

B.2.3. DNS の正引きレコードが逆引きアドレスと一致しない問題

新規のレプリカを設定する際、一連の証明書エラーが発生し、最終的には DNS の正引きレコードと逆引きアドレスが一致しない DNS エラーが発生します。

```
ipa: DEBUG: approved_usage = SSLServer intended_usage = SSLServer
ipa: DEBUG: cert valid True for "CN=replica.example.com,0=EXAMPLE.COM"
ipa: DEBUG: handshake complete, peer = 192.0.2.2:9444
Certificate operation cannot be completed: Unable to communicate with CMS
(Not Found)

...

ipa: DEBUG: Created connection context.ldap2_21534032
ipa: DEBUG: Destroyed connection context.ldap2_21534032
The DNS forward record replica.example.com. does not match the reverse
address replica.example.org
```

エラー内容:

1 つの PTR レコードに複数のホスト名を使用されています。DNS 標準では複数ホスト名の使用は可能ですが、IdM レプリカのインストールが失敗することになります。

解決方法:

「[正引きおよび逆引き DNS 設定の確認](#)」にあるように DNS 設定を確認します。

B.2.4. シリアル番号が見つからないエラー



注記

このソリューションは、ドメインレベル **0** に適用可能です。詳細は[7章 ドメインレベルの表示と引き上げ](#)を参照してください。

証明書のシリアル番号が見つからないというエラーがレプリカサーバーに表示されます。

```
Certificate operation cannot be completed: EXCEPTION (Certificate serial
number 0x2d not found)
```

エラー内容:

2 つのレプリカ間の証明書複製合意は削除されているものの、データの複製合意がまだ実行中となっています。レプリカは両方とも証明書を発行していますが、証明書についての情報が複製されなくなっています。

例:

1. レプリカ A がホストに証明書を発行します。
2. レプリカ A とレプリカ B の間には証明書複製合意がないため、この証明書はレプリカ B に複製されません。
3. ユーザーがレプリカ B を使ってホストを管理しようとします。

- レプリカ B が、ホストの証明書シリアル番号を確認できないというエラーを返します。これは、レプリカ B のデータディレクトリーにはホストに関する情報はああるものの、証明書ディレクトリーにはホストの証明書がないためです。

解決方法:

- ipa-csreplica-manage connect** コマンドを使って、2 つのレプリカ間の証明書サーバー複製を有効にします。command. See [「複製合意の作成と削除」](#) を参照してください。
- 一方のレプリカをもう一方のレプリカから再度初期化して、同期します。[「レプリカの再初期化」](#) を参照してください。



警告

再初期化を行ったレプリカのデータは、もう一方のレプリカのデータで上書きされ、情報が失われる可能性があります。

B.2.5. レプリカ更新ベクター (RUV) 消去のエラー



注記

このソリューションは、ドメインレベル **0** に適用可能です。詳細は[7章 ドメインレベルの表示と引き上げ](#)を参照してください。

IdM トポロジーからレプリカを削除した後も、古くなった RUV レコードが 1 つ以上の残りのレプリカに存在します。

考えられる原因:

- レプリカを削除する際に、[「複製合意の削除」](#) にあるように複製合意を最初に削除しなかったため。
- レプリカの削除時に別のレプリカがオフラインであったため。

エラー内容:

残りのレプリカは削除されたレプリカからの更新を想定し続けます。



注記

レプリカ削除の適切な手順は、[「レプリカの削除」](#) で説明しています。

解決方法:

更新を想定しているレプリカ上で、RUV レコードを消去します。

- ipa-replica-manage list-ruv** コマンドを使って古くなった RUV についての詳細を一覧表示します。レプリカの ID が表示されます。

```
# ipa-replica-manage list-ruv
server1.example.com:389: 6
```

```
server2.example.com:389: 5
server3.example.com:389: 4
server4.example.com:389: 12
```

2. **ipa-replica-manage clean-ruv *replica_ID*** コマンドを使って、破損している RUV を消去します。このコマンドは、指定されたレプリカに関連付けられている RUV を消去します。

古い RUV のあるレプリカについて、コマンドを繰り返します。

```
# ipa-replica-manage clean-ruv 6
# ipa-replica-manage clean-ruv 5
# ipa-replica-manage clean-ruv 4
# ipa-replica-manage clean-ruv 12
```



警告

ipa-replica-manage clean-ruv は慎重に使用してください。有効なレプリカ ID でこのコマンドを実行すると、複製データベースにあるそのレプリカに関連する前データが破損することになります。

その場合には、「[レプリカの再初期化](#)」にあるように別のレプリカから当該レプリカを再初期化してください。

3. **ipa-replica-manage list-ruv** を再実行します。

- コマンドを実行して破損した RUV が表示されなければ、レコードは正常に消去されています。
- まだ破損した RUV が表示される場合は、以下のタスクを使用して手動でこれを消去します。

```
dn: cn=clean replica_ID, cn=cleanallruv, cn=tasks, cn=config
objectclass: extensibleObject
replica-base-dn: dc=example,dc=com
replica-id: replica_ID
replica-force-cleaning: no
cn: clean replica_ID
```

どのレプリカの RUV を消去するか分からない場合は、以下の手順に従います。

1. お使いのサーバーでアクティブなレプリカの ID を見つけ、そこから破損していない信頼できるレプリカの ID リストを作成します。

有効なレプリカの ID を見つけるには、トポロジー内の全ノードに対して以下の LDAP クエリーを実行します。

```
# ldapsearch -p 389 -h IdM_node -D "cn=directory manager" -W -b
"cn=config" "(objectclass=nsds5replica)" nsDS5ReplicaId
```

2. 全サーバーで **ipa-replica-manage list-ruv** を実行します。破損していないレプリカの ID リストにないレプリカ ID を書き留めます。
3. 破損しているレプリカ ID すべてで **ipa-replica-manage clean-ruv *replica_ID*** を実行します。

B.2.6. 失われた CA サーバーの復旧



注記

このソリューションは、ドメインレベル **0** に適用可能です。詳細は [7章 ドメインレベルの表示と引き上げ](#) を参照してください。

CA が 1 つのサーバーにしかインストールされていない状態で、このサーバーが失敗して失われてしまいました。

エラー内容:

IdM ドメインの CA 設定が利用できません。

解決方法:

オリジナルの CA サーバーのバックアップがある場合は、サーバーを復旧してレプリカに CA をインストールすることが可能です。

1. CA サーバーをバックアップから復旧します。詳細は [「バックアップの復元」](#) を参照してください。

これで CA サーバーがレプリカで利用可能になります。

2. 元のサーバーとレプリカ間の複製合意を削除して、複製の競合を防ぎます。詳細は [「複製合意の作成と削除」](#) を参照してください。
3. CA をレプリカにインストールします。 [「レプリカのマスター CA サーバーへのプロモート」](#) を参照してください。
4. オリジナルの CA サーバーの使用を停止します。詳細は [「レプリカの削除」](#) を参照してください。

元の CA サーバーのバックアップがない場合は、サーバーが失敗すると CA 設定は失われ、復旧できなくなります。

B.3. IDENTITY MANAGEMENT クライアント

本セクションでは、Red Hat Enterprise Linux の IdM で一般的なクライアントの問題について説明します。

その他のリソース:

- **/etc/sss.conf** ファイルを検証する方法については、[『システムレベルの認証ガイド』](#) を参照してください。

B.3.1. 外部 DNS を使用すると逆引きルックアップをクライアントで解決できない問題

外部 DNS サーバーが IdM サーバーについて間違ったホスト名を返します。IdM サーバーに関する以下のエラーが Kerberos データベースに表示されます。

```
Jun 30 11:11:48 server1 krb5kdc[1279](info): AS_REQ (4 etypes {18 17 16 23}) 192.0.2.1: NEEDED_PREAUTH: admin EXAMPLE COM for krbtgt/EXAMPLE COM
EXAMPLE COM, Additional pre-authentication required
Jun 30 11:11:48 server1 krb5kdc[1279](info): AS_REQ (4 etypes {18 17 16 23}) 192.0.2.1: ISSUE: authtime 1309425108, etypes {rep=18 tkt=18 ses=18},
admin EXAMPLE COM for krbtgt/EXAMPLE COM EXAMPLE COM
Jun 30 11:11:49 server1 krb5kdc[1279](info): TGS_REQ (4 etypes {18 17 16 23}) 192.0.2.1: UNKNOWN_SERVER: authtime 0, admin EXAMPLE COM for
HTTP/server1.wrong.example.com@EXAMPLE.COM, Server not found in Kerberos
database
```

エラー内容:

外部 DNS ネームサーバーが IdM サーバーについて間違ったホスト名を返すか、応答しません。

解決方法:

1. お使いの DNS 設定を検証し、IdM が使用する DNS ドメインが適切に委任されていることを確認します。詳細は、「[ホスト名および DNS の設定](#)」を参照してください。
2. 逆引き (PTR) DNS レコード設定を確認します。詳細は「[32章DNS の管理](#)」を参照してください。

B.3.2. クライアントが DNS ゾーンに追加されない問題

ipa-client-install ユーティリティー実行時に、**nsupdate** ユーティリティーがクライアントを DNS ゾーンに追加することに失敗します。

エラー内容:

DNS 設定が正しくありません。

解決方法:

1. 親ゾーンから IdM への DNS 委任の設定を確認します。詳細は「[ホスト名および DNS の設定](#)」を参照してください。
2. IdM ゾーンでの動的更新が有効になっていることを確認します。詳細は「[動的 DNS 更新の有効化](#)」を参照してください。

IdM における DNS 管理の詳細については、「[逆引き DNS ゾーン管理](#)」を参照してください。Red Hat Enterprise Linux での DNS 管理の詳細については、『ネットワークガイド』の [11.2.3. 「ゾーンファイルの編集」](#) セクションを参照してください。

B.3.3. クライアント接続の問題

ユーザーがマシンにログインできません。ユーザーやグループ情報へのアクセス (**getent passwd admin** コマンドなど) に失敗します。

エラー内容:

クライアントの認証問題は、System Security Services Daemon (SSSD) サービスの問題であることが多くあります。

解決方法:

`/var/log/sss/` の SSSD ログをチェックします。このディレクトリーには、`sss_example.com.log` などの DNS ドメイン用のログファイルが格納されています。

ログに十分な情報がない場合は、ログのレベルを上げます。

1. `/etc/sss/sss.conf` ファイルで、`[domain/example.com]` セクションを探します。`debug_level` オプションを調整して、ログにより多くの情報を記録するようにします。

```
debug_level = 9
```

2. `sss` サービスを再起動します。

```
# systemctl start sssd
```

3. 再度 `sss_example.com.log` をチェックします。ファイルにはより多くのエラーメッセージが含まれているはずです。

B.4. ログインと認証の問題

B.4.1. ipa コマンド実行時の Kerberos GSS の失敗

サーバーのインストール直後に `ipa` コマンドを実行しようとする、以下のような Kerberos エラーが発生します。

```
ipa: ERROR: Kerberos error: ('Unspecified GSS failure. Minor code may provide more information', 851968)/('Decrypt integrity check failed', -1765328353)
```

エラー内容:

DNS が正しく設定されていません。

解決方法:

DNS 設定をチェックします。

- IdM サーバーの DNS 要件については、[「ホスト名および DNS の設定」](#) を参照してください。
- Active Directory 信頼の DNS 要件については、『Windows 統合ガイド』の [DNS およびレールム設定](#) を参照してください。

B.4.2. GSS-API 使用時の SSH 接続の失敗

SSH を使用したユーザーの IdM マシンへのログインが失敗します。

エラー内容:

SSH がセキュリティーに GSS-API を使用して IdM リソースに接続しようとする、GSS-API は最初に DNS レコードを検証します。SSH の失敗は、多くの場合、逆引き DNS エントリーが間違っていることが原因です。レコードが間違っていると、SSH は IdM リソースの場所を特定できません。

解決方法:

[「ホスト名および DNS の設定」](#) にあるように DNS 設定をチェックします。

一時的な回避策としては、SSH 設定内で逆引き DNS 検索を無効にすることもできます。これを実行するには、`/etc/ssh/ssh_config` ファイル内で **GSSAPITrustDNS** を **no** に設定します。逆引き DNS レコードを使用する代わりに、SSH は特定のユーザー名を GSS-API に直接渡します。

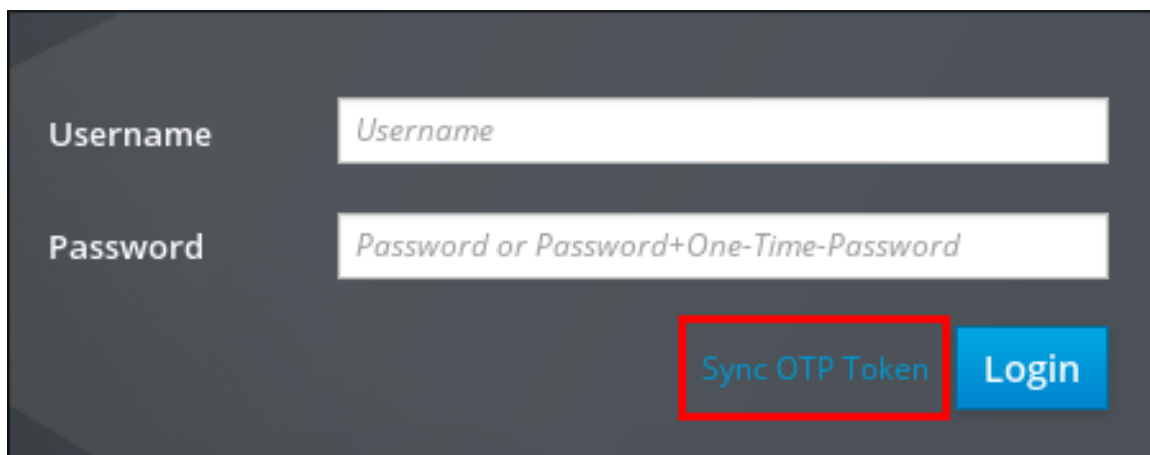
B.4.3. OTP トークンが同期されない問題

トークンが非同期なため、OTP を使った認証が失敗します。

解決方法:

トークンを再同期します。トークンはその種類や、ユーザーにトークン設定変更のパーミッションがあるかどうかに関わらず、だれでも再同期することができます。

1. IdM web UI では、ログインページの **Sync OTP Token** をクリックします。



図B.1 OTP トークンの同期

コマンドラインでは、**ipa otptoken-sync** コマンドを実行します。

2. トークンの再同期に必要な情報を提供します。たとえば、IdM は標準パスワードとトークンが生成する 2 つのトークンコードを要求します。



注記

再同期は、パスワードの有効期限が切れていても、実行できます。期限切れのパスワードを使用してトークンを再同期した後に IdM にログインすると、システムがパスワードの変更を要求します。

B.4.4. スマートカード認証でタイムアウトエラーメッセージが出て失敗する問題

sssd_pam.log と **sssd_EXAMPLE.COM.log** のファイルに以下のようなタイムアウトのエラーメッセージが含まれます。

```
Wed Jun 14 18:24:03 2017) [sssd[pam]] [child_handler_setup] (0x2000):
Setting up signal handler up for pid [12370]
(Wed Jun 14 18:24:03 2017) [sssd[pam]] [child_handler_setup] (0x2000):
Signal
handler set up for pid [12370]
(Wed Jun 14 18:24:08 2017) [sssd[pam]] [pam_initgr_cache_remove] (0x2000):
[idmeng] removed from PAM initgroup cache
(Wed Jun 14 18:24:13 2017) [sssd[pam]] [p11_child_timeout] (0x0020):
Timeout
```



```
reached for p11_child.  
(Wed Jun 14 18:24:13 2017) [sssd[pam]] [pam_forwarder_cert_cb] (0x0040):  
get_cert request failed.  
(Wed Jun 14 18:24:13 2017) [sssd[pam]] [pam_reply] (0x0200): pam_reply  
called  
with result [4]: System error.
```

エラー内容:

転送されたスマートカードリーダーやオンライン証明書状態プロトコル (OCSP) を使用する場合は、ユーザーがスマートカードを使って認証できるようにするために、特定のデフォルト値を調整する必要があります。

解決方法:

認証先のサーバーまたはクライアントで、**/etc/sss/sss.conf** ファイルに変更を加えます。

1. **[pam]** セクションの **p11_child_timeout** の値を 60 秒に増やします。
2. **[domain/EXAMPLE.COM]** セクションの **krb5_auth_timeout** の値を 60 秒に増やします。
3. 証明書で OCSP を使用している場合は、OCSP サーバーに到達できることを確認します。OCSP サーバーに直接到達できない場合は、プロキシ OCSP サーバー設定で **/etc/sss/sss.conf** に以下のオプションを追加します。

```
certificate_verification =  
ocsp_default_responder=http://ocsp.proxy.url,  
ocsp_default_responder_signing_cert=nickname
```

nickname を、**/etc/pki/nssdb/** ディレクトリー内の証明書に署名する OCSP のニックネームに置き換えます。

これらのオプションの詳細については、**sss.conf(5) man** ページを参照してください。

付録C IDENTITY MANAGEMENT ファイルおよびログのリファレンス

C.1. IDENTITY MANAGEMENT 設定ファイルおよびディレクトリー

表C.1 IdM サーバーおよびクライアント設定ファイルとディレクトリー

ディレクトリーまたはファイル	詳細
<code>/etc/ipa/</code>	IdM のメインの設定ディレクトリーです。
<code>/etc/ipa/default.conf</code>	IdM の主要な設定ファイル。サーバーやクライアントの起動時や、ユーザーが ipa ユーティリティーを使用する際に参照されます。
<code>/etc/ipa/server.conf</code>	オプションの設定ファイルはデフォルトでは存在しません。IdM サーバーが起動時に参照されます。 このファイルが存在する場合は、 /etc/ipa/default.conf よりも優先されます。
<code>/etc/ipa/cli.conf</code>	オプションの設定ファイルはデフォルトでは存在しません。ユーザーが ipa ユーティリティーを使用する場合に参照されます。 このファイルが存在する場合は、 /etc/ipa/default.conf よりも優先されます。
<code>/etc/ipa/ca.crt</code>	IdM サーバーの認証局で発行された認証局証明書です。
<code>~/.ipa/</code>	ユーザーが IdM コマンドを初回実行時に、ローカルシステムに作成されるユーザー固有の IdM ディレクトリー ユーザーは、 ~/.ipa/ に、ユーザー固有の default.conf 、 server.conf または cli.conf ファイルを作成することで、個別設定のオーバーライドを設定することができます。
<code>/etc/sss/sss.conf</code>	SSSD が使用する IdM ドメインや IdM サービスの設定
<code>/usr/share/sss/sss.api.d/sss-ipa.conf</code>	IdM 関連の SSSD オプションのスキーマおよびその値
<code>/etc/gssproxy/</code>	GSS プロキシプロトコルの設定用のディレクトリー。このディレクトリーには、各 GSS-API サービスのファイルと、一般的な /etc/gssproxy/gssproxy.conf ファイルが含まれます。

表C.2 システムサービスファイルとディレクトリー

ディレクトリーまたはファイル	詳細
/etc/sysconfig/	systemd 固有のファイル

表C.3 Web UI ファイルおよびディレクトリー

ディレクトリーまたはファイル	詳細
/etc/ipa/html/	IdM web UI が使用する HTML ファイルのシンボリックリンク
/etc/httpd/conf.d/ipa.conf	web UI アプリケーションの Apache ホストで使用する設定ファイル
/etc/httpd/conf.d/ipa-rewrite.conf	
/etc/httpd/conf/ipa.keytab	Web サーバーが使用する keytab ファイル
/usr/share/ipa/	web UI で使用される HTML ファイル、スクリプト、およびスタイルシートすべてのディレクトリー
/usr/share/ipa/ipa.conf	
/usr/share/ipa/updates/	IdM の LDAP データ、設定、スキーマの更新が含まれます。
/usr/share/ipa/html/	web UI で使用される HTML ファイル、JavaScript ファイル、およびスタイルシートが含まれます。
/usr/share/ipa/migration/	IdM サーバーを移行モードで実行する際に使用される HTML ページ、スタイルシート、および Python スクリプトが含まれます。
/usr/share/ipa/ui/	IdM 操作を実行するため UI で使用されるスクリプトが含まれます。
/etc/httpd/conf.d/ipa-pki-proxy.conf	Web サーバーと証明書システムのブリッジ用の設定

表C.4 Kerberos ファイルおよびディレクトリー

ディレクトリーまたはファイル	詳細
/etc/krb5.conf	Kerberos サービスの設定ファイル
/var/lib/sss/pubconf/krb5.include.d/	Kerberos クライアント設定用の IdM 固有のオーバーライドが含まれます。

表C.5 ディレクトリーサーバーのファイルおよびディレクトリー

ディレクトリーまたはファイル	詳細
<code>/var/lib/dirsrv/slapd-<i>REALM_NAME</i>/</code>	IdM サーバーで使用される Directory Server インスタンスに関連するデータベース
<code>/etc/sysconfig/dirsrv</code>	dirsrv systemd サービスの IdM 固有の設定
<code>/etc/dirsrv/slapd-<i>REALM_NAME</i>/</code>	IdM サーバーで使用される Directory Server インスタンスに関連する設定ファイルおよびスキーマファイル

表C.6 証明書システムのファイルとディレクトリー

ディレクトリーまたはファイル	詳細
<code>/etc/pki/pki-tomcat/ca/</code>	IdM 認証局インスタンスのメインのディレクトリー
<code>/var/lib/pki/pki-tomcat/conf/ca/CS.cfg</code>	IdM 認証局インスタンスのメインの設定ファイル

表C.7 キャッシュファイルとディレクトリー

ディレクトリーまたはファイル	詳細
<code>~/.cache/ipa/</code>	IdM クライアントのサーバー別の API スキーマが含まれます。IdM は、クライアント上の API スキーマを 1 時間キャッシュします。

表C.8 システムのバックアップファイルとディレクトリー

ディレクトリーまたはファイル	詳細
<code>/var/lib/ipa/sysrestore/</code>	IdM サーバーのインストール時に再設定されたスクリプトおよびシステムファイルのバックアップが格納されます。NSS、Kerberos (krb5.conf と kdc.conf の両ファイル)、および NTP のオリジナルの .conf ファイルなどが含まれます。
<code>/var/lib/ipa-client/sysrestore/</code>	IdM クライアントのインストール時に再設定されたスクリプトおよびシステムファイルのバックアップが格納されます。一般的には SSSD 認証サービスの sssd.conf ファイルなどが含まれます。

C.2. IDENTITY MANAGEMENT ログファイルおよびディレクトリー

表C.9 IdM サーバーおよびクライアントのログファイルおよびディレクトリー

ディレクトリーまたはファイル	詳細
<code>/var/log/ipaserver-install.log</code>	IdM サーバーのインストールログ
<code>/var/log/ipareplica-install.log</code>	IdM レプリカのインストールログ
<code>/var/log/ipaclient-install.log</code>	IdM クライアントのインストールログ
<code>/var/log/sss/</code>	SSSD のログファイル
<code>~/.ipa/log/cli.log</code>	XML-RPC 呼び出しで返されるエラーと ipa ユーティリティーの応答に関するログファイルです。ツールを実行する システムユーザー のホームディレクトリーに作成されます。IdM のユーザー名とは異なるユーザー名場合があります。
<code>/etc/logrotate.d/</code>	DNS、SSSD、Apache、Tomcat、および Kerberos のログローテーションのポリシー

表C.10 Apache サーバーのログファイル

ディレクトリーまたはファイル	詳細
<code>/var/log/httpd/</code>	Apache web サーバーのログファイル
<code>/var/log/httpd/access_log</code>	Apache サーバーの標準アクセスおよびエラーログ。IdM Web UI および XML-RPC コマンドラインのインターフェースが Apache を使用するため、IdM 固有のメッセージが Apache メッセージに合わせて記録されます。
<code>/var/log/httpd/error_log</code>	
詳細は、Apache ドキュメントの「 Log Files 」を参照してください。	

表C.11 証明書システムのログファイル

ディレクトリーまたはファイル	詳細
<code>/var/log/pki/pki-ca-spawn.time_of_installation.log</code>	IdM 認証局のインストールログ
<code>/var/log/pki/pki-kra-spawn.time_of_installation.log</code>	IdM KRA のインストールログ
<code>/var/log/pki/pki-tomcat/</code>	PKI の操作ログ用の最上位ディレクトリー。CA および KRA ログが含まれます。

ディレクトリーまたはファイル	詳細
<code>/var/log/pki/pki-tomcat/ca/</code>	証明書の操作関連のログを含むディレクトリー。 IdM ではサービスプリンシパル、ホスト、および証明書を使用するその他のエンティティーに使用されます。
<code>/var/log/pki/pki-tomcat/kra</code>	KRA 関連のログが含まれるディレクトリー
<code>/var/log/messages</code>	証明書のエラーメッセージなど、他のシステムメッセージが含まれます。
詳細は、Red Hat Certificate System 『Administration Guide』の「 Configuring Subsystem Logs 」を参照してください。	

表C.12 ディレクトリーサーバーのログファイル

ディレクトリーまたはファイル	詳細
<code>/var/log/dirsrv/slapd-<i>REALM_NAME</i>/</code>	IdM サーバーで使用する Directory Server インスタンスに関連するログファイル。ここに記録される操作データの大半は、サーバーレプリカの対話に関連します。
<code>/var/log/dirsrv/slapd-<i>REALM_NAME</i>/access</code>	ドメイン Directory Server インスタンスに対して試行されたアクセスおよび動作の詳細情報が含まれます。
<code>/var/log/dirsrv/slapd-<i>REALM_NAME</i>/errors</code>	
<code>/var/log/dirsrv/slapd-<i>REALM_NAME</i>/audit</code>	ディレクトリーサーバー設定で監査が有効になっている場合には、ディレクトリーサーバーの操作の監査証跡が含まれます。
詳細は、Red Hat Directory Server ドキュメントの「 Monitoring Server and Database Activity 」と「 Log File Reference 」を参照してください。	

表C.13 Kerberos ログファイル

ディレクトリーまたはファイル	詳細
<code>/var/log/krb5kdc.log</code>	Kerberos KDC サーバーのプライマリーログファイル
<code>/var/log/kadmind.log</code>	Kerberos 管理サーバーのプライマリーログファイル
これらのファイルの場所は <code>krb5.conf</code> ファイルで設定します。システムによっては場所が異なる場合があります。	

表C.14 DNS ログファイル

ディレクトリーまたはファイル	詳細
<code>/var/log/messages</code>	<p>DNS のエラーメッセージなど、他のシステムメッセージが含まれます。</p> <p>このファイルの DNS ロギングはデフォルトでは有効ではありません。有効化するには、# /usr/sbin/rndc querylog コマンドを実行し、ロギングを無効化するには、もう一度このコマンドを実行します。</p>

その他のリソース

- **journalctl** ユーティリティーの使用方法に関する情報は、『システム管理者のガイド』 [「Journal の使用」](#) を参照してください。**systemd** のユニットファイルのロギングの出力を表示するには **journalctl** を使用してください。

C.3. IDM ドメインサービスとログローテーション

一部の IdM ドメインサービスでは、ログローテーションおよび圧縮処理を行う場合にシステムの **logrotate** サービスを使用するものがあります。

- **named** (DNS)
- **httpd** (Apache)
- **tomcat**
- **sssd**
- **krb5kdc** (Kerberos ドメインコントローラー)

logrotate 設定ファイルは **/etc/logrotate.d/** ディレクトリーに格納されます。

例C.1 デフォルトの **httpd** のログローテーションファイルは **/etc/logrotate.d/httpd** にあります。

```
/var/log/httpd/*log {
    missingok
    notifempty
    sharedscripts
    delaycompress
    postrotate
        /sbin/service httpd reload > /dev/null 2>/dev/null || true
    endscript
}
```



警告

大半のサービスの **logrotate** ポリシーファイルでは、以前のログと同じ名前、デフォルトの所有者、デフォルトのパーミッションで新しいログファイル作成します。ただし、**named** および **tomcat** のファイルでは、特別な **create** ルールにより、明示的なパーミッション、ユーザー、グループの所有権でこの動作が設定されます。

named および **tomcat** ログファイルを所有するユーザーとグループまたはパーミッションは変更しないようにしてください。IdM の動作および SELinux 設定の両方に必要な値です。ログローテーションポリシーやファイルの所有権を変更すると IdM ドメインサービスが実行に失敗する可能性があります。

その他のリソース

- Dogtag Certificate System および IdM でバックエンドとして使用される 389 Directory Server インスタンスには独自の内部ログローテーションポリシーがあります。Red Hat Directory Server 『Administration Guide』の「[Configuring Log Files](#)」を参照してください。
- ログファイルの圧縮設定やサイズなど、その他に考えられるログローテーションの設定に関する詳細は、『システム管理者のガイド』の「[ログローテーション](#)」か、logrotate(8) の man ページを参照してください。

付録D ドメインレベル 0 でのレプリカの管理

本セクションでは、ドメインレベル 0 (7章 [ドメインレベルの表示と引き上げ](#) を参照) でのレプリカ管理について説明しています。ドメインレベル 1 でのレプリカ管理については、以下のリンクを参照してください。

- [「レプリカの作成: 概要」](#)
- [6章 レプリケーショントポロジーの管理](#)

D.1. レプリカ情報ファイル

レプリカ作成プロセス中に、**ipa-replica-prepare** ユーティリティは、`/var/lib/ipa/` ディレクトリー内にレプリカサーバーの名前を付けた**レプリカ情報ファイル**を作成します。このレプリカ情報ファイルは GPG 暗号化ファイルで、マスターサーバー用のレムおよび設定情報が含まれています。

ipa-replica-install レプリカ設定スクリプトは、レプリカ情報ファイルに含まれている情報を基に Directory Server インスタンスを設定し、**レプリカ初期化** プロセスを開始します。このプロセス中にスクリプトは、マスターサーバーからレプリカにデータをコピーします。レプリカ情報ファイルは、そのファイルが作成された特定マシン上のレプリカのインストールにのみ使用できます。複数マシン上の複数のレプリカ作成には使用できません。

D.2. レプリカの作成

以下のセクションでは、最も重要なレプリカインストールのシナリオについて説明します。

- これらの手順と例は相互排除的なものではなく、CA、DNS、および他のコマンドラインオプションは同時に使うことができます。以下のセクション例は、各設定エリアで必要なものを明確にするために個別に示されています。
- **ipa-replica-install** ユーティリティは多くのオプションを受け付けます。これらの完全一覧については、`ipa-replica-install(1) man` ページを参照してください。

D.2.1. DNS なしのレプリカのインストール

1. マスター IdM サーバー上で、**ipa-replica-prepare** ユーティリティを実行して、**レプリカ**の完全修飾ドメイン名 (FQDN) を追加します。レプリカの IP アドレスに他のサーバーが到達できないと、**ipa-replica-prepare** スクリプトはその IP アドレスの確認や検証を実行しないことに注意してください。



重要

完全修飾ドメイン名は有効な DNS 名である必要があります。つまり、許可されるのは数字、アルファベット、ハイフン (-) のみです。ホスト名にアンダースコアのような他の文字があると、DNS エラーが発生します。また、ホスト名はすべて小文字を使用する必要があります。大文字は使用できません。

命名プラクティスに関する他の推奨事項については、[Red Hat Enterprise Linux セキュリティガイド](#) を参照してください。

マスターサーバーが統合 DNS で設定されている場合は、**--ip-address** オプションを使ってレプリカマシンの IP アドレスを指定します。すると、インストールスクリプトはレプリカに逆引きゾーンを設定するかどうかを尋ねます。IdM サーバーが統合 DNS で設定されている場合

にのみ、**--ip-address** を渡します。これ以外の場合にこのオプションを渡すと、更新する DNS レコードが存在しないため、DNS レコード操作が失敗して、レプリカ作成も失敗することになります。

プロンプトが出たら、最初のサーバーの Directory Manager (DM) パスワードを入力します。**ipa-replica-prepare** の出力では、以下のようにレプリカ情報ファイルの場所が示されます。

```
[root@server ~]# ipa-replica-prepare replica.example.com --ip-
address 192.0.2.2
Directory Manager (existing master) password:

Do you want to configure the reverse zone? [yes]: no
Preparing replica for replica.example.com from server.example.com
Creating SSL certificate for the Directory Server
Creating SSL certificate for the dogtag Directory Server
Saving dogtag Directory Server port
Creating SSL certificate for the Web Server
Exporting RA certificate
Copying additional files
Finalizing configuration
Packaging replica information into /var/lib/ipa/replica-info-
replica.example.com.gpg
Adding DNS records for replica.example.com
Waiting for replica.example.com. A or AAAA record to be resolvable
This can be safely interrupted (Ctrl+C)
The ipa-replica-prepare command was successful
```



警告

レプリカ情報ファイルには機密情報が含まれています。適切な措置を講じてこの情報を保護してください。

ipa-replica-prepare で使用可能な他のオプションについては、ipa-replica-prepare(1) man ページを参照してください。

2. レプリカマシン上で、ipa-server パッケージをインストールします。

```
[root@replica ~]# yum install ipa-server
```

3. 最初のサーバーからレプリカマシンに、レプリカ情報ファイルをコピーします。

```
[root@server ~]# scp /var/lib/ipa/replica-info-
replica.example.com.gpg root@replica:/var/lib/ipa/
```

4. レプリカマシン上で、**ipa-replica-install** ユーティリティーを実行してレプリカ情報ファイルの場所を追加し、レプリカ初期化プロセスを開始します。プロンプトが出たら、オリジナルのマスターサーバーの Directory Manager および管理者パスワードを入力し、インストールスクリプトが完了するまで待機します。

```
[root@replica ~]# ipa-replica-install /var/lib/ipa/replica-info-
replica.example.com.gpg
Directory Manager (existing master) password:

Run connection check to master
Check connection from replica to remote master 'server.example.com':

...

Connection from replica to master is OK.
Start listening on required ports for remote master check
Get credentials to log in to remote master
admin@MASTER.EXAMPLE.COM password:

Check SSH connection to remote master

...

Connection from master to replica is OK.

...

Configuring NTP daemon (ntpd)
[1/4]: stopping ntpd
[2/4]: writing configuration

...

Restarting Directory server to apply updates
[1/2]: stopping directory server
[2/2]: starting directory server
Done.
Restarting the directory server
Restarting the KDC
Restarting the web server
```



注記

インストールされているレプリカファイルが現行のホスト名と一致しない場合は、スクリプトは警告メッセージを表示し、確認を求めます。マルチホームのマシンなどの場合には、ホスト名が一致しない場合でも継続できることがあります。

ipa-replica-install で使用可能な他のオプションについては、**ipa-replica-prepare(1)** man ページを参照してください。**ipa-replica-install** が受け付けるオプションの 1 つに **--ip-address** があります。これを **ipa-replica-install** に追加する場合は、**--ip-address** はローカルインターフェイスに関連付けられた IP アドレスのみを受け付けます。

D.2.2. DNS ありのレプリカのインストール

統合 DNS のあるレプリカをインストールする方法については、「[DNS なしのレプリカのインストール](#)」にある DNS なしでのインストールの手順と同じですが、以下のオプションを **ipa-replica-install** に追加します。

- **--setup-dns**
- **--forwarder**

詳細は、「[DNS ありのレプリカのインストール](#)」を参照してください。

以下に例を示します。

```
[root@replica ~]# ipa-replica-install /var/lib/ipa/replica-info-
replica.example.com.gpg --setup-dns --forwarder 198.51.100.0
```

ipa-replica-install を実行した後に、DNS エントリーが正常に作成されたことを確認します。またオプションで、別の DNS サーバーをバックアップとして追加することもできます。詳細は、「[DNS ありのレプリカのインストール](#)」を参照してください。

D.2.3. 各種 CA 設定を伴うレプリカのインストール



警告

Red Hat では、複数のサーバーに CA サービスをインストールしておくことを強く推奨しています。CA サービスを含む最初のサーバーのレプリカをインストールする方法についての情報は、「[CA を設定したレプリカのインストール](#)」を参照してください。

CA が 1 つのサーバーにしかインストールされていないと、CA サーバーが故障した際に CA 設定が失われて回復できない恐れがあります。詳細については、「[失われた CA サーバーの復旧](#)」を参照してください。

Certificate System CA がインストールされたサーバーからレプリカをインストールする初期サーバーが Red Hat Certificate System インスタンスで設定されている場合 (root CA もしくは外部 CA に従属しているかに関わらず) にレプリカ上で CA を設定するには、「[DNS なしのレプリカのインストール](#)」に記載の基本的なインストール手順に従いますが、さらに **--setup-ca** オプションを **ipa-replica-install** ユーティリティに追加します。この **--setup-ca** オプションは、CA 設定を初期サーバーの設定からコピーします。

```
[root@replica ~]# ipa-replica-install /var/lib/ipa/replica-info-
replica.example.com.gpg --setup-ca
```

Certificate System CA がインストールされていないサーバーからレプリカをインストールする

CA のないレプリカのインストールについては、「[DNS なしのレプリカのインストール](#)」にあるインストールの手順と同じですが、初期サーバーで **ipa-replica-prepare** を実行する際に以下のオプションを追加します。

- **--dirsrv-cert-file**
- **--dirsrv-pin**
- **--http-cert-file**

- **--http-pin**

詳細は「[CA がインストールされていないサーバーからのレプリカのインストール](#)」を参照してください。

以下に例を示します。

```
[root@server ~]# ipa-replica-prepare replica.example.com --dirsrv-cert-file /tmp/server.key --dirsrv-pin secret --http-cert-file /tmp/server.crt --http-cert-file /tmp/server.key --http-pin secret --dirsrv-cert-file /tmp/server.crt
```

D.2.4. 新たなレプリカ合意の追加

ipa-replica-install を使ってレプリカをインストールすると、初期のレプリカ合意がマスターサーバーとレプリカ間で作成されます。レプリカを他のサーバーや他のレプリカと接続するには、**ipa-replica-manage** ユーティリティで新たな合意を追加します。

マスターサーバーと新規レプリカに CA がインストールされている場合は、CA のレプリカ合意も作成されます。他のサーバーやレプリカに新たな CA レプリカ合意を追加するには、**ipa-csreplica-manage** ユーティリティを使用します。

新たなレプリカ合意を追加する方法については、「[レプリカとレプリカ合意の管理](#)」を参照してください。

D.3. レプリカとレプリカ合意の管理

本セクションでは、レプリカ合意とその管理方法について説明します。



注記

新たなレプリカ合意をセットアップするガイドラインについては、「[レプリカトポロジーの推奨事項](#)」を参照してください。

D.3.1. レプリカ合意についての説明

レプリカ合意は、参加するレプリカ間でのデータのコピーです。レプリカ合意は双方向のものです。1 台目のレプリカから 2 台目のレプリカにデータが複製されるほかに、2 台目のレプリカから 1 台目のレプリカにもデータが複製されます。



注記

初期のレプリカ合意は、**ipa-replica-install** スクリプトが 2 つのレプリカ間にセットアップします。最初のレプリカのインストールについては、[4 章 Identity Management のレプリカのインストールとアンインストール](#)を参照してください。

レプリカ合意のタイプ

Identity Management は、以下の 3 つのタイプのレプリカ合意をサポートしています。

- ユーザー、グループ、およびポリシーなどのディレクトリーデータを複製するレプリカ合意。これらの合意は、**ipa-replica-manage** ユーティリティで管理します。

- 証明書サーバーデータを複製するレプリカ合意。これらの合意は、**ipa-csreplica-manage** ユーティリティで管理します。
- Active Directory サーバーとユーザー情報を複製する同期合意。この合意については、本書では説明していません。IdM と Active Directory との同期に関するドキュメントについては、[Windows 統合ガイド](#) を参照してください。

ipa-replica-manage と **ipa-csreplica-manage** では、同じ形式と引数を使用します。以下のセクションでは、これらのユーティリティを使用して実行するレプリカ管理のうち、特に重要な操作を取り上げます。これらのユーティリティについての詳細情報は、ipa-replica-manage(1) と ipa-csreplica-manage(1) の man ページを参照してください。

D.3.2. レプリカ合意の一覧表示

あるレプリカで現在設定されているディレクトリーデータのレプリカ合意を一覧表示するには、**ipa-replica-manage list** コマンドを使用します。

1. **ipa-replica-manage list** を引数なしで実行すると、レプリカトポロジー内の全レプリカが一覧表示されます。出力で、必要なレプリカを見つけます。

```
$ ipa-replica-manage list
server1.example.com: master
server2.example.com: master
server3.example.com: master
server4.example.com: master
```

2. レプリカのホスト名を **ipa-replica-manage list** に追加して実行すると、レプリカ合意が一覧表示されます。

```
$ ipa-replica-manage list server1.example.com
server2.example.com: replica
server3.example.com: replica
```

この出力では、**server1.example.com** が更新を送信する宛先が表示されています。

証明書サーバーのレプリカ合意を一覧表示するには **ipa-csreplica-manage list** コマンドを使用します。

D.3.3. 複製合意の作成と削除

レプリカ同意の作成

新規のレプリカ合意を作成するには、**ipa-replica-manage connect** コマンドを使用します。

```
$ ipa-replica-manage connect server1.example.com server2.example.com
```

このコマンドで *server1.example.com* から *server2.example.com* へ、および *server2.example.com* から *server1.example.com* への双方向の新規レプリカ合意が作成されます。

ipa-replica-manage connect で 1 つのサーバーだけを指定知ると、IdM はローカルホストとその指定されたサーバー間のレプリカ合意を作成します。

証明書サーバーのレプリカ合意を新規作成するには **ipa-csreplica-manage connect** コマンドを使用します。

複製合意の削除

レプリカ合意を削除するには、**ipa-replica-manage disconnect** コマンドを使用します。

```
$ ipa-replica-manage disconnect server1.example.com server4.example.com
```

このコマンドで *server1.example.com* から *server4.example.com* へ、および *server4.example.com* から *server1.example.com* へのレプリケーションが無効になります。

ipa-replica-manage disconnect コマンドは、レプリカ合意が削除されるだけです。Identity Management レプリカトポロジー内のサーバーはどちらも残されたままです。すべてのレプリカ合意とレプリカに関するデータを削除するには、**ipa-replica-manage del** コマンドを使用します。これで Identity Management ドメインからレプリカが完全に削除されます。

```
$ ipa-replica-manage del server2.example.com
```

証明書サーバーのレプリカ合意を削除するには、**ipa-csrelica-manage disconnect** コマンドを使用します。同様に、2 つのサーバー間証明書合意とデータすべてを削除するには、**ipa-csrelica-manage del** コマンドを使用します。

D.3.4. 手動によるレプリカ更新の開始

直接のレプリカ合意のあるレプリカ間におけるデータ変更は、ほぼ即座に複製されます。ただし、直接のレプリカ合意に参加していないレプリカは、更新を即座には受け取りません。

場合によっては、予定外のレプリカ更新を手動で開始する必要があることもあります。たとえば、メンテナンスのためにレプリカをオフラインにする前に、予定の更新までキュー待ちとなっていた変更はすべて、1 つ以上の他のレプリカに送信する必要があります。このような場合、レプリカをオフラインにする前に、手動によるレプリカ更新を開始することができます。

手動によるレプリカ更新を開始するには、**ipa-replica-manage force-sync** コマンドを使用します。このコマンドを実行するローカルホストは、更新を受け取るレプリカになります。更新の送信先となるレプリカを指定するには、**--from** オプションを使用します。

```
$ ipa-replica-manage force-sync --from server1.example.com
```

証明書サーバーのデータのレプリカ更新を開始するには、**ipa-csrelica-manage force-sync** コマンドを使用します。

D.3.5. レプリカの再初期化

レプリカが長期間オフラインだった場合やそのデータベースが破損してしまった場合は、これを *再初期化* することができます。これは初期化に類似したもので、初期化については「[レプリカの作成: 概要](#)」で説明しています。再初期化を実行すると、レプリカは更新されたデータでリフレッシュされます。



注記

この状況では、予定されているレプリカ更新や手動によるレプリカ更新は役に立ちません。これらのレプリカ更新では、レプリカは変更されたエントリーを相互に送信するだけで、再初期化とは違い、データベース全体のリフレッシュは行われません。

レプリカ上のデータレプリカ合意を再初期化するには、**ipa-replica-manage re-initialize** コマンドを使用します。このコマンドを実行するローカルホストは、再初期化されるレプリカになります。データの取得元となるレプリカを指定するには、**--from** オプションを使用します。

```
$ ipa-replica-manage re-initialize --from server1.example.com
```

証明書サーバーのレプリカ合意を再初期化するには **ipa-csreplica-manage re-initialize** コマンドを使用します。

D.3.6. レプリカの削除

レプリカを削除または 降格 すると、トポロジから IdM レプリカが削除されるので、IdM リクエストを処理しなくなります。また、IdM ドメインからホストマシン自体も削除されます。

レプリカを削除するには、レプリカ上で以下のステップを実行します。

1. IdM ドメインの全レプリカ合意を一覧表示します。出力にあるレプリカのホスト名を書き留めます。

```
$ ipa-replica-manage list
server1.example.com: master
server2.example.com: master
server3.example.com: master
server4.example.com: master
```

2. **ipa-replica-manage del** コマンドを使って当該レプリカ用に設定された全合意とそのレプリカについての全データを削除します。

```
$ ipa-replica-manage del server3.example.com
```

3. レプリカがそれ自体の CA で設定されていた場合は **ipa-csreplica-manage del** コマンドも使用して証明書サーバーのレプリカ合意もすべて削除します。

```
$ ipa-csreplica-manage del server3.example.com
```



注記

このステップは、レプリカ自体が IdM CA で設定されていた場合にのみ必要となります。マスターサーバーまたは他のレプリカのみが CA で設定されていた場合は必要ありません。

4. IdM サーバーパッケージをアンインストールします。

```
$ ipa-server-install --uninstall -U
```

D.4. レプリカのマスター CA サーバーへのプロモート

複数のレプリカがあるトポロジでは、そのうちの 1 つがマスター CA として機能し、CA サブシステムの証明書の更新を管理したり、証明書失効リスト (CRL) を生成したりします。デフォルトでは、レプリカを作成する元となる最初のサーバーがマスター CA となります。

マスター CA サーバーをオフラインにする、または使用停止にする場合は、別の CA サーバーをプロモートして、それをマスター CA とします。

- レプリカが CA サブシステムの証明書更新を処理するように設定してください。詳細は、「[証明書更新を処理するサーバーの変更](#)」を参照してください。

- レプリカが CRL を生成するように設定します。「[CRL を生成するサーバーの変更](#)」を参照してください。

D.4.1. 証明書更新を処理するサーバーの変更

どのサーバーが現行の更新マスターであるかを確認するには、以下を実行します。

- Red Hat Enterprise Linux 7.3 およびそれ以降では、**ipa config-show | grep "CA renewal master"** コマンドを使用します。

```
$ ipa config-show | grep "CA renewal master"
IPA CA renewal master: server.example.com
```

- Red Hat Enterprise Linux 7.2 およびそれ以前では、**ldapsearch** ユーティリティーを使用します。以下の例では、更新マスターは **server.example.com** になります。

```
$ ldapsearch -H ldap://$HOSTNAME -D 'cn=Directory Manager' -W -b
'cn=masters,cn=ipa,cn=etc,dc=example,dc=com' '(&(cn=CA)
(ipaConfigString=caRenewalMaster))' dn
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <cn=masters,cn=ipa,cn=etc,dc=example,dc=com> with scope
subtree
# filter: (&(cn=CA)(ipaConfigString=caRenewalMaster))
# requesting: dn
#
# CA, server.example.com, masters, ipa, etc, example.com
dn:
cn=CA,cn=server.example.com,cn=masters,cn=ipa,cn=etc,dc=example,dc=c
om

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

別のサーバーが証明書更新を処理するように設定するには、**ipa-csreplica-manage** ユーティリティーを使用します。

```
# ipa-csreplica-manage set-renewal-master
```

このコマンドは自動的に以前の CA を更新マスターからクローンに再設定します。

付録E 改訂履歴

改訂 7.0-33.2	Thu Nov 30 2017	Terry Chuang
翻訳ファイルを XML ソースバージョン 7.0-33 と同期		
改訂 7.0-33.1	Thu Nov 30 2017	Terry Chuang
翻訳ファイルを XML ソースバージョン 7.0-33 と同期		
改訂 7.0-33	Mon Nov 20 2017	Aneta Šteflová Petrová
『ユーザーおよびグループのスキーマ』と『パスワードポリシーの定義』の章を更新		
改訂 7.0-32	Mon Oct 9 2017	Aneta Šteflová Petrová
数カ所を若干修正		
改訂 7.0-31	Tue Sep 12 2017	Aneta Šteflová Petrová
Web UI のスクリーンショットおよび手順を更新。『Identity Management のスマートカード認証』を若干修正		
改訂 7.0-30	Mon Aug 28 2017	Aneta Šteflová Petrová
『Identity Management のスマートカード認証』、『Identity Management の設定ファイルおよびディレクトリー』および複数のスクリーンショットを更新。その他、マイナーな更新		
改訂 7.0-29	Tue Jul 18 2017	Aneta Šteflová Petrová
7.4 GA 公開用ドキュメントバージョン		
改訂 7.0-28	Mon Apr 24 2017	Aneta Šteflová Petrová
ユーザーグループ、ホストグループ、automember の管理のトピックを更新および統合。その他のマイナーな更新		
改訂 7.0-27	Mon Apr 10 2017	Aneta Šteflová Petrová
Identity Management 用の TLS の更新トピックを追加。さまざまな箇所にマイナー修正および更新		
改訂 7.0-26	Mon Mar 27 2017	Aneta Šteflová Petrová
クライアントのインストール後の留意事項およびパスワードのリセットの有効化のトピックを追加。無効だったリンクを修正。その他のマイナーな更新		
改訂 7.0-25	Mon Feb 27 2017	Aneta Šteflová Petrová
Kerberos ドメインの管理、アップグレード、HBAC の章を更新。さまざまな章でその他の更新		
改訂 7.0-24	Wed Dec 7 2016	Aneta Šteflová Petrová
automember およびパスワードポリシーの章を更新。NIS サポートプラグインの説明を追加。その他のマイナーな更新		
改訂 7.0-23	Tue Oct 18 2016	Aneta Šteflová Petrová
7.3 GA 公開用バージョン		
改訂 7.0-22	Fri Jul 29 2016	Aneta Petrová
Vault の使用に関する章を追加		
改訂 7.0-21	Thu Jul 28 2016	Marc Muehlfeld
はじめにの章を更新。その他のマイナーな修正		
改訂 7.0-19	Tue Jun 28 2016	Aneta Petrová
図を更新。IdM を使用する利点のセクションを「はじめに」の章に追加し、その他マイナーな修正や調整を追加		
改訂 7.0-18	Fri Jun 10 2016	Aneta Petrová
はじめに、サーバーインストール、トラブルシューティングの章を更新。ユーザー、ホスト、サーバー証明書の章を追加。他の章にドメイン DNS 設定変更の章に統合。その他のマイナーな修正		
改訂 7.0-17	Fri May 27 2016	Aneta Petrová

ユーザーライフサイクルの図を追加

改訂 7.0-16 **Thu Mar 24 2016** **Aneta Petrová**
ユーザーライフサイクルを追加。ユーザーアカウント、ユーザー認証、レプリカの管理の章を更新

改訂 7.0-15 **Thu Mar 03 2016** **Aneta Petrová**
複数の DNS のセクションを更新。PAM サービスのドメインの制限をシステムレベルの認証ガイドに移動

改訂 7.0-14 **Tue Feb 09 2016** **Aneta Petrová**
スマートカード、ID ビュー、および OTP を追加。Web UI スクリーンショット、管理の基本、ドメインの制限の章を更新。
アンインストールの手順をインストールの章に移動。インデックスをコメントアウトし、その他の更新を追加

改訂 7.0-13 **Thu Nov 19 2015** **Aneta Petrová**
証明書プロファイル管理、レプリカのマスターへのプロモートにマイナーな更新を追加

改訂 7.0-12 **Fri Nov 13 2015** **Aneta Petrová**
7.2 GA リリース向けのバージョンに DNS およびその他のセクションを追加更新

改訂 7.0-11 **Thu Nov 12 2015** **Aneta Petrová**
7.2 GA リリース向けのバージョン

改訂 7.0-10 **Fri Mar 13 2015** **Tomáš Čapek**
7.1 向けの最終変更を含む非同期更新

改訂 7.0-8 **Wed Feb 25 2015** **Tomáš Čapek**
7.1 GA リリースバージョン

改訂 7.0-6 **Fri Dec 05 2014** **Tomáš Čapek**
スプラッシュページでの分類順序を更新するため再構築

改訂 7.0-4 **Wed Jun 11 2014** **Ella Deon Ballard**
初版